

หุ่นยนต์สำรวจ
OBSERVATION ROBOT



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมระบบควบคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ปีการศึกษา 2536 นั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

033200

ปริญญาโทปีการศึกษา 2536

ภาควิชา วิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง หุ่นยนต์สำรวจ

ผู้จัดทำ

1. น.ส. พรทิพย์ กิจวนิชประเสริฐ 33100237

2. น.ส. พรวิไล ห่อสุวรรณชัย 33100248



อาจารย์ที่ปรึกษา

อาจารย์ที่ปรึกษา

อาจารย์ พรสุท เทศเจริญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หุ่นยนต์สำรวจ
OBSERVATION ROBOT

โดย น.ส.พรทิพย์ กิจวนิชประเสริฐ
 น.ส.พรวิไล ห่อสุวรรณชัย
อาจารย์ที่ปรึกษา

รศ.ดร.โยธิน เปรมปราณีรัชต์
อาจารย์ พรสุข เกศเจริญ

บทคัดย่อ

ปัญญานิพนธ์นี้เป็นการออกแบบและสร้างระบบควบคุมให้แก่ หุ่นยนต์สำรวจ ซึ่งใช้ไมโครคอนโทรลเลอร์ 8032 เป็นตัวควบคุมการเคลื่อนที่ของหุ่นยนต์ ในพื้นที่ ๆ เราต้องการสำรวจ โดยมีกล้องวิดีโอไว้สำหรับตรวจจับภาพ และมีอัลตราโซนิกเป็นตัววัดระยะทางระหว่างตัวรถกับสิ่งกีดขวางแล้วส่งให้กับคอมพิวเตอร์ เมื่อคอมพิวเตอร์มีคำสั่งมาอ่านระยะทางจากไมโครคอนโทรลเลอร์บนตัวรถ จากนั้นผู้ใช้ที่ควบคุมอยู่ที่หน้าจอคอมพิวเตอร์ก็จะนำข้อมูลทั้งหมดที่ได้ มาทำการตัดสินใจว่าควรให้รถเคลื่อนที่ไปด้วยความเร็ว และมุมเลี้ยวเท่าใด จึงสามารถหลบหลีกสิ่งกีดขวางได้แล้วทำการส่งระดับความเร็ว และมุมเลี้ยวที่ต้องการให้กับไมโครคอนโทรลเลอร์ โดยการรับส่งข้อมูลต่าง ๆ จะอาศัยคลื่นวิทยุ เมื่อไมโครคอนโทรลเลอร์รับข้อมูลมา ก็จะทำการสร้างสัญญาณพัลส์ไปควบคุมการเคลื่อนที่ของตัวรถ โดยในการสำรวจนี้จะทำให้ผู้ใช้ทราบว่าพื้นที่บริเวณนั้นมีอะไรอยู่บ้าง

ABSTRACT

This thesis presents the design and implementation of control system for Observation Robot. Microcontroller 8032 is used to control the movement of robot in the observed area. Moreover it has a V.D.O. camera to detect the pictures and ultrasonic sensor to measure the distance between the Observation Robot and obstacle. Then microcontroller will send the data to computer when it has the command to read the distance from microcontroller on the robot. After that the user, controlling in front of the screen, will take all the data to decide speed level and steer level which can control the robot to avoid the obstacles. The data are sent by radio wave. When microcontroller receives the data, it will generate the pluse signals to control the movement of the car. Finally the user will know that what the area has.

กิตติกรรมประกาศ

ปริญญาานิพนธ์ ฉบับนี้สำเร็จลงได้ด้วยดี ก็เพราะได้รับความกรุณาจาก ดร. โยธิน เปรมปราณีรัชต์ และ อาจารย์พรสุข เทศเจริญ เป็นอย่างมาก รวมทั้งการให้แนวทางตลอดจนข้อแนะแนวต่าง ๆ ผู้เขียนต้องขอกราบขอบพระคุณอาจารย์ทั้งสองท่านเป็นอย่างสูงมา ณ โอกาสนี้ด้วย

นอกจากนี้ยังต้องขอขอบคุณ พี่ เพื่อน และ น้อง ๆ ทุกคน ที่ได้ให้ความช่วยเหลือในเรื่องต่าง ๆ จนทำให้ปริญญาานิพนธ์นี้สำเร็จลุล่วงไปได้ด้วยดี

น.ส. พรทิพย์ กิจวนิชประเสริฐ

น.ส. พรวิไล ห่อสุวรรณชัย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
กิตติกรรมประกาศ	II
สารบัญ	III
สารบัญรูป	V
สารบัญตาราง	VII
บทที่ 1. บทนำ	1
1.1 ความเป็นมาของโครงการ	1
1.2 หลักการทำงานของ Observation Robot	2
1.3 วัตถุประสงค์	3
1.4 ขอบเขตงาน	3
1.5 เนื้อหาของปริิฤญานิพนธ์	3
บทที่ 2. ระบบควบคุม	5
2.1 ระบบสื่อสารข้อมูลระหว่างคอมพิวเตอร์	5
2.2 ระบบควบคุม ดี ซี มอเตอร์	12
2.3 การคำนวณหาขนาดของมอเตอร์	18
2.4 การขับเคลื่อนและการเลี้ยว	18
2.5 การตรวจจับสิ่งกีดขวาง	20
2.6 แบตเตอรี่	22
บทที่ 3. การวิเคราะห์และออกแบบส่วนต่าง ๆ ในโครงการ	24
3.1 การตรวจจับสิ่งกีดขวางโดยคลื่นอัลตราโซนิก	24
3.2 การควบคุมความเร็ว	28
3.3 การบังคับเลี้ยว	31
3.4 ไมโครคอนโทรลเลอร์	31
3.5 ตำแหน่งการจัดวางอุปกรณ์	47
บทที่ 4. โปรแกรมควบคุมการทำงาน	48
4.1 FLOW CHART	48
4.2 คำอธิบายโปรแกรม	56
บทที่ 5. ผลการทดลอง	58
บทที่ 6. สรุปผล ปัญหาและแนวทางแก้ไข	65

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ของโรงเรียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก. โปรแกรมควบคุมการทำงาน	67
ภาคผนวก ข. แนวทางการพัฒนา	75
เอกสารอ้างอิง	76



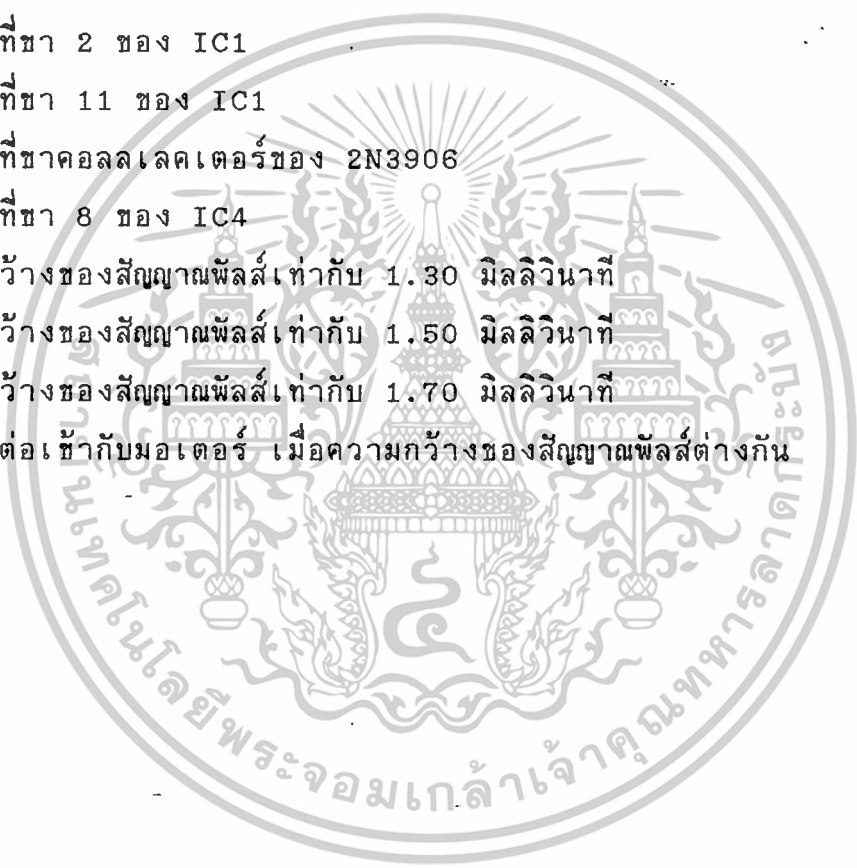
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
1.1 แผนผังหลักการทำงานของ Observation Robot	3
2.1 ขบวนการเอาท์พุท	5
2.2 ขบวนการอินพุท	6
2.3 การจัด PIA ให้ทำงานแบบต่าง ๆ	6
2.4 การรับและการส่งข้อมูลแบบอนุกรม	8
2.5 ประเภทของการติดต่อสื่อสารแบบอนุกรม	8
2.6 รูปแบบของการสื่อสารแบบอะซิงโครนัส	9
2.7 ลักษณะของข้อมูลแบบขนานเมื่อถูกเปลี่ยนเป็นแบบอนุกรม	10
2.8 ช่วงเวลาของสัญญาณเมื่อส่งด้วยอัตรา 2400 บิต	11
2.9 แผนผังการส่งข้อมูลแบบอนุกรม	12
2.10 ลักษณะการทำงานของบิตสตาร์ท	12
2.11 ระบบการควบคุม ดี ซี มอเตอร์แบบพื้นฐาน	12
2.12 ระบบบังคับความเร็ว	14
2.13 วงจรสำหรับวัดความเร็วของมอเตอร์	15
2.14 การวัดความเร็วของมอเตอร์	16
2.15 ระบบควบคุมตำแหน่งหมุน	17
2.16 ลักษณะการเลี้ยงของรลยนต์	19
3.1 แผนผังการทำงานของตัวตรวจจับวัตถุและวัดระยะทาง	25
3.2 วงจรที่สมบูรณ์ของตัวตรวจจับและวัดระยะทาง	26
3.3 แผนผังเวลาแสดงรูปคลื่นตามจุดต่าง ๆ ในวงจร	27
3.4 แสดงวงจรของ Speed Control	29
3.5 TMOD:Timer/Counter Mode Control Register	34
3.6 Timer/Counter1 Mode 0:13 Bits Counter	34
3.7 TCON:Timer/Counter Control Register	35
3.8 Timer/Counter1 Mode 2:8 Bits Auto-Reload Counters	36
3.9 Timer/Counter0 Mode 3:Two 8 Bits Counters	36
3.10 Timer2 Capture Mode	37
3.11 Timer2 in Auto-Reload Mode	38
3.12 T2CON:Timer/Counter2 Control Register	38

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่	หน้า
3.13 SCON:Serial Port Control Register	40
3.14 Timer1 Generated Commoning Used Baud Rate	42
3.15 MCS-51 Interrupt Sources	42
3.16 IE:Interrupt Enable	44
3.17 IP:Interrupt Priority Register	45
3.18 ภาพถ่ายตัวจริงของโครงงานนี้	46
3.19 ตำแหน่งการจัดวางอุปกรณ์ต่าง ๆ	47
5.1 สัญญาณอินพุท	60
5.2 สัญญาณที่ขา 4 ของ IC1	60
5.3 สัญญาณที่ขา 2 ของ IC1	61
5.4 สัญญาณที่ขา 11 ของ IC1	61
5.5 สัญญาณที่ขาคอลเลคเตอร์ของ 2N3906	61
5.6 สัญญาณที่ขา 8 ของ IC4	62
5.7 ความกว้างของสัญญาณพัลส์เท่ากับ 1.30 มิลลิวินาที	62
5.8 ความกว้างของสัญญาณพัลส์เท่ากับ 1.50 มิลลิวินาที	62
5.9 ความกว้างของสัญญาณพัลส์เท่ากับ 1.70 มิลลิวินาที	62
5.10 สัญญาณที่ต่อเข้ากับมอเตอร์ เมื่อความกว้างของสัญญาณพัลส์ต่างกัน	63



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่		หน้า
3.1	แสดงการกำหนดค่า BIT ในรีจิสเตอร์ T2CON	37
5.1	แสดงความเร็วของมอเตอร์ขณะไม่มีวัตถุวางอยู่บนตัวรถ	58
5.2	แสดงความเร็วของมอเตอร์ขณะที่มีน้ำหนักวางอยู่บนตัวรถ	59
5.3	แสดงรัศมีการเลี้ยวที่ต่ำความกว้างของสัญญาณพัลส์ต่าง ๆ	59



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

ในปัจจุบันนี้ เทคโนโลยีด้านหุ่นยนต์และการนำคอมพิวเตอร์มาประยุกต์ใช้ในการควบคุมการทำงานของเครื่องจักรกลต่าง ๆ ได้เจริญก้าวหน้าอย่างรวดเร็ว จึงมีการนำหุ่นยนต์มาใช้ในการทำงานที่เสี่ยงต่ออันตรายต่าง ๆ ภายใต้งานอุตสาหกรรมแทนที่คน อีกทั้งยังเพื่อเพิ่มความรวดเร็วและแน่นอนในการทำงาน และช่วยลดต้นทุนในการผลิตด้วย ดังนั้น จึงมีความจำเป็นอย่างยิ่งที่จะต้องมีการพัฒนาการของหุ่นยนต์ เพื่อให้สามารถทำงานได้ดีและใกล้เคียงกับมนุษย์มากขึ้น อีกทั้งยังต้องมีการพัฒนาบุคลากรที่จะมาทำการควบคุมหุ่นยนต์ เพื่อที่จะรองรับความเจริญก้าวหน้าของหุ่นยนต์ที่จะเกิดขึ้นในอนาคต

หุ่นยนต์อาจแบ่งได้เป็น 2 ประเภทใหญ่ ๆ คือ

1. Fixed Robot คือ หุ่นยนต์ที่ถูกตรึงกับฐานที่ถูกยึดคงที่ ไม่สามารถเคลื่อนที่ได้อิสระ
2. Mobile Robot คือ หุ่นยนต์ที่ถูกตรึงกับฐานที่เคลื่อนที่ได้อย่างอิสระ โดยที่ฐานอาจมีล้อหรือตีนระบบ

จากทั้งสองประเภทของหุ่นยนต์ คือ Fixed Robot และ Mobile Robot หุ่นยนต์ประเภทที่สอง ค่อนข้างจะเป็นหุ่นยนต์ที่มีความยืดหยุ่นสูงกว่าประเภทแรก ซึ่งในปริณยานิพนธ์นี้ ก็จะเป็นเรื่องเกี่ยวกับ Mobile Robot

Mobile Robot มีด้วยกันหลายชนิด ซึ่งโดยทั่วไปแล้ว Mobile Robot จะหมายถึง เครื่องจักรซึ่งสามารถเคลื่อนที่ได้ และทิศทางที่เคลื่อนที่ถูกควบคุมโดยอัตโนมัติ ซึ่งการเคลื่อนที่นั้นมีด้วยกันหลายรูปแบบ อาจเป็นการเคลื่อนที่บนอากาศหรือใต้น้ำ ซึ่งมี 3 Translational และ 3 Rotational หรืออาจเป็นการเคลื่อนที่เฉพาะพื้นผิว ซึ่งมี 2 Translational และ 1 Rotational หรืออาจจะถูกจำกัดทิศทาง ถ้ามีแค่ 1 Translational และไม่มี Rotational แต่ถ้า Robot นี้ เคลื่อนที่เฉพาะพื้นผิวมีการใช้ข้อต่อ อาจจะทำให้ตัว Robot เคลื่อนที่ได้ใน 3 Dimension

การเคลื่อนที่ของ Robot จะใช้ ล้อ, Tracks, ขา หรือ Air Cushions แต่อันหลังไม่ค่อยนิยม ซึ่งส่วนใหญ่แล้วก็จะใช้ล้อเพราะเป็นวิธีที่ง่าย จะใช้กับพื้นถนนที่เรียบ สำหรับระบบล้อนั้นมีด้วยกันหลายแบบ อาทิเช่น

ไม่ว่ากรณีใดๆ - ระบบ 3 ล้อ เลี้ยวโดยล้อหน้า 1 ล้อ โดยให้ล้อ หลัง 2 ล้อ เป็นล้อขับ

-ระบบ 4 ล้อ 2 ล้อหน้าเป็นทั้งล้อขับและล้อสำหรับเลี้ยว ส่วน 2 ล้อหลัง เป็นล้อช่วย

ระบบล้อที่ใช้กันอาจเป็น ระบบ 3 ล้อ หรือ 4 ล้อ แต่เมื่อเปรียบเทียบกับระบบ 4 ล้อ กับ 3 ล้อแล้ว ระบบ 4 ล้อจะมีเสถียรภาพดีกว่า กำลังลากจูงดีกว่า ช่องว่างทางกลน้อยกว่า แต่ระบบ 3 ล้อ ก็มีข้อดีตรงที่ระบบเลี้ยวไม่ยุ่งยากซับซ้อน โครงรมมีน้ำหนักเบาว่า ซึ่งทำให้ประหยัดพลังงานจากแบตเตอรี่ได้มาก

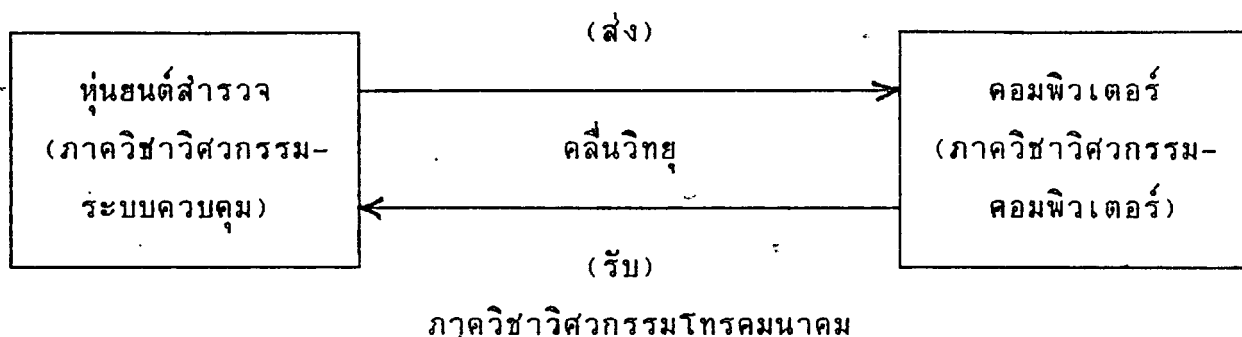
1.1 ความเป็นมาของโครงการ

การควบคุมการเคลื่อนที่ของ Robot มีอยู่ด้วยกันหลายวิธี ตัวอย่างเช่น ระบบ-AGV (Automated Guided Vehicle) ที่นิยมใช้กันอย่างแพร่หลายในโรงงานอุตสาหกรรมแบบอัตโนมัติ ส่วนใหญ่เป็นระบบที่ใช้ทางเดินนำร่อง ได้แก่ สายตัวนำฝังใต้พื้น แลปสีติดบนพื้น สำหรับการฝังสายตัวนำใต้พื้นนั้น ทำให้ระบบไม่ยืดหยุ่น การแก้ไขเปลี่ยนแปลงกระทำได้ยาก นอกจากนั้นในระบบ AGV ที่ซับซ้อน การติดตั้งรวมทั้งการสื่อสารด้วยวิธีเหนี่ยวนำผ่านสายตัวนำ จะเสียค่าใช้จ่ายสูง ปัจจุบันทิศทางการพัฒนา AGV จึงมุ่งไปที่การพยายามทำให้ AGV สามารถเคลื่อนที่ได้อย่างอิสระโดยไม่อาศัยระบบการนำทาง การพัฒนา AGV มีการวิจัยทั้งจากบริษัทผู้ผลิตและจากสถาบันวิจัยต่าง ๆ เพื่อหาเทคโนโลยีใหม่ๆ สำหรับ AGV เพื่อให้ AGV พึ่งพาทางเดินนำร่องน้อยลง โดยให้มีโปรแกรมควบคุมการทำงาน จากปัญหาที่กล่าวมาแล้วดังกล่าวข้างต้น ในโครงการนี้จึงได้พัฒนาการเคลื่อนที่จากระบบ AGV โดยไม่ต้องใช้ทางเดินนำร่อง แต่อาศัยกล้องวิดีโอ และคลื่นอัลตราโซนิก นอกจากนี้ยังอาศัยการควบคุมจากไมโครคอนโทรลเลอร์และคอมพิวเตอร์ควบคุมการเคลื่อนที่ โดยโครงการนี้เป็นความร่วมมือกันระหว่าง 3 ภาค คือ ภาควิชาวิศวกรรมระบบควบคุม ภาควิชาวิศวกรรมคอมพิวเตอร์ และ ภาควิชาวิศวกรรมโทรคมนาคม

1.2 หลักการทำงานของ Observation Robot

หุ่นยนต์สำรวจ (ซึ่งในที่นี้เราจะขอใช้คำว่ารถแทน) จะเคลื่อนที่ไปในพื้นที่ที่เราต้องการจะสำรวจ โดยที่รถสำรวจจะมีกล้องวิดีโอไว้สำหรับตรวจจับภาพ และเครื่องตรวจจับสิ่งกีดขวางด้วยอัลตราโซนิก คอยวัดระยะทางระหว่างตัวรถกับสิ่งกีดขวาง แล้วส่งให้กับคอมพิวเตอร์ เมื่อคอมพิวเตอร์มีคำสั่งมาอ่านระยะทางจากไมโครคอนโทรลเลอร์บนตัวรถ จากนั้น ผู้ใช้ที่ทำการควบคุมอยู่ที่หน้าจอคอมพิวเตอร์ก็จะนำข้อมูลทั้งหมดที่ได้มาทำการตัดสินใจว่าควรให้รถเคลื่อนที่ไปด้วยความเร็ว และมุมเลี้ยวเท่าใด จึงสามารถหลบหลีกสิ่งกีดขวางได้ แล้วทำการส่งระดับความเร็ว และมุมเลี้ยวที่ต้องการให้กับไมโครคอนโทรลเลอร์ โดยการรับส่งข้อมูลต่าง ๆ จะอาศัยคลื่นวิทยุ เมื่อไมโครคอนโทรลเลอร์รับข้อมูลมาก็จะทำการสร้างสัญญาณพัลส์ไปควบคุมการเคลื่อนที่ของตัวรถให้ได้ตามที่ผู้ใช้ต้องการ โดยใน

การสำรวจนี้จะทำให้ผู้ใช้ทราบว่า พื้นที่บริเวณนี้มีอะไรอยู่บ้าง



รูปที่ 1.1 แสดงแผนผังหลักการทำงานของ Observation Robot

1.3 วัตถุประสงค์

เพื่อศึกษาถึงส่วนขับเคลื่อนและส่วนตรวจจับสิ่งกีดขวางแล้วควบคุมการเคลื่อนที่ของ Robot ให้สามารถสำรวจพื้นที่และสามารถหลบหลีกสิ่งกีดขวางได้ นอกจากนี้ยังสามารถเคลื่อนที่ไปตามพื้นที่ที่กำหนดได้

1.4 ขอบเขตงาน

ในโครงงานนี้ได้แบ่งงานออกเป็น 2 ส่วน คือ

1. ส่วนของ Hardware

- ประกอบด้วย
 - ตัวรถ
 - ตัวตรวจจับสิ่งกีดขวางโดยใช้คลื่นอัลตราโซนิก
 - Speed Control
 - Power Supply

2. ส่วนของ Software

โปรแกรมควบคุมการเคลื่อนที่ของตัวรถ โดยควบคุมความเร็ว ตำแหน่ง และการบังคับเลี้ยว ซึ่งทำการรับข้อมูลจาก Computer และทำการส่งระยะห่างของสิ่งกีดขวางที่รับมาจาก Ultrasonic Sensor เพื่อให้หลบหลีกสิ่งกีดขวางและสำรวจพื้นที่ได้

1.5 เนื้อหาของปริิญาานิพนธ์

บทที่ 1 กล่าวถึง Mobile Robot, ความเป็นมาของโครงงานนี้, หลักการทำงานของ Observation Robot, วัตถุประสงค์, ขอบเขตและเนื้อหาของปริิญาานิพนธ์ เอกสารนี้เป็นเอกบทที่ 2 กล่าวถึงกรายละเอียดและทฤษฎีต่างๆที่เกี่ยวข้องกับระบบควบคุม Observation Robot อีกในปริิญาานิพนธ์นี้ยังได้แก่ต้องระบบสื่อสารข้อมูลระหว่างคอมพิวเตอร์,

ระบบควบคุม ดี ซี มอเตอร์, การขับเคลื่อน และการเลี้ยว, การตรวจจับสิ่งกีดขวางและ
แบดเตอร์

บทที่ 3 กล่าวถึงการออกแบบ และการสร้างระบบควบคุม Observation
Robot ได้แก่ Ultrasonic Sensor การบังคับเลี้ยวและการควบคุมความเร็ว คุณสมบัติ
ของไมโครคอนโทรลเลอร์และการทำขบวนการ Serial, Timer และการ Interrupt
โครงสร้างของหุ่นยนต์สำรวจ

บทที่ 4 กล่าวถึงโปรแกรมควบคุมการทำงาน ซึ่งประกอบด้วย โปรแกรมหลัก
ซึ่งทำหน้าที่กำหนดค่าพารามิเตอร์และค่าเริ่มต้นต่าง ๆ ของรีจิสเตอร์ และมีโปรแกรมย่อย
ในส่วนของการรับส่งข้อมูลระหว่างพอร์ตอนุกรมและคอมพิวเตอร์ การรับข้อมูลจากSensor
และนำไปประมวลผล การสร้างสัญญาณพัลส์ส่งไปควบคุมความเร็วและบังคับเลี้ยว

บทที่ 5 ผลการทดลองการทำงานของระบบต่าง ๆ

บทที่ 6 สรุปผลการทดลอง รวมทั้งปัญหาและแนวทางแก้ไข

ภาคผนวก ก. แสดงรายละเอียดของโปรแกรมโดยสมบูรณ์

ภาคผนวก ข. กล่าวถึงแนวทางในการพัฒนา AGV ในขั้นต่อไป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ระบบควบคุม

2.1 ระบบสื่อสารข้อมูลระหว่างคอมพิวเตอร์

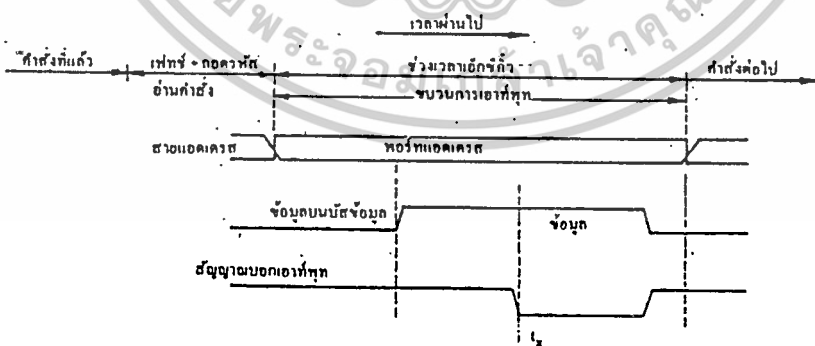
การทำงานของระบบไมโครคอมพิวเตอร์ ซีพียูจะมีการติดต่อกับระบบภายนอก อยู่เสมอ การติดต่อที่ทำตลอดเวลา คือ ขบวนการเขียนและอ่านข้อมูลจากหน่วยความจำ นอกเหนือไปจากหน่วยความจำแล้ว ซีพียูยังต้องติดต่อกับระบบอื่น ๆ อีกคือระบบอินพุตและเอาต์พุต ซึ่งเป็นการส่งและรับข้อมูลจากอุปกรณ์ภายนอก ในหัวข้อนี้จะกล่าวถึงขบวนการอินพุตและเอาต์พุต

ขบวนการอินพุตและเอาต์พุต (Input & Output Sequence)

ขบวนการอินพุตและเอาต์พุตก็คือการส่งข้อมูลจากซีพียู และรับข้อมูลจากพอร์ตตามลำดับ การรับส่งเกี่ยวข้องกับสัญญาณต่อไปนี้คือ

1. สัญญาณเลือกพอร์ตต่าง ๆ (Address Lines)
2. สัญญาณข้อมูล
3. สัญญาณควบคุม

- สัญญาณเลือก ทำหน้าที่เลือกพอร์ตใดพอร์ตหนึ่งจากพอร์ตทั้งหมด สัญญาณควบคุมทำหน้าที่บอกทิศทาง และเวลาการย้ายข้อมูลว่าออกจากซีพียูหรือเข้าหาซีพียูในจังหวะใด ขบวนการอินพุตและเอาต์พุต เริ่มจากที่ไมโครโปรเซสเซอร์อ่านคำสั่ง (ซึ่งบอกว่าจะให้ทำ I/O) จากหน่วยความจำและถอดรหัสแล้วไมโครโปรเซสเซอร์จะเริ่มขบวนการอินพุตและเอาต์พุตตามคำสั่งนั้นดังรูปที่ 2.1

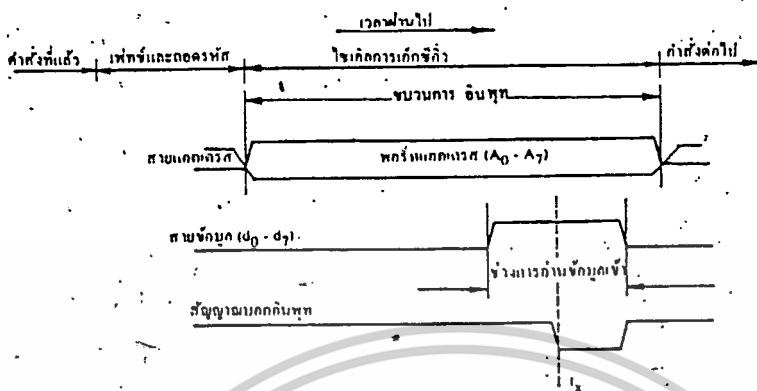


รูปที่ 2.1 ขบวนการเอาต์พุต

หลังจากที่ทำการเฟรชและถอดรหัสแล้ว ค่าของพอร์ตจะถูกส่งออกมาบนสายแอดเดรสในเวลาถัดมา ข้อมูลก็จะถูกส่งออกมาจากซีพียูออกมาที่บัสข้อมูลถัดมาอีกเล็กน้อย

หลังจากนั้นซีพียูส่งสัญญาณควบคุมออกมา ก็เป็นการบอกวงจรรีโมเตอร์เฟสถึงความพร้อมของข้อมูลบนสายบัสด้วย

ในทำนองเดียวกันรูปที่ 2.2 เป็นขบวนการอินพุทข้อมูลจากอินพุทพอร์ต



รูปที่ 2.2 ขบวนการอินพุท

หลังจากที่เฟิร์มและถอดรหัสแล้ว ซีพียูจะส่งที่อยู่ของพอร์ตออกมาทางสายแอดเดรส และในเวลา t_x ซีพียูจะส่งสัญญาณควบคุมการอินพุทออกมาอ่านเอาข้อมูลบนบัสข้อมูลเข้าไปในซีพียู

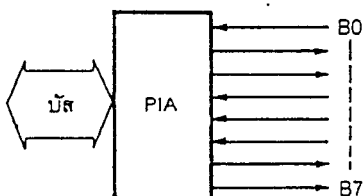
การอินเทอร์เฟส

การเชื่อมต่อระบบไมโครคอมพิวเตอร์กับอุปกรณ์ภายนอกส่วนใหญ่ จะใช้การเชื่อมต่อแบบขนาน (Parallel Transmission) กับแบบอนุกรม (Serial Transmission)



(ก) ส่งข้อมูลออกครั้งละไบต์ไปให้อุปกรณ์บริวาร

(ข) รับข้อมูลเข้าครั้งละไบต์จากอุปกรณ์บริวาร



(ค) รับส่งข้อมูลครั้งละไม่ถึงไบต์ โดยแบ่งแต่ละบิตว่ารับหรือส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้นรูปที่ 2.3 มีแสดงการจัด PIA ให้ทำงานแบบต่างๆ ทุกครั้งที่มีการนำไปใช้

-การเชื่อมต่อแบบขนาน

หน่วยรับส่งข้อมูลเข้าออกแบบขนาน สามารถรับข้อมูลขนานกันครั้งละไบต์ ส่งข้อมูลออกขนานกันครั้งละไบต์ หรือจะรับข้อมูลเข้า และส่งข้อมูลออกครั้งละไม่ถึงไบต์ก็ได้ ดังแสดงในรูปที่ 2.3

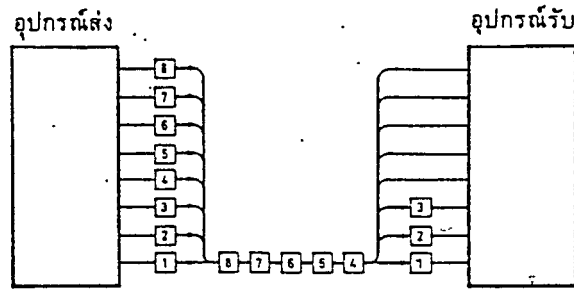
นอกเหนือจากการรับส่งข้อมูลแล้ว จะต้องมีการแลกเปลี่ยน สัญญาณควบคุม ระหว่างหน่วยรับส่งข้อมูลเข้าออก และอุปกรณ์บริวาร สัญญาณควบคุมมีประโยชน์มากเพราะ โดยปกติอุปกรณ์บริวารจะทำงานในอัตราที่ช้ากว่าระบบคอมพิวเตอร์ หากระบบข้อมูลคอมพิวเตอร์รับส่งข้อมูลด้วยอัตราที่เร็วจนอุปกรณ์บริวารรับไม่ทันก็จะเกิดการสูญหายของข้อมูล วิธีแก้คือ อุปกรณ์บริวารจะมีสัญญาณที่แจ้งว่าตนเองพร้อมที่จะรับข้อมูลเพิ่มเติม และหน่วยรับส่งข้อมูลเข้าออกก็จะมีสัญญาณคอยแจ้งว่าจะมีการส่งข้อมูล การประสานงานกันระหว่างข้อมูลทั้ง 2 นี้เราเรียกว่า วิธีจับมือ (Handshaking) ระหว่างหน่วยรับส่งข้อมูลเข้าออก และอุปกรณ์บริวาร ซึ่งทำให้การรับส่งข้อมูลมีประสิทธิภาพ ตลอดจนปราศจากการสูญหาย เนื่องจากหน่วยรับส่งข้อมูลแบบขนานสามารถจัดข้อมูลได้หลายแบบ ดังแสดงในรูปที่ 2.3 เพราะฉะนั้นภายในหน่วยนี้ เช่น PIA จะมีรีจิสเตอร์ควบคุมสำหรับเก็บข้อมูลบ่งลักษณะการทำงาน

-การเชื่อมต่อแบบอนุกรม

การเชื่อมต่อแบบอนุกรม การเชื่อมต่อแบบนี้ได้นำไปประยุกต์ใช้งานกันอยู่มากพอสมควร เช่น การโอนย้ายข้อมูลระหว่างไมโครคอมพิวเตอร์ เพราะทำได้สะดวกง่าย และสิ้นเปลืองค่าใช้จ่ายน้อยกว่าการต่อแบบขนาน นอกจากนี้การเชื่อมต่อแบบอนุกรมยังทำให้สามารถส่งผ่านข้อมูลไปในระยะทางไกล ๆ ได้

ในการโอนถ่ายข้อมูลแบบอนุกรมนั้นข้อมูลจะได้รับการส่งออกมาครั้งละ 1 บิต ระหว่างจุดรับและจุดส่ง จะเห็นว่า การส่งข้อมูลแบบอนุกรมนี้นี้จะช้ากว่าการส่งข้อมูลแบบขนาน แต่ยังใช้การส่งข้อมูลชนิดนี้อยู่ เพราะว่าตัวกลางการสื่อสารต้องการเพียงช่องเดียว หรือมีสายเพียงคู่เดียว ค่าใช้จ่ายในการใช้ตัวกลาง เช่น สายไฟจะต้องถูกกว่าการส่งข้อมูลแบบขนานอย่างแน่นอน สำหรับการส่งข้อมูลในระยะทางไกล ๆ ก็จะเป็นเรื่องดี เพราะเรามีระบบการสื่อสารทางโทรศัพท์ไว้ใช้งานอยู่แล้ว จึงสามารถนำมาใช้งานในการส่งข้อมูลแบบอนุกรมนี้นี้ได้

รูปที่ 2.4 แสดงให้เห็นการส่งข้อมูลแบบอนุกรม ข้อมูลจากจุดส่งจะถูกเปลี่ยนให้เป็นแบบอนุกรมเสียก่อนแล้วค่อยทยอยส่งครั้งละ 1 บิตไปยังจุดรับ ณ จุดรับจะมีกลไกในการเปลี่ยนข้อมูลที่ส่งมาครั้งละบิตเป็นสัญญาณแบบขนาน ซึ่งลงตัวพอดี นั่นคือ บิต 1 ลงที่บัสข้อมูลเส้นที่ 1 พอดี การที่จะแปลงสัญญาณจากแบบอนุกรมครั้งละบิตให้ลงตัวพอดีนั้นจำเป็นต้องมีกลไกที่เหมาะสมเพื่อป้องกันการผิดพลาดในการรับ กลไกที่ว่านี้แบ่งออกเป็น 2 แบบ คือ แบบซิงโครนัส (Synchronous) และแบบอะซิงโครนัส (Asynchronous) นำไปใช้



รูปที่ 2.4 การรับและการส่งข้อมูลแบบอนุกรม

รูปแบบของการติดต่อสื่อสารแบบอนุกรม

การติดต่อแบบอนุกรมอาจจะแบ่งตามรูปลักษณะการส่งข้อมูลได้ 3 แบบ คือ

1. แบบซิมเพลกซ์ (Simplex) เป็นการส่งข้อมูลได้ทางเดียวเท่านั้น บางครั้งก็เรียกว่า การส่งทิศทางเดียว

2. แบบฮาล์ฟดูเพลกซ์ (Half Duplex) เป็นการส่งและรับข้อมูลได้ทั้ง 2 สถานะ แต่จะต้องผลัดกันรับและผลัดกันส่งข้อมูล จะส่งและรับข้อมูลพร้อมกันไม่ได้

3. แบบฟูลดูเพลกซ์ (Full Duplex) ทั้ง 2 สถานะ สามารถรับและส่งข้อมูลในเวลาเดียวกัน



รูปที่ 2.5 ประเภทของการติดต่อสื่อสารแบบอนุกรม

รูปที่ 2.5 แสดงถึงรูปแบบของการส่งข้อมูลประเภทต่าง ๆ จะเห็นว่าการส่งข้อมูลแบบฟูลดูเพลกซ์ และฮาล์ฟดูเพลกซ์จะไม่ขึ้นอยู่กับจำนวนสายในการติดต่อ เพราะบางครั้งอาจใช้วิธีการแยกความถี่ที่ต่างกันระหว่างสัญญาณข้อมูลของฝ่ายส่งและฝ่ายรับ

เอกสารนี้เป็นเอกความเร็วในการถ่ายโอนข้อมูลแบบอนุกรมไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ความเร็วของการถ่ายโอนข้อมูลแบบอนุกรมมีหน่วยวัดเป็นบิตต่อวินาทีไปหรือ



เรียกว่า บีพีเอส(bps) แต่เรายังมีหน่วยหนึ่งที่ใช้กันมากคือ โบท์เรตหรืออัตราโบด (Baud Rate) ซึ่งหมายถึงการเปลี่ยนแปลงของสัญญาณ 1 ครั้งอาจจะแสดงถึงการส่งข้อมูลแบบอนุกรมได้มากกว่า 1 บิต อัตราการส่งข้อมูลเป็นจำนวนบิตจึงเท่ากับ อัตราโบดคูณกับจำนวนบิตใน 1 โบท์

การสื่อสารแบบอะซิงโครนัส

การสื่อสารแบบนี้ประกอบด้วยบิตเริ่มต้นหรือบิตสตาร์ท(Start Bit) และบิตสิ้นสุดหรือบิตสต็อป(Stop Bit)ดังรูปที่ 2.6



รูปที่ 2.6 รูปแบบของการสื่อสารแบบอะซิงโครนัส

ขณะที่สถานะของการส่งเป็นแบบว่างหรือแบบไอดีล (Idle) คือ ยังไม่มีสัญญาณส่งออกมาจะมีสัญญาณหรือแรงดันตลอดเวลา เพื่อความแน่ใจว่าฝ่ายรับยังติดต่อกับฝ่ายส่ง ฝ่ายส่งจะเริ่มส่งข้อมูลบิตจุดเริ่มต้น สัญญาณของอะซิงโครนัสจะเป็น 0 หนึ่งช่วงสัญญาณนาฬิกา บิตนี้จะเรียกว่า บิตสตาร์ท ข้อมูล 1 ตัวอักษรที่ตามหลังบิตสตาร์ทจะมีขนาดตั้งแต่ 5 บิตถึง 8 บิต โดยบิตที่มีค่าน้อยที่สุด (LSB) จะถูกส่งออกมาก่อนและไล่ไปจนถึงบิตที่มีค่ามากที่สุด (MSB) การเข้ารหัสอักขระนี้ส่วนมากจะนิยมใช้ รหัสแอสกี (ASCII Code)

แรกเริ่มทีเดียวของการส่งข้อมูลจะส่งรหัสโบดอต (Baudot Code) ซึ่งใช้ 5 บิตในการแทนอักขระ 1 ตัว ส่วนที่ตามหลังข้อมูลก็จะเป็นบิตพาริตี ซึ่งอาจจะใช้หรือไม่ใช้ก็ได้ บิตพาริตีจะทำหน้าที่ตรวจสอบความถูกต้องของสัญญาณที่ได้รับ บิตพาริตีอาจจะแบบคู่ (Even) หรือเป็นแบบคี่ (Odd) ก็ได้ หมายความว่า ถ้าหากเป็นพาริตีคู่ จำนวนบิตที่เป็น 1 ในช่วงบิตข้อมูลกับบิตพาริตีรวมกันแล้วต้องเป็นเลขคู่ ผู้ส่งข้อมูลจะต้องทำหน้าที่ตรวจสอบข้อมูลแล้วใส่บิตพาริตีเอง

ฝ่ายรับ เมื่อรับสัญญาณแล้วก็ต้องตรวจสอบดูว่าเป็นจริงดังสถานะการณที่ตั้งเอาไว้หรือไม่ หากผิดพลาดก็หมายความว่า สัญญาณที่รับนั้นผิดพลาดไปจากสถานีส่งที่ส่งออกมา ทั้งนี้ทั้งนั้นจะต้องผิดเป็นจำนวนคี่เท่านั้นคือ ผิดไป 1 บิต 3 บิตหรือ 5 บิตพร้อมกัน จึงจะตรวจสอบได้ว่าผิด เช่น ถ้าผิดเป็นจำนวนคู่ผลรวมของจำนวนบิตที่เป็น 1 ก็ยังเป็นคู่อยู่ดี

ทั้งนี้ทั้งนั้นไม่ได้หมายความว่า พาริตีคู่จะตรวจสอบความผิดพลาดเป็นจำนวนคู่ได้ ความจริงแล้วสามารถตรวจสอบความผิดพลาดได้เหมือนพาริตีคู่ แต่แทนที่จะตรวจสอบดูว่าสัญญาณที่รับเข้ามามีจำนวนคู่ ก็ตรวจสอบดูว่ามีจำนวนคี่หรือเปล่า อย่างไรก็ตาม โอกาสที่จะผิดพลาดเป็น 2, 4, 6 หรือ 8 บิตพร้อมกันมีน้อยมากถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลับมาดูสัญญาณอะซิงโครนัส หลังจากบิตพาริตีแล้วก็ต้องมีบิตสตอปซึ่งเป็น 1 ความกว้างของบิตสตอปอาจจะเป็น 1, 1.5 หรือ 2 พัลส์ของสัญญาณนาฬิกา ซึ่งแล้วแต่ผู้รับและผู้ส่งจะตกลงใช้กันเอง

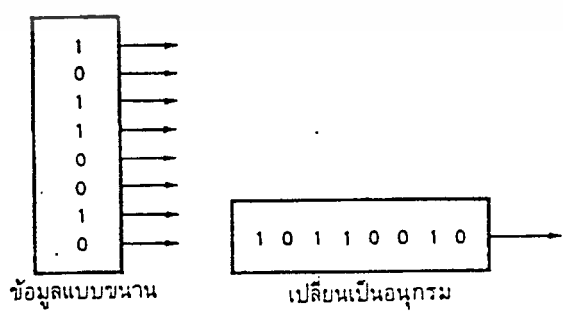
การเริ่มใช้พอร์ทอนุกรมจึงจำเป็นต้องตั้งค่าต่าง ๆ สำหรับการสื่อสาร ซึ่งมีดังต่อไปนี้ คือ

1. ความเร็วของการส่ง
2. ความยาวของรหัส 1 อักขระ
3. บิตตรวจสอบ
4. จำนวนบิตสตอป

การสื่อสารแบบซิงโครนัส

ข้อแตกต่างระหว่างวงจรการส่งข้อมูลอนุกรมแบบซิงโครนัสและแบบอะซิงโครนัสก็คือ ความต่อเนื่องของข้อมูลที่ส่ง ในแบบซิงโครนัส ข้อมูลที่ส่งออกมาเป็นแบบต่อเนื่อง ไม่มีบิตสตาร์ทหรือบิตสตอปหรือแม้กระทั่งบิตพาริตี รูปแบบที่ใช้ในการส่งข้อมูลแบบซิงโครนัสจึงแตกต่างไปจากการส่งข้อมูลแบบอะซิงโครนัส เช่น รูปแบบของบริษัท ไอบีเอ็ม ใช้แบบไบซิงก์หรือ Bisync (Binary Synchronous Transmission)

การซิงโครไนซ์จะทำในระดับอักขระ ซึ่งหมายความว่าอักขระแต่ละตัวมีขอบเขตที่แน่นอน แต่ละอักขระไม่มีบิตสตาร์ทหรือบิตสตอปเหมือนอะซิงโครนัส การซิงโครนัสจะกระทำกันที่จุดเริ่มต้นของการส่งข้อมูล สถานีส่งจะส่งสัญญาณที่เรียกว่า ตัวอักษรนำ (Leading Pad Character) ไปยังสถานีรับก่อนที่จะเริ่มส่งข้อมูล ตัวอักษรนำจะประกอบด้วย 0 และ 1 สลับกัน เพื่อให้สถานีรับจัดสัญญาณนาฬิกาให้ตรงกัน ก่อนส่งข้อมูลก็จะมี การส่งอักขระที่เรียกว่า Syn ตามหลังตัวอักษรนำออกมาสถานีส่งจำเป็นต้องบอกความยาวของข้อมูลมาในกลุ่มนี้ และต้องบอกเครื่องหมายที่เป็นตัวบอกจุดเริ่มต้นของข้อมูลมาด้วย

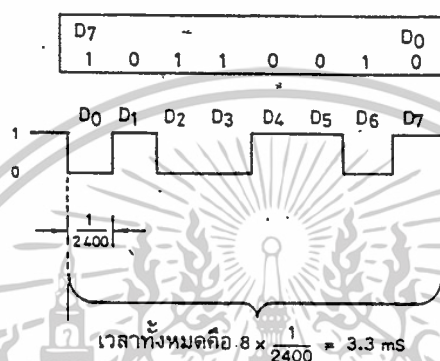


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะในรูปแบบที่ 2.7 อักขระของข้อมูลแบบขนานเมื่อถูกเปลี่ยนเป็นแบบอนุกรมมีการนำไปใช้

ระยะเวลาและอัตราการส่งข้อมูล

ตัวอย่างของการส่งข้อมูลที่มีขนาด 8 บิตจากระบบไมโครโปรเซสเซอร์, ส่งออกที่ช่องสื่อสารแบบอนุกรมแสดงได้ดังรูปที่ 2.7

ในการส่งข้อมูลแบบอนุกรมมีสิ่งสำคัญที่จะต้องพิจารณาคือ ความเร็วของข้อมูลในการส่งซึ่งเราเรียกว่า อัตราบิต(Bit Rate)ตามที่กล่าวมาแล้ว และกรณีที่ให้อัตราการเปลี่ยนแปลงของสัญญาณ f ครั้งต่อข้อมูล 1 บิต จะได้อัตราบิตเท่ากับอัตราบิตต่ออัตราบิตที่ใช้ในการส่งข้อมูลทั่ว ๆ ไปคือ 110, 150, 300, 1200, 2400, 4800 และ 9600 สมมติว่า ถ้าต้องการส่งข้อมูลด้วยอัตราบิต 2400 บิต ข้อมูลจะได้รับการส่งออกไปดังรูปที่ 2.8



รูปที่ 2.8 ช่วงเวลาของสัญญาณเมื่อส่งด้วยอัตรา 2400 บิต

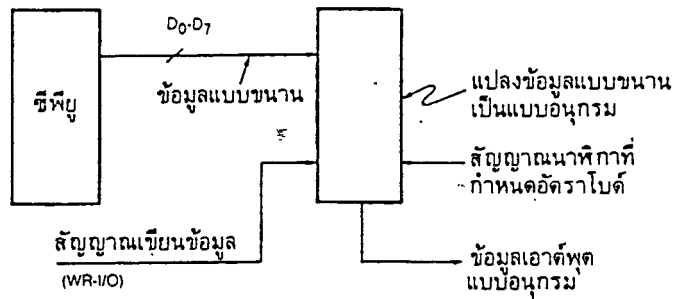
จากรูปที่ 2.8 แสดงให้เห็นช่วงเวลาของการส่ง ซึ่งในช่วงเวลาของแต่ละบิตจะมีขนาดช่วงเวลาเท่ากับ $1/2400$ เท่ากับ $.000416$ วินาที หรือ 416 ไมโครวินาที ดังนั้นถ้าต้องการส่งข้อมูลที่มีขนาด 8 บิต ก็จะใช้เวลาดังนั้น $8 * 416$ ไมโครวินาที หรือเท่ากับ 3328 ไมโครวินาที ซึ่งเมื่อเทียบกับการส่งข้อมูลแบบขนานจะใช้เวลาน้อยกว่า 1 ไมโครวินาที

การแปลงข้อมูลแบบขนานเป็นแบบอนุกรม

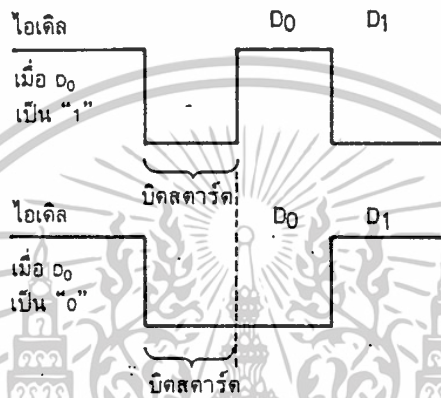
ในการสื่อสารข้อมูลแบบอนุกรมนั้น ต้องแปลงข้อมูลจากแบบขนานมาเป็นแบบอนุกรมก่อน โดยปกติจะใช้ไมโครโปรเซสเซอร์ส่งข้อมูลมายังรีจิสเตอร์ แล้วแปลงข้อมูลเป็นแบบอนุกรมกระบวนการในการแปลงข้อมูลมีดังนี้

1. ทำการเก็บข้อมูลขนาด 8 บิตไปไว้ในชิพรีจิสเตอร์
2. เลื่อนข้อมูลจากชิพรีจิสเตอร์ไปที่ละบิตตามการกำหนดโดยช่วงเวลาโดยอัตราบิตจะเป็นตัวกำหนด

ลักษณะการทำงานจะเป็นไปตามบล็อกไดอะแกรมดังรูปที่ 2.9 โดยให้ชิพผู้ส่งข้อมูลเอาท์พุทที่มีขนาด 8 บิตออกมาเป็นแบบขนานมายังชิพรีจิสเตอร์และให้มีการเลื่อนบิตออกไปกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 แผนผังของการส่งข้อมูลแบบอนุกรม



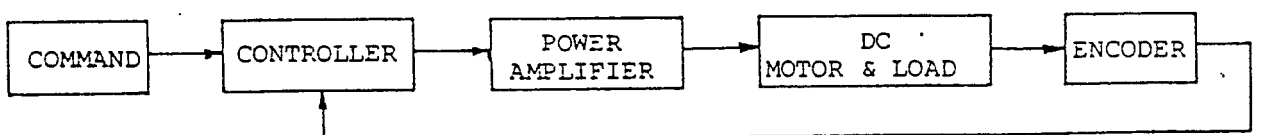
รูปที่ 2.10 ลักษณะการทำงานของบิตสตาร์ท

บิตสตาร์ทและบิตสตอป

การรับส่งข้อมูลแบบอะซิงโครนัส จะต้องมีการบอกจุดเริ่มต้นและจุดสิ้นสุดของเฟรม (Frame) ข้อมูลเสมอ โดยปกติจะให้สถานะไอเดิลเหมือนเช่นบิตสตอป ดังนั้นส่วนของบิตสตาร์ทจะตรงข้ามกับไอเดิล โดยทั่วไปของการส่งข้อมูลจะใช้ 1 บิตเป็นตัวบอกสตาร์ท และใช้ลอจิก 0 เป็นตัวบอกบิตสตาร์ทดังรูปที่ 2.10 ส่วนบิตสตอปจะยาวกว่าที่กำหนดก็ได้ ก่อนที่จะเริ่มต้นเฟรมใหม่

2.2 ระบบควบคุม ดี ซี มอเตอร์

ส่วนประกอบพื้นฐานของระบบควบคุม ดี ซี มอเตอร์แสดงได้ดังบล็อกไดอะแกรมของรูปที่ 2.11 ซึ่งประกอบด้วยบล็อกที่สำคัญ 4 บล็อกคือ :



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ รูปที่ 2.11 ระบบการควบคุม ดี ซี มอเตอร์แบบพื้นฐานครั้งที่มีการนำไปใช้

1. ตัวคอนโทรลเลอร์ เป็นส่วนของระบบที่ทำให้เกิดสัญญาณคอนโทรลไปยังคัม ดิ ซี มอเตอร์และโหลด คอนโทรลเลอร์ที่ทำให้สัญญาณคอนโทรลเป็นสัญญาณอนาลอกเราเรียกว่า อนาลอกคอนโทรลเลอร์ ส่วนคอนโทรลเลอร์ที่ทำให้สัญญาณคอนโทรลเป็นสัญญาณดิจิตอล เราเรียกว่า ดิจิตอลคอนโทรลเลอร์

2. วงจรไดรเวอร์ ส่วนประกอบของระบบที่อยู่ระหว่างตัวคอนโทรลเลอร์ กับ คัม ดิ ซี มอเตอร์และโหลด มีหน้าที่ปรับรูปและขยายสัญญาณให้เหมาะสมก่อนที่จะป้อนเข้าไปขับ คัม ดิ ซี มอเตอร์และโหลด วงจรไดรเวอร์ส่วนใหญ่ ได้แก่ เพาเวอร์แอมป์ลิไฟ ซึ่งอาจแบ่งย่อยออกเป็นลิเนียร์เพาเวอร์แอมป์ลิไฟและพัลส์วิดท์โมดูเลชันแอมป์ลิไฟ

3. ฟีดแบ็คทรานสดิวเซอร์หรือเอนโคเดอร์ เป็นสิ่งประดิษฐ์ที่ใช้รับรู้ หรือ ตรวจจับสัญญาณเอาต์พุตที่ต้องการ โดยไม่ผลของการโหลดตั้ง (loading) สัญญาณที่ตรวจจับได้นี้จะป้อนกลับไปเปรียบสัญญาณอ้างอิง ทำให้ได้สัญญาณเออร์เรอร์ ฟีดแบ็คทรานสดิวเซอร์แบ่งออกได้เป็น 2 แบบคือ อนาลอกทรานสดิวเซอร์ คือสิ่งประดิษฐ์รูปหนึ่งให้เป็นสัญญาณอนาลอก ได้แก่ พวกลาโคเซนเนอเรเตอร์ โปเทนทิโอมิเตอร์ และชิงโคร เป็นต้น ส่วนฟีดแบ็คทรานสดิวเซอร์อีกแบบหนึ่งคือ ดิจิตอลทรานสดิวเซอร์ เป็นสิ่งประดิษฐ์ที่ใช้เปลี่ยนพลังงานรูปหนึ่งให้เป็นสัญญาณดิจิตอล ได้แก่ พวกริโมทรีเอนโคเดอร์วีโวลเวอร์ แมกนิติกพิคอัพ เป็นต้น

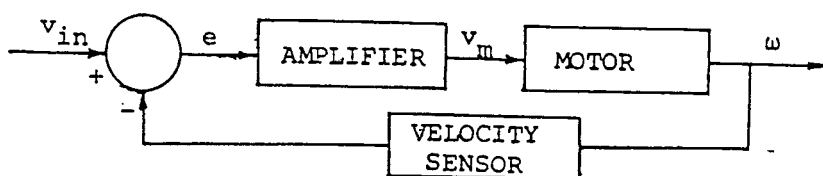
4. คัม ดิ ซี มอเตอร์และโหลด คือ ระบบที่ถูกคอนโทรลหรือส่วนที่ออกแรงทำงาน ซึ่งจะเป็นเครื่องจักรกล(คัม ดิ ซี มอเตอร์) หรืออะไรก็ตามที่ให้ตัวแปร คัม ดิ ซี มอเตอร์ในที่นี้เป็นแบบแม่เหล็กถาวรที่มีคุณสมบัติการทำงานสูง มีอาร์มาเจอร์อินดักชันซ์และแรงเฉื่อยของโรเจอร์ต่ำ

ระบบควบคุมความเร็ว

ระบบการบังคับความเร็ว สามารถจะกำหนดความเร็วให้คงที่ หรือสามารถเปลี่ยนค่าได้ตามที่ต้องการ ระบบนี้มีหลักการอยู่บนพื้นฐานของการป้อนกลับเพื่อให้ได้ความเร็วที่มีความเที่ยงตรงสูง หลักการทำงานของระบบความเร็วของมอเตอร์จะวัดได้โดยตัวเซ็นเซอร์และป้อนกลับไปยังออฟเปอเรชันแนลแอมป์ลิไฟเพื่อเปรียบเทียบกับระดับความเร็วที่ต้องการ ความแตกต่างของค่าทั้งสองจะอยู่ในทอมเออร์เรอร์ของความเร็วจึงจะได้รับการขยายและป้อนให้กับมอเตอร์ เพื่อแก้ไขความเร็วให้มีค่าถูกต้อง คือมีค่าความเบี่ยงเบนของความเร็วเป็นศูนย์หรือพยายามให้สัญญาณบังคับและความเร็วของมอเตอร์ มีความสอดคล้องกันใกล้ที่สุด การทำงานของระบบบังคับความเร็ว สามารถจะตอบสนองได้ทั้งสัญญาณคำสั่งบังคับให้ความเร็วคงที่ หรือให้ความเร็วแปรค่าไปได้ซึ่งการแปรไปของคำสั่งบังคับนั้นจะต้องอยู่ภายในแบนด์วิดท์ของระบบงานเพื่อการศึกษาเท่านั้น ไม่นุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ - บั๊ลลือคือได้อะแกร์ม์ของระบบ

บล็อกไดอะแกรมซึ่งอธิบายถึง ระบบการบังคับความเร็ว แสดงได้ดังในรูป

ที่ 2.12



รูปที่ 2.12 ระบบบังคับความเร็ว

เมื่อ V_{in} = สัญญาณคำสั่งบังคับอินพุท

V_x = สัญญาณป้อนกลับ

$e = (V_{in} - V_x) =$ สัญญาณเออร์เรอร์

สัญญาณคำสั่งบังคับอินพุท จะได้รับการเปรียบเทียบกับ สัญญาณป้อนกลับความแตกต่างของค่าทั้งสองได้เป็นสัญญาณเออร์เรอร์ป้อนผ่านแอมพลิไฟไปยังมอเตอร์ ซึ่งเป็นผลให้เออร์เรอร์ของความเร็วจะได้รับการแก้ไขให้ถูกต้อง ส่วนประกอบอันแรกของลูป ได้แก่ แอมพลิไฟที่เป็นแบบลิเนียร์แอมพลิไฟ หรือพัลซ์วิธโมดูละชันแอมพลิไฟ นอกจากนั้นแอมพลิไฟอาจจะเป็นตัวกำเนิดกระแสหรือตัวกำเนิดโวลต์เต็ม ทั้งสองกรณีเราจะกำหนดให้แอมพลิไฟมีทรานเฟอร์ฟังก์ชันเป็น $A(s)$ ส่วนประกอบที่สองในลูป ได้แก่ มอเตอร์ และแทนได้ด้วยทรานเฟอร์ฟังก์ชันเป็น $M(s)$ ส่วนประกอบที่สามของลูปความเร็ว คือ ตัวเซ็นเซอร์ความเร็วซึ่งจะเป็นตัวกำเนิดสัญญาณอย่างต่อเนื่อง และมีแมกนิจูดของสัญญาณ เป็นสัดส่วนของความเร็วของมอเตอร์ ในกรณีทั่ว ๆ ไป ตัวเซ็นเซอร์นี้ ได้แก่ ทาโคมิเตอร์ อย่างไรก็ตาม การใช้ทาโคมิเตอร์ในลูปของการป้อนกลับจำเป็นต้องพิจารณาถึงข้อเสียบางประการคือ เรื่องราคาและขนาดของระบบ เราสามารถสร้างสัญญาณป้อนกลับของความเร็วได้เหมือนกับสัญญาณที่ได้จากทาโคมิเตอร์ โดยการรวมของสัญญาณกระแสและสัญญาณโวลต์เต็มวิธีนี้อธิบายได้จากโมเดลง่าย ๆ ของมอเตอร์โวลต์เต็มซึ่งกำหนดได้โดยสมการ (2-1)

$$v_1(t) = R_u i_u(t) + K_u \omega(t) \tag{2-1}$$

เมื่อ K_u คือ ค่าคงที่ของโวลต์เสด็จย้อนกลับ

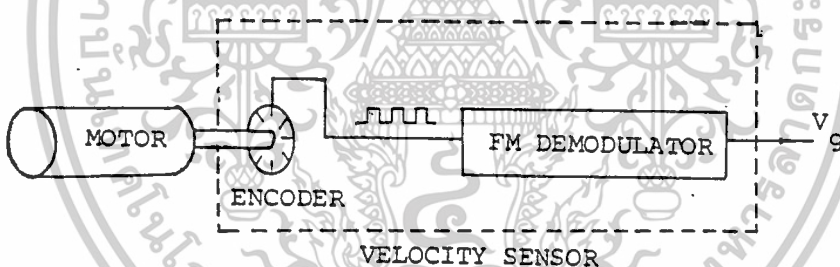
จากสมการ (2-1) เราจะหาค่าความเร็วของมอเตอร์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอก $1/K_u$ หรือ $[V_1(t) - R_u i_u(t)]$ ของเอกสารทุกครั้งที่มีการนำ (2-2)

วิธีที่ได้กล่าวแล้วอธิบายถึง หลักการสร้างสัญญาณป้อนกลับของความเร็ว โดยไม่ได้ใช้ทาโคมิเตอร์ วงจรในรูป 2.13 อาจทำให้ง่ายขึ้น โดยใช้ดิฟเฟอเรนเชียล แอมพลิไฟเพียงตัวเดียว

ปัญหาหลักในการใช้วิธีนี้สร้างสัญญาณป้อนกลับความเร็วก็คือ ความต้านทานของมอเตอร์จะเปลี่ยนแปลงไปตามอุณหภูมิ ดังนั้นสมการ(2-6) จึงไม่คงสภาพความเที่ยงตรงด้วยเหตุผลดังกล่าว เราจะต้องพิจารณามอเตอร์ ในลักษณะโมเดลที่ซับซ้อนยิ่งขึ้น โดยต้องคิดรวมถึงผลของอาร์มาร์เจอร์อินดักแตนซ์ด้วย เพื่อพัฒนาการออกแบบให้ได้ผลดียิ่งขึ้น

นอกจากนี้ยังมีอีกวิธีหนึ่งสำหรับวัดความเร็วของมอเตอร์โดยไม่ต้องใช้ทาโคมิเตอร์คือ การสร้างสัญญาณความเร็วจากเอาต์พุทของเอนโคคเตอร์ของเพลามอเตอร์ ซึ่งมีอยู่แล้วในลูฟของระบบ เพื่อใช้วัดตำแหน่งของโพล เอ็นโคคเตอร์ของเพลามอเตอร์จะสร้างสัญญาณที่มีความถี่เป็นสัดส่วนกับความเร็วเชิงมุมของมอเตอร์ หรืออาจพูดได้ว่าเอาต์พุทของเอนโคคเตอร์ คือ สัญญาณความถี่ ซึ่งได้รับจากโมดูลด้วยความเร็วของมอเตอร์ อย่างไรก็ตาม ถ้าเราต้องการสร้างความเร็วของมอเตอร์ขึ้นใหม่อีกเราจะต้องดีโมดูลความถี่เอาต์พุทของเอนโคคเตอร์ ซึ่งสามารถกระทำได้โดยใช้เอฟเอ็มดีโมดูลเตอร์ ดังแสดงในรูปที่ 2.14 หรือโดยต่อรวมกับวงจรพิเศษจำพวกแปลงความถี่เป็นโวลต์เตจ (F-V Converter)



รูปที่ 2.14 แสดงการวัดความเร็วของมอเตอร์

ระบบควบคุมตำแหน่ง

ระบบบังคับการเคลื่อนที่ ส่วนมากมักต้องการหมุนโพลด์ให้ไปหยุดในตำแหน่งที่กำหนด และยึดตำแหน่งนั้นไว้จนกว่าจะได้รับคำสั่งบังคับการหมุนอันใหม่ การเคลื่อนที่อีกแบบหนึ่งของระบบนี้คือ ในบางครั้งต้องการให้โพลด์เคลื่อนที่ไปตามระยะตำแหน่งที่กำหนดให้ข้อกำหนดการเคลื่อนที่ทั้งสองแบบดังกล่าว สามารถกระทำได้โดยอาศัยระบบการบังคับตำแหน่ง

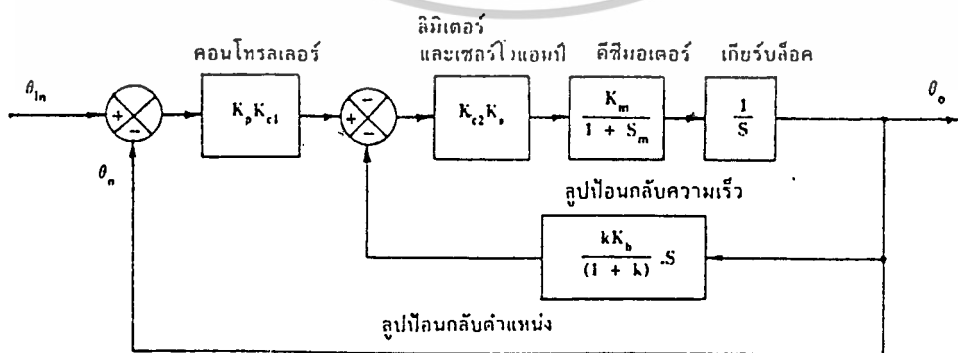
ระบบการบังคับตำแหน่ง ตำแหน่งเชิงมุมของมอเตอร์ที่ต่ออยู่กับโพลด์จะสอดคล้องกับคำสั่งอินพุท เมื่อสัญญาณบังคับคงที่เพลลาของมอเตอร์ก็จะล๊อคกับตำแหน่งที่ต้องการ เมื่อสัญญาณบังคับแปรไปอย่างต่อเนื่อง ตำแหน่งของมอเตอร์ก็จะเคลื่อนตามการบังคับที่ราบ

เท่าที่การแปรไปของสัญญาณบังคับอยู่ในแบนด์วิธของลูฟ

ตัวเซนเซอร์ตำแหน่ง อาจเป็นพวกที่ให้สัญญาณเอาท์พุทได้ต่อเนื่อง เช่น โป-เทนทิโอมิเตอร์ ซึ่งระบบการบังคับแบบนี้จะเป็นไปอย่างต่อเนื่อง หรือเป็นระบบการบังคับแบบอนาลอก ระบบการบังคับตำแหน่งแบบอนาลอกมีความละเอียดในการบังคับตำแหน่งได้ดีเยี่ยม ซึ่งเป็นคุณสมบัติที่เหมาะสมในการบังคับตำแหน่ง อย่างไรก็ตาม สัญญาณของตำแหน่งที่มีการแปรอย่างต่อเนื่องจะถูกจำกัดอยู่ในช่วงตำแหน่งช่วงหนึ่งเท่านั้น และการลิมิตนี้จะเป็นตัวจำกัดการเคลื่อนที่

ตัวเซนเซอร์ตำแหน่งแบบอื่น ได้แก่ เซนเซอร์แบบดิจิตอล คือ พวกเอนโคเดอร์สำหรับตำแหน่งของเพลลา พวกเซนเซอร์แบบนี้มีช่วงการทำงานของตำแหน่ง ไม่มีขีดจำกัด อย่างไรก็ตาม เซนเซอร์พวกนี้มีข้อเสียในเรื่องความละเอียดของการบังคับตำแหน่ง อยู่ภายในค่าที่จำกัด นอกจากนี้สัญญาณของตำแหน่ง จะอยู่ในลักษณะของสัญญาณดิจิตอล สัญญาณนี้จะต้องป้อนเข้า D-A คอนเวอร์เตอร์ก่อน เพื่อแปลงไปเป็นสัญญาณของตำแหน่งแบบอนาลอก มีหลักความเป็นจริงอยู่ว่า เราไม่สามารถจะบังคับตำแหน่งเอาท์พุทของระบบให้เที่ยงตรงได้มากไปกว่าความเที่ยงตรงที่วัดได้จากตัวเซนเซอร์

ปัญหาหลักในการออกแบบระบบจากบังคับตำแหน่ง คือ การทำให้ระบบมีเสถียรภาพได้อัตราขยายที่เหมาะสมและมีแบนด์วิธของลูฟอย่างพอเพียง ดังนั้นเราจะได้นั้นถึงปัญหาของเสถียรภาพและหลักการออกแบบต่อไป ระบบการบังคับแบบป้อนกลับ โดยทั่วไปจุดมุ่งหมายหลัก คือ ต้องการให้ระบบมีความเที่ยงตรงที่สภาวะสงบนิ่ง (Steady-State Accuracy) เป็นที่พอใจ ดังนั้นระบบจึงไม่นิยมที่จะให้มีการลดอัตราขยายลง เพื่อทำให้ระบบมีเสถียรภาพ ถ้าอัตราการขยายที่ต้องการเพื่อกำหนดความเที่ยงตรงทำให้ระบบไม่มีเสถียรภาพหรือผลตอบสนองมีการแกว่งที่เร็ว เมื่อนั้นระบบจะต้องได้รับการเปลี่ยนแปลง (มีการชดเชย) ในวิถีทางที่จะให้ได้ผลตอบสนองตามที่ต้องการ โดยไม่ทำให้ความเที่ยงตรงด้อยลงไป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งรูปที่ 2-15 ระบบควบคุมตำแหน่งหมุน เอกสารทุกครั้งที่มีการนำไปใช้

2.3 การคำนวณหาขนาดของมอเตอร์

$$\begin{aligned} \text{กำลังของเครื่องยนต์ } P &= 2\pi TeN/60 \\ Te &= mgr/G \\ N &= VG/2\pi r \end{aligned}$$

แทนค่า Te และ N

$$\begin{aligned} P &= (2\pi/60)(mgr/G)(VG/2\pi r) \\ &= mgV/60 \end{aligned}$$

P	:	กำลังของเครื่องยนต์	watt
Te	:	แรงบิดของเครื่องยนต์	N
N	:	ความเร็วของเพลาข้อเหวี่ยง	rev/min
V	:	ความเร็วของล้อ	m/min
G	:	อัตราทดเฟืองทั้งหมด	
r	:	รัศมีของล้อ	m

เช่น ถ้า ต้องการความเร็ว 15 cm/sec ซึ่งก็เท่ากับ 9 m/min และต้องการให้รับน้ำหนักได้ประมาณ 50 กิโลกรัม ดังนั้น มอเตอร์ที่ต้องการจะต้องมีขนาด

$$\begin{aligned} P &= 50 \cdot 10 \cdot 9 / 60 \\ &= 75 \text{ watt} \end{aligned}$$

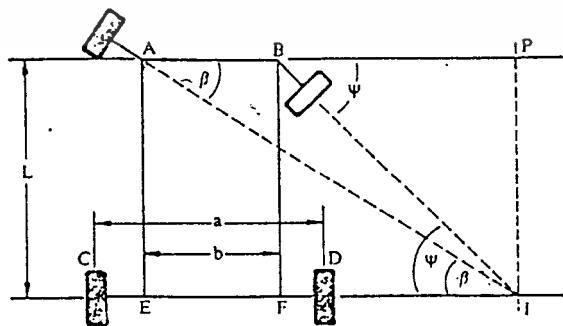
2.4 การขับเคลื่อนและการเลี้ยว

Mobile Robot ส่วนใหญ่ใช้พลังงานในการขับเคลื่อนจาก แบตเตอรี่ ชนิด ตะกั่ว-กรด(Lead-Acid) และขับเคลื่อนโดยใช้มอเตอร์กระแสตรง ซึ่งมีการควบคุมทิศทาง การหมุน, ความเร็วและตำแหน่งโดยใช้คอมพิวเตอร์(Onboard Computer) สำหรับ เทคนิคการเลี้ยวของ Mobile Robot นั้นจะขึ้นกับโครงสร้างระบบล้อ ระบบล้อตามปกติ ที่ใช้กันอาจเป็นระบบ 3 ล้อ หรือระบบ 4 ล้อก็ได้ Mobile Robot บางตัวอาจใช้ถึง 6 ล้อ ทั้งนี้ขึ้นอยู่กับผู้ผลิต เมื่อเปรียบเทียบระบบ 3 ล้อกับระบบ 4 ล้อแล้ว ระบบ 4 ล้อจะมี เสถียรภาพดีกว่า, กำลังลากจูงดีกว่า(More Traction), ช่องว่างทางกลน้อยกว่า แต่ ระบบ 3 ล้อก็มีข้อดีตรงที่ระบบเลี้ยวไม่ยุ่งยากซับซ้อน, โครงสร้างมีน้ำหนักเบากว่า ซึ่งทำให้ ประหยัดพลังงานจากแบตเตอรี่ได้มาก สำหรับโครงสร้างระบบล้อของ Mobile Robot นั้นมีหลายแบบ อาทิเช่น

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ไม่ว่ากรณีใดๆ ห้ามระบบ 3 ล้อที่เลี้ยวด้วย 2 ล้อหลัง จนถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ระบบ 3 ล้อ ล้อหน้าเป็นล้อขับและเลี้ยว ส่วนล้อหลังล้อช่วย (Caster)
- ระบบ 3 ล้อ เลี้ยวด้วยล้อหน้า 2 ล้อ โดยมีทั้งแบบเลี้ยวพร้อมกัน หรือแต่ละล้อเลี้ยวโดยอิสระ ส่วนล้อหลังเป็นล้อขับ
- ระบบ 3 ล้อ เลี้ยวโดยใช้หลักการหมุนในทิศทางตรงข้ามกัน (Differential Steering) ของล้อหลัง 2 ล้อ ส่วนล้อหน้าก็สามารถเลี้ยวได้ด้วย
- ระบบ 3 ล้อ เลี้ยวโดยใช้หลักหมุนในทิศทางตรงข้ามกันของ 2 ล้อ ส่วนล้อหน้าเป็นล้อช่วย
- ระบบ 3 ล้อ ล้อทุกล้อเป็นล้อขับที่สามารถเลี้ยวได้โดยอิสระ เทคนิคนี้มีความยืดหยุ่นสูงแต่โครงสร้างทางกลซับซ้อน
- ระบบ 4 ล้อ เลี้ยวโดยใช้หลักหมุนในทิศทางตรงข้ามกันของ 2 ล้อขับ อีก 2 ล้อเป็นล้อช่วย
- ระบบ 4 ล้อ ขับเคลื่อนได้ทั้ง 4 ล้อ และล้อแต่ละคู่ก็สามารถเลี้ยวได้ด้วย
- ระบบ 4 ล้อ 2 ล้อหน้าเป็นล้อช่วย ส่วน 2 ล้อหลังเป็นทั้งล้อขับและล้อเลี้ยว ในโครงการนี้เลือกใช้ระบบนี้
- ระบบ 4 ล้อ 2 ล้อหน้าสำหรับเลี้ยว ส่วน 2 ล้อหลังเป็นล้อขับ หลักการเดียวกับรถยนต์ทั่วไปที่ขับเคลื่อนล้อหลัง
- ระบบ 5 ล้อ ล้อที่ 5 ซึ่งเป็นล้อขับ และล้อสำหรับเลี้ยว จะติดตั้งอยู่ตรงกลางแนวแกนของ 2 ล้อหน้าซึ่งเป็นล้อช่วย ส่วน 2 ล้อหลังก็เป็นล้อช่วยธรรมดาเช่นกัน
- ระบบ 6 ล้อ ล้อที่ 5 และล้อที่ 6 ซึ่งเป็นล้อขับและล้อสำหรับเลี้ยวจะติดตั้งอยู่ตรงกลางแนวแกน 2 ล้อหน้าซึ่งเป็นล้อช่วย และ 2 ล้อหลังซึ่งเป็นล้อช่วยเช่นกันตามลำดับ

การบังคับเลี้ยวของรถยนต์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งรูปที่ 2.16 ลักษณะการเลี้ยวของรถยนต์ เอกสารทุกครั้งที่มีการนำไปใช้

เมื่อรถยนต์วิ่งไปบนถนนจะเห็นรอยล้อที่เสียดสีกันระหว่างยางกับถนน การกลิ้งตัวของยางบนผิวถนนนั้นจะพิจารณาได้จากการเขียนวงกลมที่จุดศูนย์กลางร่วมกัน ในขณะที่รถเลี้ยวแกนล้อ (Stub Axle) ของล้อจะเอียงตัวไปตามทิศทางของการเลี้ยว แนวแกนล้อทั้งสองที่หันเลี้ยวจะพบกันที่จุดศูนย์กลางของการเลี้ยว โดยจุดนี้จะอยู่ในแนวเดียวกับเพลาลัง จากรูปที่ 2.16 จะเห็นได้ว่าแกนล้อด้านในของการเลี้ยวเอียงทำมุมกับแนวเส้นเพลาเป็นมุม ψ และมีขนาดกว้างกว่ามุม β ของแกนล้อด้านนอกของการเลี้ยว

กำหนดให้ $a = CD$ คือ ช่วงกว้างระหว่างล้อ
 $b = AB$ คือ ระยะระหว่างจุดหมุนของเพลาน้ำ
 $L = AE$ คือ ช่วงยาวระหว่างล้อ
 I คือ จุดศูนย์กลางการเลี้ยวของล้อทั้ง 4

ถ้าลากเส้น IP ตั้งฉากกับ AB ที่จุด P ดังรูปที่ 4.3

$$\begin{aligned} b &= AP - BP \\ &= L \cot \beta - L \cot \psi \end{aligned}$$

$$\text{นั่นคือ } \cot \beta - \cot \psi = b/L$$

สูตรนี้เป็นสูตรเบื้องต้นถ้าหากล้อไม่ลื่นไถล อัตราส่วน b/L ก็จะเป็นไปตามสมการ

2.5 การตรวจจับสิ่งกีดขวาง

สำหรับ Mobile Robot ที่ใช้กันโดยทั่วไปนั้น จะต้องมี การป้องกันการชนกับสิ่งต่าง ๆ ในระหว่างการเคลื่อนที่ทั้งนี้เพื่อความปลอดภัยในการทำงาน เทคนิคที่ใช้ในการตรวจจับวัตถุกีดขวางมีหลายวิธี อาทิ เช่น การใช้กล้องวิดีโอ , การใช้อัลตราโซนิก เป็นต้น สำหรับโครงการนี้ได้เลือกใช้อัลตราโซนิก เนื่องจากมีราคาถูกลงกว่า, มีขนาดเล็กและน้ำหนักเบาตลอดจนวงจรที่ไม่ยุ่งยากและซับซ้อน การนำอัลตราโซนิกทรานสดิวเซอร์มาใช้ในการตรวจจับวัตถุกีดขวาง จะใช้หลักการสะท้อนกลับของคลื่นอัลตราโซนิก โดยในระหว่างรอจะมีการนับจำนวนลูกคลื่นความถี่ด้วยตัวนับ (Counter) สำหรับความถี่ที่นับนั้นจะใช้ความถี่ที่ตรงกับระยะทางที่เป็นจริง ถ้าวัตถุกีดขวางอยู่ใกล้จำนวนลูกคลื่นความถี่ที่นับได้ก็จะน้อย แต่ถ้าวัตถุกีดขวางอยู่ไกล จำนวนลูกคลื่นความถี่ที่นับได้ก็จะมาก สำหรับความถี่ของคลื่นอัลตราโซนิกจะใช้ที่ 40 KHz เนื่องจากเป็นค่าที่เหมาะสมที่สุด สำหรับทฤษฎีที่เกี่ยวข้องกับอัลตราโซนิกทรานสดิวเซอร์นั้นมีดังนี้

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คลื่นอุลตราโซนิค

โดยทั่วไปมนุษย์จะได้ยินความถี่เสียงสูงสุดประมาณ 15-16 KHz ปกติแล้ว คำว่าอุลตราโซนิคมักจะหมายถึงเสียงที่มีความถี่สูงกว่า 20 KHz ขึ้นไปเช่น ย่าน 20 KHz 25 KHz 31.5 KHz 40 KHz 45 KHz เป็นต้น คลื่นอุลตราโซนิคมีคุณสมบัติแตกต่างจากคลื่นเสียงธรรมดาตรงที่คลื่นเสียงธรรมดาตามนุษย์สามารถได้ยิน และมีลักษณะการเคลื่อนที่กระจายไปทุกทิศทาง ส่วนคลื่นอุลตราโซนิคนั้นจะมีลักษณะการเคลื่อนที่เป็นแนวเส้นตรง คล้ายกับการเคลื่อนที่ของแสง คุณสมบัติอย่างหนึ่งของคลื่นก็คือ ถ้าคลื่นมีความถี่สูงขึ้นความยาวคลื่นก็จะสั้นลง ถ้าความยาวคลื่นยาวกว่าช่องเปิดที่ให้เสียงนั้นออกมาของตัวที่กำเนิดเสียงความถี่นั้น คลื่นจะเกิดหักเหที่ขอบด้านนอกของตัวกำเนิดคลื่นเสียงทำให้เกิดการกระจายคลื่น แต่ถ้าความถี่สูงขึ้นมาอยู่ในย่านอุลตราโซนิคเช่น 40 KHz ความยาวคลื่นจะสั้นลง คลื่นเสียงก็จะมีลักษณะพุ่งออกมาเป็นลำ เรียกว่า มีทิศทางนั่นเอง

การมีทิศทางของคลื่นอุลตราโซนิค ทำให้เราสามารถนำเอาไปใช้งานได้หลายอย่าง เช่น นำไปใช้ในเครื่องควบคุมระยะไกล (Ultrasonic Remote Control) เครื่องล้างอุปกรณ์ (Ultrasonic Cleaner) โดยทำน้ำสั่นที่ความถี่สูง เครื่องวัดความหนาของวัตถุ เครื่องวัดความลึกและทำแผนที่ใต้ท้องทะเล เครื่องทดสอบการรั่วของท่อ และการใช้งานทางการแพทย์ เป็นต้น ความถี่ที่ใช้ก็มักจำกัดอยู่เพียงไม่เกิน 50 KHz เพราะความถี่สูงกว่านี้อากาศจะดูดกลืนคลื่นเสียงได้มากขึ้น ทำให้ระดับความแรงของคลื่นเสียงที่ระยะต่างออกไปลดลงอย่างรวดเร็ว คลื่นอุลตราโซนิค 40 KHz จะเป็นค่าที่เหมาะสมที่สุดที่จะนำมาใช้งาน ส่วนระยะทางสูงสุดก็ประมาณ 20 เมตร คลื่นจะเคลื่อนที่ไปได้มากน้อยเท่าใดขึ้นอยู่กับอุปกรณ์ที่ใช้สร้างและสิ่งแวดล้อมต่าง ๆ ด้วย

ชนิดของคลื่นอุลตราโซนิค

คลื่นอุลตราโซนิคที่เดินทางผ่านตัวกลางต่าง ๆ มีหลายชนิดด้วยกัน แต่ละชนิดจะแตกต่างกันตามการเคลื่อนที่ของอนุภาคในตัวกลางนั้น ชนิดของคลื่นอุลตราโซนิคมีดังนี้

ก) คลื่นตามยาว (Longitudinal Wave)

ทุก ๆ จุดบนคลื่นจะมีการเคลื่อนที่ในทิศทางเดียวกับการเคลื่อนที่ คลื่นตามยาวนี้สามารถเดินทางผ่านของแข็ง, ของเหลวและก๊าซ โดยมากถ้ากล่าวถึงความเร็วของคลื่นเสียงแล้วจะหมายถึงความเร็วของคลื่นตามยาว ในการพิจารณาคลื่นตามยาวที่เดินทางผ่านตัวกลางต่าง ๆ ได้นั้นตัวกลางจะต้องมีขนาดใหญ่พอเมื่อเทียบกับความยาวคลื่น

ข) คลื่นตามขวาง (Transverse Wave)

ทุก ๆ จุดบนคลื่นจะมีการเคลื่อนที่ในทิศทางตั้งฉากกับทิศทางการเคลื่อนที่ของคลื่น คลื่นชนิดนี้จะเดินทางผ่านตัวกลางที่มีขนาดของตัวกลางใหญ่กว่าขนาดของความยาวคลื่น สามารถเดินทางผ่านตัวกลางที่เป็นของแข็งหรือไม่สามารถเดินทางผ่านตัวกลางที่เป็น

ของเหลวและก๊าซได้ ความเร็วของคลื่นตามขวางจะมีน้อยกว่าคลื่นตามยาว ในขณะที่ผ่านตัวกลางชนิดเดียวกัน

ค) คลื่นผิวหน้า (Surface Wave or Rayleigh)

คล้ายคลื่นตามขวาง แต่จะต่างกันตรงที่ว่า การเปลี่ยนแปลงตำแหน่งของอนุภาคไม่เป็นไปเพียงทิศทางที่ตั้งฉากกับทิศทางของการเคลื่อนที่เพียงอย่างเดียว แต่มีการเปลี่ยนแปลงในทิศทางเดียวกับทิศทางของการเคลื่อนที่ด้วย จึงทำให้คลื่นเคลื่อนที่ไปตามระนาบในแนวนอน ด้วยเหตุนี้คลื่นจึงเดินทางไปเฉพาะบนผิวของตัวกลางเท่านั้น

อูลตราโซนิคทรานสดิวเซอร์ (Ultrasonic Transducer)

เป็นอุปกรณ์ที่สามารถแปลงพลังงานในรูปอื่นให้มาเป็นพลังงานทางกลโดยการสั่นไปมา ทำให้เกิดคลื่นเสียงย่านอูลตราโซนิคกระจายไปในอากาศได้

ในปัจจุบัน อูลตราโซนิค ทรานสดิวเซอร์ มีหลายแบบขึ้นอยู่กับหลักการใช้แบบที่นิยมใช้กันมาก ได้แก่ แบบเพียโซอิเล็กทริก (Piezoelectric) ซึ่งแปลงมาระหว่างพลังงานพลังงานไฟฟ้าและพลังงานกลโดยมีถี่เรโซแนนซ์ (Resonance) ค่าหนึ่ง แบบแมกนีโตสตริกทีฟ (Magnetostrictive) ซึ่งแปลงไปมาระหว่างพลังงานไฟฟ้ากับขดลวดกับตำแหน่งความยาวของแกนเหล็กที่สวมขดลวดนั้นอยู่ และแบบอิเล็กโตรสตริกทีฟ (Electrostrictive) ซึ่งแปลงไปมาระหว่างพลังงานไฟฟ้ากับพลังงานกล

สำหรับอูลตราโซนิคทรานสดิวเซอร์ที่ใช้ในงานวิจัยนี้เป็น แบบเพียโซอิเล็กทริก ซึ่งจะมีอยู่ 2 อย่างคือ ตัวส่ง (Transmitter) และตัวรับ (Receiver) ซึ่งตัวส่งเป็นอูลตราโซนิคทรานสดิวเซอร์ที่ออกแบบให้แปลงสัญญาณไฟฟ้าที่ให้ตัวมันเป็นคลื่นเสียงย่านอูลตราโซนิคหน้าที่ของตัวส่งจะคล้ายลำโพง ส่วนตัวรับเป็นอูลตราโซนิคทรานสดิวเซอร์ ที่ออกแบบให้แปลงคลื่นเสียงย่านอูลตราโซนิคที่มากกระทบตัวมันให้ออกมาเป็นสัญญาณไฟฟ้า หน้าที่ของตัวรับจึงคล้ายไมโครโฟน

2.6 แบตเตอรี่

แบตเตอรี่มี 5 ชนิดแต่ละชนิดมีรูปร่าง , โครงสร้าง และขนาดต่างกัน

- 1) ซิงค์ (Zinc) ไม่นิยมใช้ในหุ่นยนต์เพราะกำลังงานหมดเร็ว
- 2) อัลคาไลน์ (Alkaline) ราคาแพงกว่าซิงค์ พลังงานตลอดจนอายุการใช้งานนานกว่า แต่ข้อเสียของแบตเตอรี่คือ ไม่สามารถบรรจุใหม่ได้ (Recharge) ได้
- 3) นิกเกิล-แคดเมียม (Nickel-Cadmium) หรือเรียกว่า NI-CAD สามารถประจุใหม่ได้หาง่าย ราคาไม่แพงนัก แต่มีกำลังงานน้อย เหมาะใช้ในหุ่นยนต์ขนาดเล็กใช้งานได้ไม่นานเหมือนซิงค์ และอัลคาไลน์ ในปัจจุบันมีถ่านนิกเกิล-แคดเมียมความจุสูงกว่าประมาณ 2-3 เท่าซึ่งเหมาะกับงานด้านนี้แต่ราคาไม่แพง เจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) ตะกั่ว-กรด มีเห็นใช้ทั่วไปในรถยนต์ รถมอเตอร์ไซด์ สามารถประจุใหม่ได้ มีทั้งขนาดเล็กและขนาดใหญ่ มีกำลังงานสูงแต่หนักมาก ทำให้ robot ต้องแบกน้ำหนักเพิ่มขึ้น นอกจากนี้เราจะเห็นว่ามีการใช้แบตเตอรี่ตะกั่ว-กรดกับระบบจ่ายไฟไม่ขาดตอน (UPS) สำหรับระบบคอมพิวเตอร์หรือระบบโทรศัพท์ สำหรับ Robot ที่ใช้กันอยู่ โดยมากมักจะใช้แบตเตอรี่แบบนี้

5) เจล-เซลล์ (Gel-Cell) ประจุใหม่ได้ คล้ายตะกั่วกรดลิธแอลิต แต่กระจายกระแสไฟสูงกว่า

ข้อกำหนดของแบตเตอรี่

1) แรงดัน สำหรับแบตเตอรี่ที่ประจุใหม่ได้นั้น ขณะประจุเต็มจะมีแรงดันสูงกว่าระดับปกติที่กำหนดประมาณ 20-30 % เช่น แบตเตอรี่ 12 โวลต์ ชนิดตะกั่วกรด เมื่อประจุเต็มจะมีค่าประมาณ 13.8 โวลต์ แบตเตอรี่ทั้งหมดจะถือว่าหมดสภาพ เมื่อแรงดันลดต่ำลงเหลือประมาณ 80 % จากระดับปกติ เช่น แบตเตอรี่ 6 โวลต์ จะถือว่าหมดสภาพถ้าแรงดันตกต่ำกว่า 4.8 โวลต์ ขณะที่แบตเตอรี่หมดสภาพ มันจะไม่สามารถจ่ายไฟได้ตามอัตราปกติต่อไป การทดสอบแรงดันแบตเตอรี่จะต้องทำขณะที่แบตเตอรี่ใช้งานอยู่ การทดสอบจะคลาดเคลื่อน ถ้าขณะทดสอบไม่มีโหลด

2) กระแส-เวลา (Ampere-Hour) แสดงถึงปริมาณความจุของแบตเตอรี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การวิเคราะห์และออกแบบส่วนต่าง ๆ ในโครงการ

3.1 การตรวจจับสิ่งกีดขวางโดยคลื่นอัลตราโซนิค

ในโครงการนี้ได้ใช้คลื่นอัลตราโซนิคในการตรวจจับวัตถุและวัดระยะของวัตถุ นั้น ๆ โดยอาศัยหลักการสะท้อนของคลื่นอัลตราโซนิค บนตัวรถของโครงการนี้จะมีการติดตั้งวงจรส่วนนี้ทั้งหมด 3 ชุด โดยติดตั้งไว้ที่ด้านหน้าและด้านข้าง 2 ข้าง เพื่อที่จะสามารถตรวจจับและวัดระยะห่างของวัตถุทั้ง 3 ด้าน แล้วนำข้อมูลที่ได้ไปใช้ควบคุมทิศทางการเคลื่อนที่ของตัวรถเพื่อหลบหลีกและเลือกเส้นทางเดินที่ถูกต้อง

เนื่องจากคลื่นเสียงเดินทางในอากาศด้วยความเร็วคงที่ และสามารถรู้ค่าความเร็วได้ ดังนั้นช่วงเวลาในการเดินทางจากเครื่องส่งไปยังวัตถุและสะท้อนกลับมาสามารถนำมาใช้คำนวณหาระยะทางได้ เมื่อเครื่องเริ่มส่งแถบคลื่นอัลตราโซนิคออกไป วงจรนับจะเริ่มทำการนับพัลส์ ซึ่งมีความถี่ที่เราป้อนให้แก่ตัวนับ เมื่อได้รับสัญญาณสะท้อนกลับ ชุดพัลส์ที่ส่งไปวงจรนับจะหยุดนับ ดังนั้นเพื่อให้ได้ระยะทางที่ถูกต้องเวลาที่ใช้ในการเดินทางของคลื่นเสียงจึงต้องถูกหารด้วยสอง

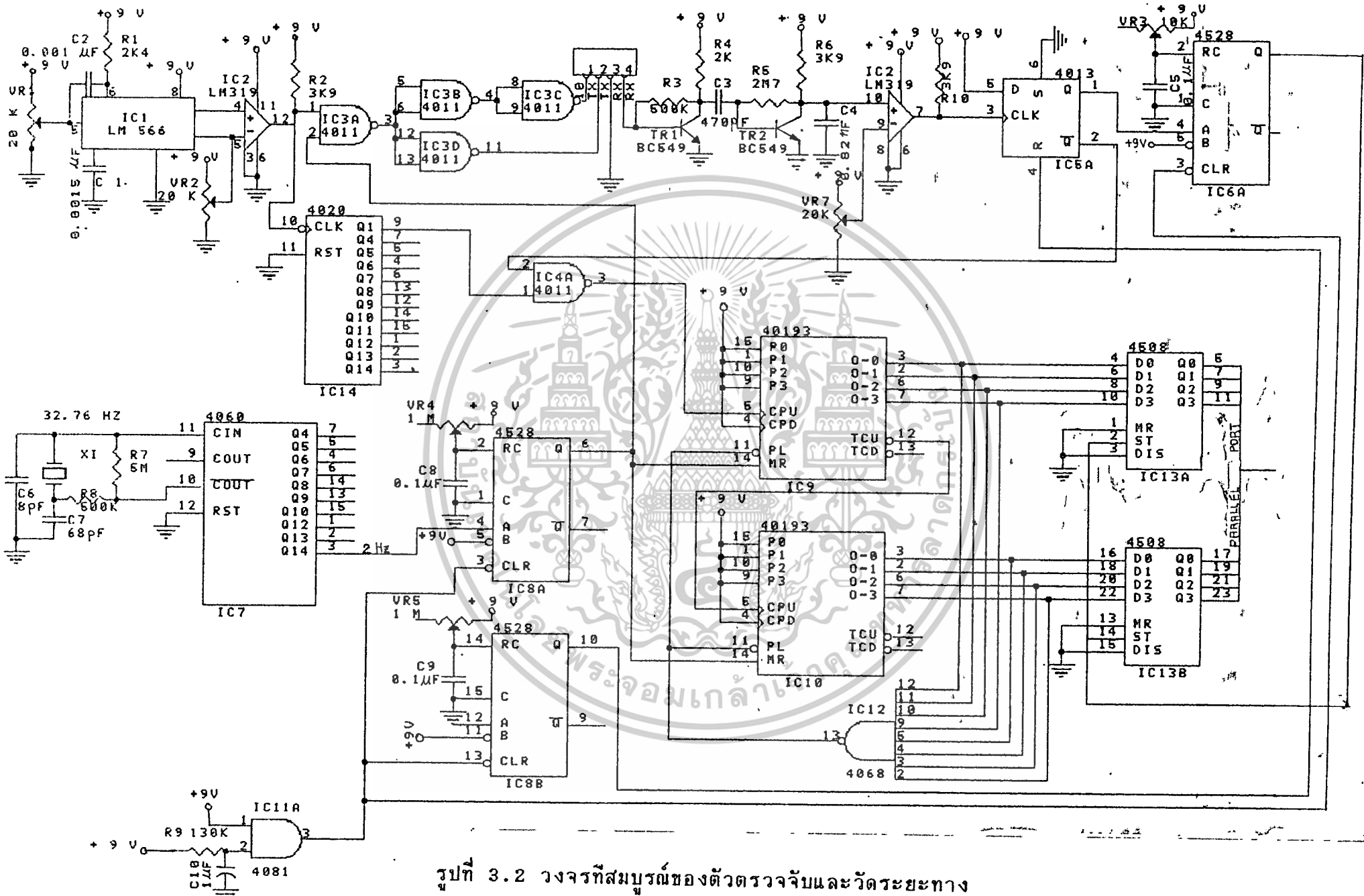
รูปที่ 3.1 แสดงแผนผังของตัวตรวจจับ และวัดระยะทาง ซึ่งประกอบด้วยเครื่องส่ง, เครื่องรับ, วงจรนับซึ่งสามารถแสดงค่าที่นับได้ ซึ่งวงจรนับจะทำงานและหยุดทำงานด้วยพัลส์ที่ได้จากเครื่องส่งและเครื่องรับและวงจรถ้าเน็ดความถี่ โดยจะมีการส่งคลื่นอัลตราโซนิคความถี่ 40 KHz จากตัวส่งทีละประมาณ 10-20 พัลส์ โดยใช้ความถี่ในการส่งเท่ากับ 2 Hz หรือส่งประมาณทุก ๆ 0.5 sec วงจรนับจะเริ่มนับตามสัญญาณพัลส์ของความถี่ที่กำหนดไว้ และเมื่อมีการสะท้อนกลับมายังตัวรับสัญญาณนั้นจะถูกขยาย และส่งไปกระตุ้นเพื่อทำการหยุดนับ สูตรการคำนวณหาจำนวนสัญญาณนาฬิกา

$$C = 2fm/v$$

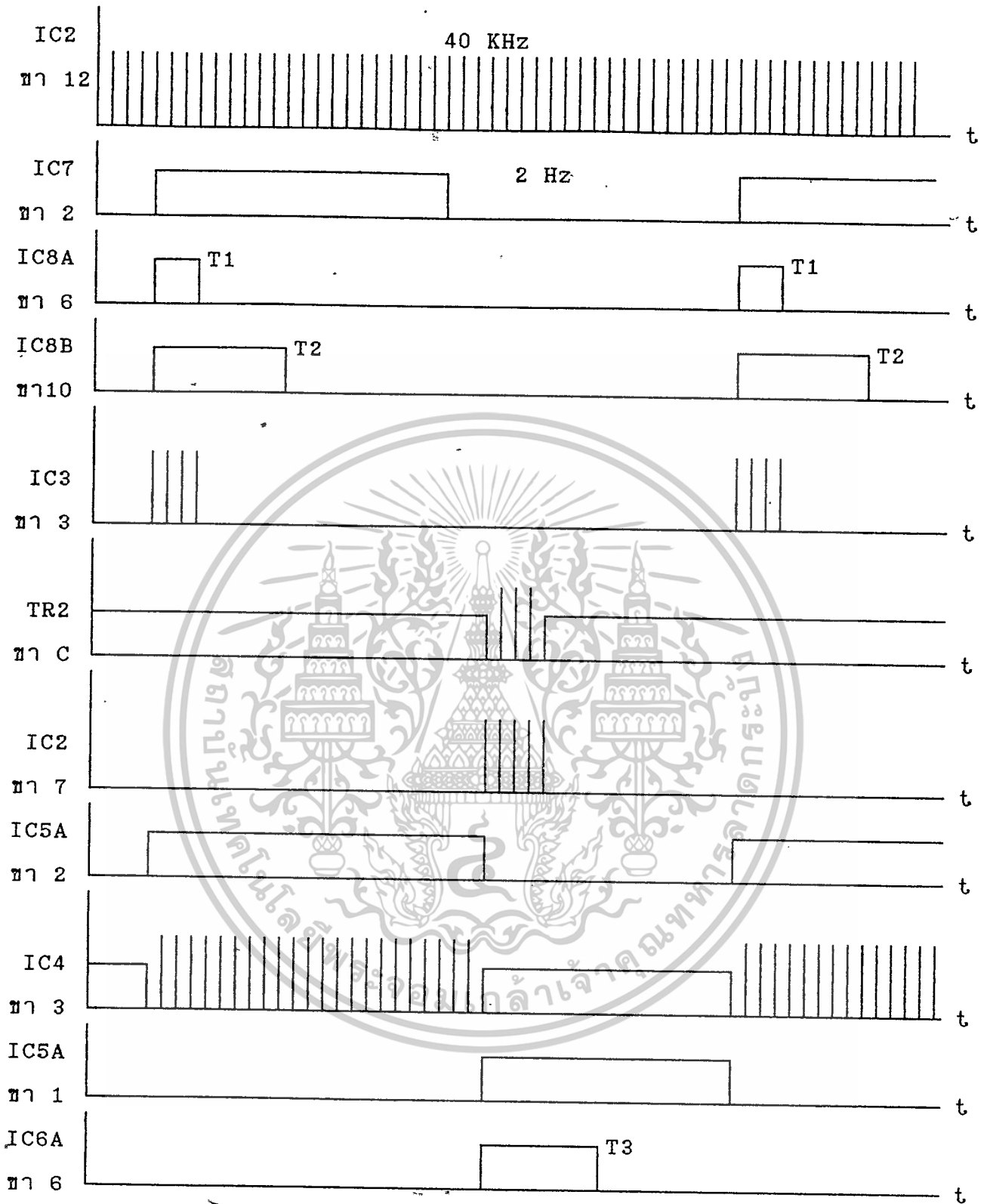
โดยที่

- C : จำนวนสัญญาณนาฬิกา (ลูก)
- f : ความถี่ที่ส่งให้วงจรนับ (Hz)
- m : ระยะทาง (m)
- v : ความเร็วของเสียง (m/s)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 วงจรที่สมบูรณ์ของตัวตรวจจับและวัดระยะเวลาทาง



รูปที่ 3.3 แผนผังเวลาแสดงรูปคลื่นตามจุดต่าง ๆ ในวงจร

Hz (IC7) จะไปกระตุ้นโมโนสเตเบิล 2 ตัวพร้อมกัน โดยที่โมโนสเตเบิลตัวแรก (IC8A) จะสร้างพัลส์ T1 ซึ่งสามารถกำหนดความกว้างของพัลส์ได้โดยการปรับ VR4 สัญญาณพัลส์ T1

มีหน้าที่ 2 อย่างคือ ส่งสัญญาณไปให้ส่วนเครื่องส่งส่งคลื่นออกไป และรีเซ็ต Counter (IC9, IC10) ให้พร้อมที่จะนับในครั้งต่อไป ส่วนโมโนสเตเบิลตัวที่สอง (IC8B) จะสร้างพัลส์ T2 ซึ่งสามารถกำหนดความกว้างของพัลส์ได้โดยการปรับ VR5 สัญญาณพัลส์ T2 มีหน้าที่คอยรีเซ็ต ดี-ฟลิปฟล็อป (IC5A) ซึ่งต่อขาข้อมูลไว้ที่ลอจิก "1" เมื่อมีการรีเซ็ต ดี-ฟลิปฟล็อป สัญญาณที่ขา Q จะไปทำให้ IC4A ปล่อยสัญญาณให้ counter นับทันที แต่อย่างไรก็ตาม จำนวนความถี่ที่นับได้จะถูกเก็บไว้ที่ตัวกักข้อมูล (Latch Data: IC13)

ส่วนที่เป็นเครื่องรับ เมื่อตัวรับคลื่นอนุตราโซนิคได้รับคลื่นที่สะท้อนกลับมา ก็จะทำการขยายสัญญาณโดยผ่านวงจรขยายที่ประกอบด้วยทรานซิสเตอร์ TR1, TR2 แล้วส่งเข้าขา CLK ของ ดี-ฟลิปฟล็อป ซึ่งจะส่งผลให้ขา Q ของดี-ฟลิปฟล็อปไปหยุดการนับของ Counter ขา Q ของดี-ฟลิปฟล็อปจะไปกระตุ้นให้โมโนสเตเบิล (IC6/1) สร้างสัญญาณพัลส์ที่มีความกว้าง T3 ทุกครั้งที่ขา Q ของดีฟลิปฟล็อปเปลี่ยนค่าจาก "0" เป็น "1" กล่าวคือเมื่อมีสัญญาณอนุตราโซนิคสะท้อนกลับมา พัลส์ T3 นี้จะมีหน้าที่อินเวิร์ตตัวกักข้อมูล (IC13) ให้ปล่อยข้อมูลออกไป ซึ่งข้อมูลที่ออกไปจะแสดงถึง จำนวนสัญญาณนาฬิกา ซึ่งต้องนำไปแปลงเป็นระยะทางดังสูตรที่ได้กล่าวมาแล้วข้างต้น

รูปที่ 3.3 แสดงแผนผังเวลาแสดงรูปคลื่นตามจุดต่าง ๆ ของวงจร เนื่องจากส่วนต่าง ๆ ของวงจรนี้ต้องอาศัยจังหวะการทำงานที่เหมาะสมจะถูกต้อง ดังนั้นในการปรับแต่งวงจรควรคำนึงถึงลักษณะของพัลส์และความถี่ที่ตำแหน่งต่าง ๆ ด้วย

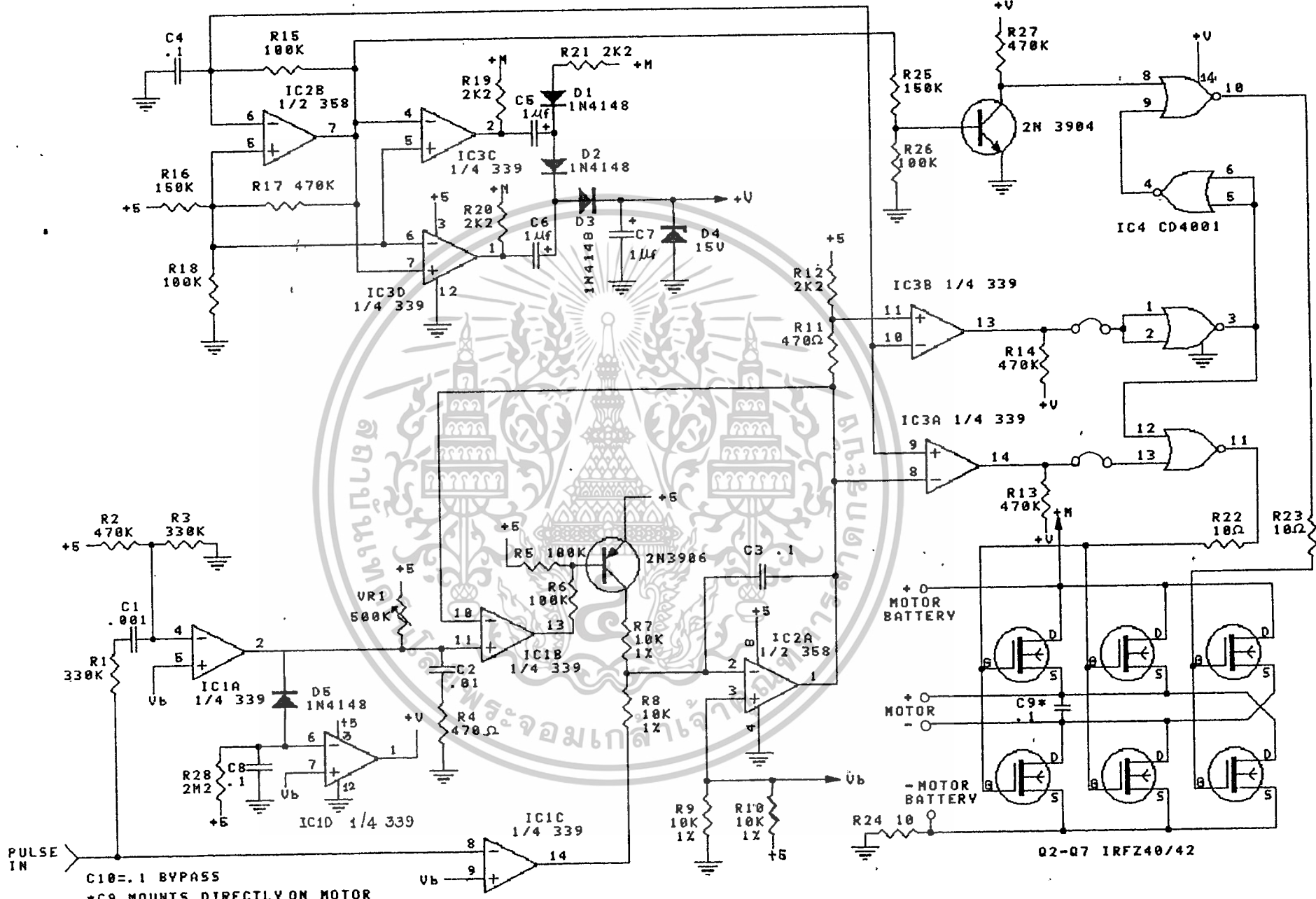
3.2 การควบคุมความเร็ว

ในโครงการนี้การควบคุมความเร็ว สามารถทำได้โดยการใช้สัญญาณพัลส์ป้อนให้กับ Speed Control ซึ่ง Speed Control จะทำการควบคุมโวลเตจที่จะป้อนให้กับมอเตอร์อีกทีหนึ่ง ความกว้างของสัญญาณพัลส์ที่แตกต่างกัน จะทำให้ล้อหมุนด้วยความเร็วและทิศทางที่แตกต่างกัน ซึ่งในโครงการนี้สัญญาณพัลส์กว้าง 1.5 มิลลิวินาที และมีคาบเวลา 19 มิลลิวินาที จะทำให้ล้อหยุดหมุน รถไม่เคลื่อนที่ แต่ถ้าสัญญาณพัลส์กว้างมากกว่า 1.5 มิลลิวินาที รถจะเคลื่อนที่ไปข้างหน้า และความเร็วเพิ่มขึ้น ถ้าความกว้างพัลส์มากขึ้น แต่ถ้าสัญญาณพัลส์กว้างน้อยกว่า 1.5 มิลลิวินาที รถจะเคลื่อนที่ถอยหลัง และความเร็วจะเพิ่มขึ้น ถ้าความกว้างของพัลส์ลดลง

หลักการการทำงานของ Speed Control

รูปที่ 3.4 แสดงวงจรที่สมบูรณ์ของ Speed Control ในวงจรนี้ Reverse จะมีความเร็วเป็นครึ่งหนึ่งของ Forward เพื่อประหยัด MOSFET 2 ตัว แต่ซึ่งจริง ๆ แล้วส่วนนี้มีประโยชน์มากทีเดียวเพราะว่ารถวิ่งด้วยความเร็ว Forward ได้เต็มที่และ Reverse ได้เต็มที่ จะทำให้เป็นอันตรายเป็นต่อเกียร์ส่งเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.4 ควบคุมความเร็ว Speed Control



SCHMATIC FOR SPEED CONTROL

สัญญาณพัลส์อินพุตถูกป้อนให้กับ IC1(LM339) IC1A จะหน่วงสัญญาณพัลส์ไป ซึ่งระยะเวลาสามารถหาค่าได้จาก R1, R2, R3 และ C1 เพื่อการ Discharge C2 หลังจากนั้น C2 จะ Charge ผ่าน VR1 (เป็นตัวปรับให้ได้ค่ากลาง ณ ตำแหน่งที่ลือไม่หมุนเมื่อจ่ายสัญญาณพัลส์ 1.5 มิลลิวินาที) สัญญาณพัลส์จะถูก IC1B หน่วงไป โดยหาช่วงเวลาหน่วงเวลาได้จาก VR1, C2 และ โวลเตจที่ขา อินเวอร์สอินพุต Q1 จะอินเวอร์สเอาท์พุท IC1C

IC2A(LM538) จะทำการเปรียบเทียบสัญญาณพัลส์ที่มาจาก Q1 ซึ่งเข้าไปยังขาอินเวอร์สอินพุต กับระดับโวลต์เตจ +Vb และทำการอินทิเกรท ค่าผิดพลาด R7-R10 จะช่วยลดริปเปิลที่เอาท์พุทของ IC2A

IC2A จะอยู่ใน Negative Feedback Loop ซึ่งป้อนกลับให้กับ IC1B ดังนั้นเอาท์พุทโวลเตจจะเปลี่ยนแปลงจนกระทั่งความกว้างของสัญญาณพัลส์ ซึ่งถูกสร้างโดย IC1B ตรงกับสัญญาณอินพุตพัลส์ โวลเตจที่เอาท์พุทของ IC2A เป็นสัดส่วนกับความกว้างของสัญญาณพัลส์ และป้อนให้กับตัวเปรียบเทียบ Forward และ Reverse IC3A และ IC3B ตามลำดับ R11 และ R12 จะทำการไบแอสตัวเปรียบเทียบ เพื่อป้องกันไม่ให้ทั้งสองตัว แอคทีฟ พร้อมกัน

อินพุตอีกส่วนหนึ่งจะป้อนให้กับตัวเปรียบเทียบ IC3A และ IC3B เป็นคลื่นสามเหลี่ยม ถูกสร้างโดย IC2B เอาท์พุทของ IC2A จะเปลี่ยนแปลงขึ้นและลง (ขึ้นอยู่กับความกว้างของพัลส์ที่ได้รับ) ตัวเปรียบเทียบจะทำการเปรียบเทียบแล้วให้สัญญาณพัลส์ที่มี Duty Cycle เปลี่ยนแปลงไปขึ้นอยู่กับสัญญาณอินพุต ซึ่งโวลเตจจะมีค่า +5V ของแหล่งจ่ายถึง +V. ซึ่งเป็นตัวจ่ายให้กับ NOR Gate หน้าที่ของ NOR Gate คือ

1. มีความสามารถในการ Drive สูง เพื่อให้ได้สัญญาณที่เรียบป้อนให้กับ MOSFET

2. ป้องกันไม่ให้ Forward และ Reverse MOSFETS ทำงานพร้อมกันในเวลาเดียวกัน

3. ทำให้ความเร็ว Reverse เป็นครึ่งหนึ่งของ Full Scale

โวลเตจทริปเปิลเลอร์ (Voltage Tripler) ประกอบด้วย C5, C6, C7, D1, D2 และ D3 เพื่อสร้าง +V จาก คลื่นสี่เหลี่ยมของ IC3C และ IC3D ซึ่งได้รับ Input จาก IC2B D4 จะจำกัด +V ถึง 15V ซึ่งเป็นค่าสูงสุดที่ CD4001 สามารถทนได้

MOSFET 6 ตัว Q2 ถึง Q7 4 ตัวจะสำหรับ Forward และ 2 ตัวสำหรับ Reverse R22, R23 และ R24 เป็น RF Chokes

เอาท์พุทคลื่นสี่เหลี่ยมจาก IC2B จะถูกอินเวอร์สโดย Q8 และป้อนให้กับนอร์เกต และอินพุตอีกส่วนหนึ่งของนอร์เกตจะมาจากสัญญาณพัลส์ของตัวเปรียบเทียบ Reverse IC3B เพื่อให้ Half Scale Reverse

IC1D และส่วนประกอบของมันจะช่วยเพิ่มความปลอดภัยขึ้น ขณะที่ยังมีอินพุทพัลส์อยู่ C8 จะทำการดีสชาร์จผ่าน D5 แต่เมื่อไม่มีสัญญาณพัลส์ป้อนให้ R28 จะชาร์จผ่าน C8 และ IC1D จะ Pull Low เพื่อกำจัด +V Supply

3.3 การบังคับเลี้ยว

การควบคุมการบังคับเลี้ยว นั้น มีลักษณะเช่นเดียวกับการควบคุมความเร็วคือ ใช้ความกว้างของสัญญาณพัลส์ควบคุมวงเลี้ยวและทิศทางของล้อ ต่างกันที่สัญญาณพัลส์จะถูกป้อนให้กับ Servo Motor ซึ่งจะไปควบคุมรัศมีการเลี้ยวของล้อ ซึ่งสัญญาณพัลส์กว้าง 1.35 มิลลิวินาที และมีคาบเวลา 19 มิลลิวินาที ให้รถวิ่งตรงไป แต่ถ้าสัญญาณพัลส์กว้างกว่านี้ ล้อจะเลี้ยวไปทางขวา และถ้าน้อยกว่านี้ล้อจะเลี้ยวไปทางซ้าย โดยมีคาบเวลาคงที่เท่ากับ 19 มิลลิวินาที เช่นเดียวกับการควบคุมความเร็ว

3.4 ไมโครคอนโทรลเลอร์

การเลือกไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ MCS-51 เป็นไมโครคอมพิวเตอร์แบบชิพเดี่ยว (ไม่ต้องต่อกับอุปกรณ์ภายนอกก็สามารถทำงานได้) มีความสะดวกในการใช้งานและเขียนโปรแกรมควบคุมด้วยภาษาเบสิกได้ โดยไม่ต้องศึกษาการทำงานของวงจรเหมือนกับภาษาแอสเซมบลี แต่ภาษาแอสเซมบลีก็ยังสามารถใช้ในการเขียนโปรแกรมได้เช่นกัน

MCS-51 นี้เป็นอุปกรณ์ที่ออกแบบเพื่อตอบสนองความต้องการของผู้ใช้คือ มีสายอินพุทและเอาต์พุทภายในตัวเอง พอร์ตของอินพุทและเอาต์พุทบัฟเฟอร์ อินเตอร์เฟส และสายควบคุมอื่น ๆ ที่ใช้สำหรับแยกข้อมูลกับแอดเดรสออกจากกันและยังมีชุดคำสั่งเพิ่มขึ้นเป็นพิเศษ เพื่อจัดการข้อมูลพร้อมด้วยวงจรตั้งเวลาและวงจรรีบด้วย (ปกติวงจรรีบจะสามารถทำงานเป็นวงจรตั้งเวลาได้ด้วย จึงเรียกควบคู่กันไป คือ วงจรตั้งเวลา/วงจรรีบ)

สำหรับในโครงการนี้ได้เลือกใช้บอร์ด ANT-32 ซึ่งมีคุณสมบัติดังนี้

1. สร้างโดย CHMOS เทคโนโลยี และทำงานด้วยแหล่งจ่ายไฟขนาด 5 V. เพียงแหล่งเดียว
2. CPU มีขนาด 8 บิต
3. เป็นบอร์ดคอนโทรลใช้กับไมโครคอนโทรลเลอร์ในตระกูล MCS-51 ใช้ CPU เบอร์ 8032 ทำงานที่ความถี่สัญญาณนาฬิกา 11.0592 MHz
4. ใช้งานหน่วยความจำบนบอร์ดได้ 3 ตำแหน่งด้วยกัน คือ
 - หน่วยความจำโปรแกรม (Program Memory) ใช้กับ EPROM ขนาด 8-32 KByte เบอร์ 2764, 27128 หรือ 27256
 - หน่วยความจำข้อมูล (Data Memory) ใช้กับ RAM ขนาด 8 KByte

เบอร์ 6264 หรือ 32 KBYTE เบอร์ 62256 สามารถแบคอัพข้อมูลได้โดยใช้แบตเตอรี่
ลิเทียม

-หน่วยความจำโปรแกรมและข้อมูล (Program and Data Memory) ใช้
กับ EPROM, RAM หรือ EEPROM ขนาด 8-32 KByte โดยใช้ EPROM เบอร์ 2764,
27256 ใช้ RAM เบอร์ 6264, 62256 หรือ EEPROM เบอร์ 2864(A), 28256(A)

5. ชุดแบงค์ (Bank) รีจิสเตอร์มี 4 ชุด แต่ละชุดมีรีจิสเตอร์ 8 ตัว
 6. มีตัวจับเวลา/ตัวนับ ขนาด 16 บิต 3 ชุด
 7. มีวงจร Serial Interface Driver RS232 ด้วยชิพเบอร์ MAX232
สำหรับการต่อเข้ากับเครื่องไมโครคอมพิวเตอร์
 8. มีวงจร Watchdog Timer, Power Up/Down Reset ด้วยชิพเบอร์
MAX691
 9. มีพอร์ต I/O 8255 จำนวน 2 ตัว (48 บิต) สำหรับต่อใช้งานภายนอก
 10. มีวงจร RTC (Real Time Clock) ใช้ชิพเบอร์ DS1202
 11. มีคอนเนกเตอร์สำหรับ PORT 1 ของไมโครคอนโทรลเลอร์โดยเฉพาะ
 12. มีคอนเนกเตอร์สำหรับ System Bus ทำให้ขยายระบบได้ง่าย และ
สามารถใช้กับบอร์ดขยายต่าง ๆ ที่มีขึ้นในอนาคต
 13. สามารถเลือกพอร์ตและชนิดหน่วยความจำ หรือกำหนดคุณสมบัติต่าง ๆ
ของบอร์ดได้ด้วยจิมพ์เปอร์
 14. สามารถพัฒนาโปรแกรมได้ทั้งภาษาเบสิกและแอสเซมบลี โดยใช้ซอฟต์แวร์
พร้อม BASIC32 และ REM31
 15. มีซอฟต์แวร์แฟลช สำหรับผู้ใช้ที่จะกำหนดได้ถึง 128 ตำแหน่งบิต
 16. โครงสร้างอินเทอร์รัพท์ทำได้ 6 แหล่ง พร้อมด้วยการจัดไพริอริตี้ (prior
ity) ได้ 2 ระดับ
 17. ตัวโปรแกรมเมอร์สามารถใช้งานแบบบูลีน (Boolean) ได้ จึงเหมาะสม
สำหรับการใช้งานควบคุม
 18. มีคำสั่งคูณ และหารทางฮาร์ดแวร์ ทำได้ภายใน 4 ไมโครวินาที
 19. ตัวเลขทางคณิตศาสตร์ ใช้ได้ทั้งแบบไบนารี และแบบเดซิมีล
 20. การใช้พื้นที่ สแต็ก สำหรับโปรแกรมย่อยต่าง ๆ ทำได้ง่ายและกว้าง
- จากคุณสมบัติที่กล่าวมาแล้วทั้งหมดนี้ จะเห็นได้ว่า บอร์ด ANT-32 เหมาะที่
จะนำมาใช้ในงานควบคุมการเคลื่อนที่ของตัวรถในโครงการนี้

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง การศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด TIMER/COUNTER ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใน 8051 มี Timer/Counter Register ขนาด 16 บิต จำนวน 2 ตัว คือ Timer0 และ Timer1 แต่ถ้าเป็น 8052 จะมี Timer/Counter เพิ่มมาให้อีก 1 ตัว คือ Timer2 โดย Timer/Counter ทั้งสามตัวเป็นรีจิสเตอร์ที่อยู่ในหน่วยความจำ บริเวณ SFR ซึ่งผู้ใช้สามารถกำหนดการทำงานให้เป็น Timer หรือ Counter ได้อย่างใดอย่างหนึ่ง โดยมีรายละเอียดดังนี้

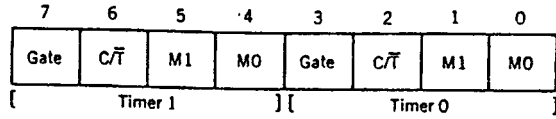
- Timer Mode ค่าของรีจิสเตอร์จะถูกเพิ่มค่าทุก ๆ Machine Cycle ดังนั้นจึงสามารถคิดว่ามีเป็นการนับ Machine Cycle ได้ และเนื่องจากใน 1 Machine Cycle ประกอบไปด้วย 12 Oscillator Period ดังนั้นอัตราเร็วในการนับ (Counter Rate) จึงมีค่าเป็น $1/12$ ของความถี่ออสซิลเลเตอร์ที่ใช้ในระบบ

- Counter Mode ค่าในรีจิสเตอร์ จะถูกเพิ่มค่าทีละหนึ่ง ตามการเปลี่ยนสถานะที่ของ T0, T1 หรือ T2 (ใน 8052) โดยตรวจสอบการเปลี่ยนค่าจาก 1 เป็น 0 (1 to 0 Transition) ในการตรวจสอบการเปลี่ยนค่าจาก 1 เป็น 0 ที่กล่าวมานี้จะถูกตรวจสอบ (Sampled) ระหว่าง S5P2 ของแต่ละ Machine Cycle โดยมีรายละเอียดในการตรวจสอบดังนี้

เมื่อมีสัญญาณที่ขา T0, T1 หรือ T2 มีค่าเป็น 1 (High) ใน S5P2 ของ Cycle ใด และใน Cycle ถัดไปที่ S5P2 สัญญาณที่ขา T0, T1 หรือ T2 มีสถานะเป็น 0 (Low) จะทำให้ค่าที่อยู่ในรีจิสเตอร์ (Timer/Counter Register) ถูกเพิ่มอีก 1 ในช่วง S3P1 ของ Cycle ที่ถัดจาก Cycle ซึ่งตรวจพบการเปลี่ยนค่า จะเห็นว่า ต้องใช้ Machine Cycle จำนวน 2 Cycle (24 Oscillator Period) เพื่อตรวจสอบการเปลี่ยนสถานะจาก 1 เป็น 0 ที่ขา T0, T1 หรือ T2 ดังนั้นอัตราการนับสูงสุดจึงมีค่าเท่ากับ $1/24$ ของ Oscillator Frequency โดยไม่มีข้อจำกัดในเรื่อง Duty Cycle ของสัญญาณ แต่เพื่อให้มั่นใจว่าระดับที่อ่านถูกตรวจสอบ (Sampled) เข้ามาอย่างน้อย 1 ครั้ง ก่อนที่มันจะเปลี่ยนระดับ จึงควรให้สัญญาณคงค่า 0 หรือ 1 ไว้อย่างน้อย 1 Machine Cycle เต็ม นอกจากนี้ผู้ใช้จะสามารถเลือกการทำงานของรีจิสเตอร์ให้เป็น Timer หรือ Counter ได้แล้ว ในโหมด Timer หรือ Counter ยังมีโหมดการทำงานพิเศษแยกย่อยลงไปอีกถึง 4 โหมด ตามความเหมาะสมของการทำงาน (T0, T1 มีให้เลือก 4 โหมด แต่ T2 มีให้เลือก 3 โหมด) ดังจะได้กล่าวต่อไป

Timer0 และ Timer1

Timer/Counter 2 ตัวนี้มีอยู่ทั้งใน 8051 และ 8052 โดยผู้ใช้สามารถจะเลือกการทำงานให้เป็น Timer หรือ Counter ด้วยการเปลี่ยนค่าบิตใน Control Bit C/T* ซึ่งอยู่ใน SFR TMOD (รูปที่ 3.5) โดยถ้าบิตนี้เป็น 0 หมายถึงเลือกให้เป็น Timer (นับจำนวน Machine Cycle) ถ้าบิตนี้เป็น 1 หมายถึงเลือกให้เป็น Counter (นับจำนวนการเปลี่ยนสถานะจาก 1 เป็น 0 ที่ขา T0, T1) อย่างไรก็ตามถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



THE TIMER MODE CONTROL (TMOD) SPECIAL FUNCTION REGISTER

Bit	Symbol	Function
7/3	Gate	OR gate enable bit which controls RUN/STOP of timer 1/0. Set to 1 by program to enable timer to run if bit TR1/0 in TCON is set and signal on external interrupt INT1/0 pin is high. Cleared to 0 by program to enable timer to run if bit TR1/0 in TCON is set.
6/2	C/T	Set to 1 by program to make timer 1/0 act as a counter by counting pulses from external input pins 3.5 (T1) or 3.4 (T0). Cleared to 0 by program to make timer act as a timer by counting internal frequency.
5/1	M1	Timer/counter operating mode select bit 1. Set/cleared by program to select mode.
4/0	M0	Timer/counter operating mode select bit 0. Set/cleared by program to select mode.

M1	M0	Mode
0	0	0
0	1	1
1	0	2
1	1	3

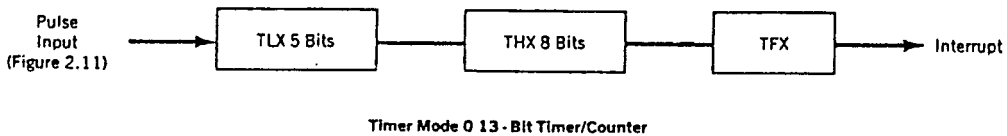
TMOD is not bit addressable

รูปที่ 3.5 TMOD : Timer/Counter Mode Control Register

Timer/Counter ทั้ง 2 ตัวนี้มีโหมดการทำงานอยู่ 4 โหมด โดยถูกเลือกตามค่าของบิต M0, M1 ใน TMOD เช่นเดียวกับบิต C/T* การทำงานในโหมด 0, 1, 2 จะคล้าย ๆ กันสำหรับ Timer/Counter ทั้ง 2 แต่ในโหมด 3 จะมีการทำงานที่ต่างออกไปจากทั้ง 3 โหมด ดังนี้

Mode 0

แต่ละ Timer ในโหมดนี้จะถูกทำงานเป็น Counter ขนาด 8 บิต ซึ่งจะถูกเพิ่มค่าต่อเมื่อนับจำนวน Machine Cycle ได้ 32 Cycle แล้ว นั่นคือทำงานเป็น Counter ขนาด 13 บิตนั่นเอง โดยการนับ Machine Cycle จำนวน 32 Cycle จะใช้รีจิสเตอร์ TLx จำนวน 5 บิต ($2^5=32$) และนำมาประกอบกับรีจิสเตอร์ THx จำนวน 8 บิต เพื่อให้เป็น Counter ขนาด 13 บิต รูปที่ 3.6 แสดงการทำงานในโหมด 0 ของ Timer 1



รูปที่ 3.6 Timer/Counter1 Mode 0 : 13 Bits Counter

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้กับภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย จากรูปจะเห็นว่า รีจิสเตอร์ TL1 และ TH1 จะมาประกอบกันเข้าเป็น Counter ขนาด 13 บิต เมื่อค่าในรีจิสเตอร์ทั้งสองเปลี่ยนจาก 01 ทั้งหมดไปเป็น 01 ทั้ง

หมด จะมีผลไปเช็ท Timer Interrupt Flag: TF1 อินพุทที่ใช้ นับถูก Enable ให้กับ Timer เมื่อ TR1 และ Gate = 0 หรือสัญญาณที่ขา INT1* เป็น 1 อย่างใดอย่างหนึ่ง (การเช็ทให้ Gate เป็น 1 จะยอมให้ Timer ถูกควบคุมโดยสัญญาณที่ขา INT1* ทำให้การวัดความกว้างของพัลส์ทำได้ง่าย) TR1 เป็นบิตในรีจิสเตอร์ TCON ดังรูปที่ 3.7 ส่วน Gate อยู่ใน TMOD ดังได้กล่าวมาแล้ว

รีจิสเตอร์ขนาด 13 บิต ประกอบด้วย 8 บิตของ TH1 และ 5 บิตของ TL1 ส่วน 3 บิตบนของ TL1 ไม่ถูกกำหนดและไม่ถูกใช้ การเช็ทค่าของ Run Flag (TR1) ไม่ได้เคลียร์ค่าในรีจิสเตอร์ทั้งสอง

7	6	5	4	3	2	1	0
TF1	TR1	TFO	TRO	IE1	IT1	IE0	IT0

THE TIMER CONTROL (TCON) SPECIAL FUNCTION REGISTER

Bit	Symbol	Function
7	TF1	Timer 1 Overflow flag. Set when timer rolls from all ones to zero. Cleared when processor vectors to execute interrupt service routine located at program address 0018h.
6	TR1	Timer 1 run control bit. Set to 1 by program to enable timer to count; cleared to 0 by program to halt timer. Does not reset timer.
5	TFO	Timer 0 Overflow flag. Set when timer rolls from all ones to zero. Cleared when processor vectors to execute interrupt service routine located at program address 0008h.
4	TRO	Timer 0 run control bit. Set to 1 by program to enable timer to count; cleared to 0 by program to halt timer. Does not reset timer.
3	IE1	External interrupt 1 edge flag. Set to 1 when a high to low edge signal is received on port 3 pin 3.3 (INT1). Cleared when processor vectors to interrupt service routine located at program address 0013h. Not related to timer operations.
2	IT1	External interrupt 1 signal type control bit. Set to 1 by program to enable external interrupt 1 to be triggered by a falling edge signal. Set to 0 by program to enable a low level signal on external interrupt 1 to generate an interrupt.
1	IE0	External interrupt 0 edge flag. Set to 1 when a high to low edge signal is received on port 3 pin 3.2 (INT0). Cleared when processor vectors to interrupt service routine located at program address 0003h. Not related to timer operations.
0	IT0	External interrupt 0 signal type control bit. Set to 1 by program to enable external interrupt 0 to be triggered by a falling edge signal. Set to 0 by program to enable a low level signal on external interrupt 0 to generate an interrupt.

Bit addressable as TCON.0 to TCON.7

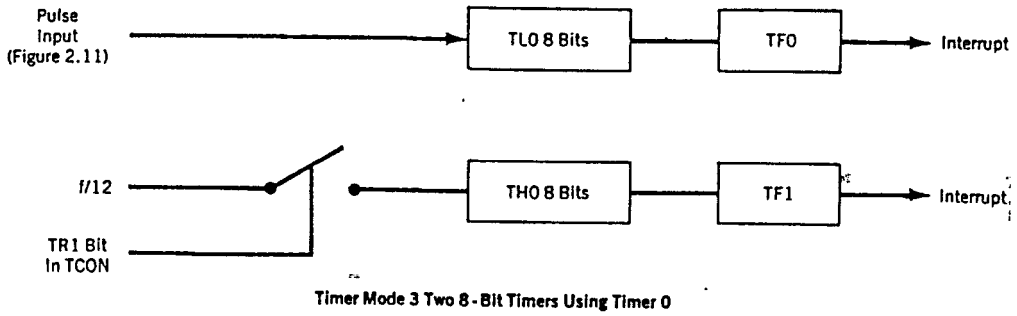
รูปที่ 3.7 TCON : Timer/Counter Control Register

Mode 1

ในโหมด 1 นี้มีการทำงานเหมือนกับโหมด 0 ทุกประการ เว้นแต่ค่าใน Timer Register จะถูก Run ทั้ง 16 บิตเลย นั่นคือในโหมดนี้จะเป็น Counter หรือ Timer ขนาด 16 บิตแทน

Mode 2

ในโหมด 2 จะกำหนด Timer Register เป็น 8-Bits Counter (TLx) ซึ่งมีการโหลดค่าเองโดยอัตโนมัติด้วยค่าในรีจิสเตอร์ THx เมื่อเกิด Overflow ซึ่งค่าใน THx นี้สามารถกำหนดได้ล่วงหน้าโดยซอฟต์แวร์ และจะไม่เปลี่ยนแปลงเมื่อถูกโหลดไปไว้ใน TL1



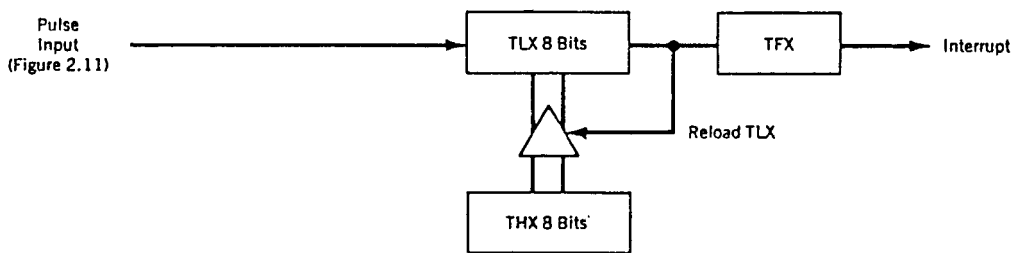
รูปที่ 3.8 Timer/Counter 1 Mode 2: 8 Bits Auto-Reload

Mode 3

ในการทำงานของ Timer/Counter โหมด 3 นี้ Timer1 จะไม่มีการนับ ซึ่งมีผลเหมือนกับให้ค่า $TR1 = 0$

Timer0 ในโหมด 3 จะทำให้ TLO และ THO เหมือนเป็น Counter ขนาด 8 บิต แยกต่างหากกัน 2 ตัว Logic สำหรับการทำงานในโหมด 3 ของ Timer 0 แสดงในรูปที่ 3.9 ซึ่งจะเห็นว่า TLO ใช้ Timer0 Control Bits C/T*, Gate, TR0, INTO* และ TFO ส่วน THO จะถูกบังคับให้เป็น Timer (นับจำนวน Machine Cycle) และรับการให้ TR1 และ TF1 จาก Timer1 ดังนั้นขณะนี้ THO จะควบคุม Timer1 Interrupt

Mode 3 มีเพื่อการใช้งานที่ต้องการ Timer หรือ Counter 8 บิตเพิ่มขึ้น เมื่อใช้ Timer0 ในโหมด 3 8051 สามารถมองเหมือนว่ามี Timer/Counter 3 ตัว และ 8052 มี Timer/Counter 4 ตัว เมื่อ Timer0 อยู่ในโหมด 3 Timer1 สามารถถูกควบคุมการนับให้เริ่มนับ หรือ หยุดนับได้โดยการสวิตช์ออกจากโหมด 3 ไปสู่โหมดอื่น ดังนั้น จึงมี Timer/Counter ใช้มากขึ้นอีก (เพิ่มจาก TLO และ THO) ทำให้สามารถใช้กับ Serial Port โดยเป็น Baud Rate Generator แต่จริง ๆ แล้ว Timer1 สามารถถูกใช้ในงานใด ๆ ก็ได้ ที่ไม่ต้องการการอินเทอร์รัพท์ ทั้งนี้เพราะการอินเทอร์รัพท์ของ Timer1 ถูกใช้โดย Timer0 ไปแล้วนั่นเอง



Timer Mode 2 Auto-Reload of TL from TH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 รูปที่ 3.9 Timer/Counter 0 Mode 3: Two 8-Bits Counters

Timer 2

Timer2 เป็น Timer/Counter ขนาด 16 บิต ซึ่งมีเฉพาะใน 8052 โดยมันสามารถทำงานเป็น Timer หรือ Counter อย่างใดอย่างหนึ่งเหมือน Timer0 และ Timer1 โดยสามารถเลือกได้ด้วยบิต C/T*2 ใน SFR T2CON ใน Timer2 นี้มีการทำงานอยู่ 3 โหมดด้วยกัน คือ Capture, Auto Reload และ Baud Rate Generator ซึ่งสามารถเลือกได้โดยการกำหนดค่าบิตในรีจิสเตอร์ T2CON ดังแสดงในตารางที่ 3.1

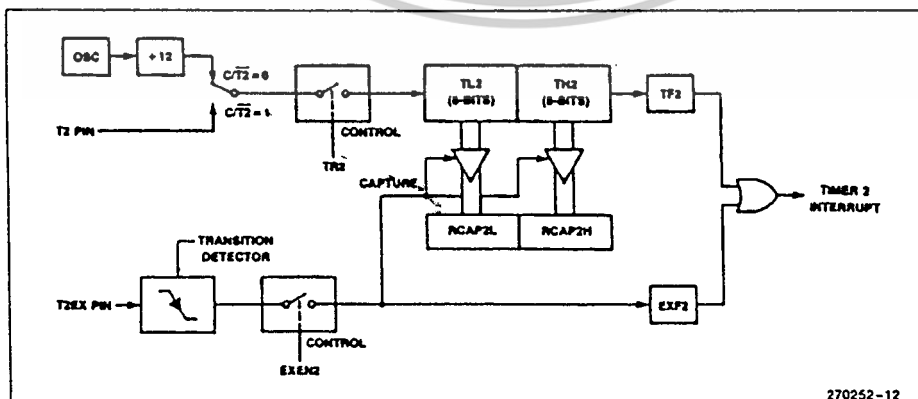
RCLK + TCLK	CP/RL2	TR2	Mode
0	0	1	16-bit Auto-Reload
0	1	1	16-bit Capture
1	X	1	Baud Rate Generator
X	X	0	(off)

ตารางที่ 3.1 แสดงการกำหนดค่า BIT ในรีจิสเตอร์ T2CON

การทำงานแต่ละโหมดของ Timer2 มีรายละเอียดดังนี้

Capture Mode

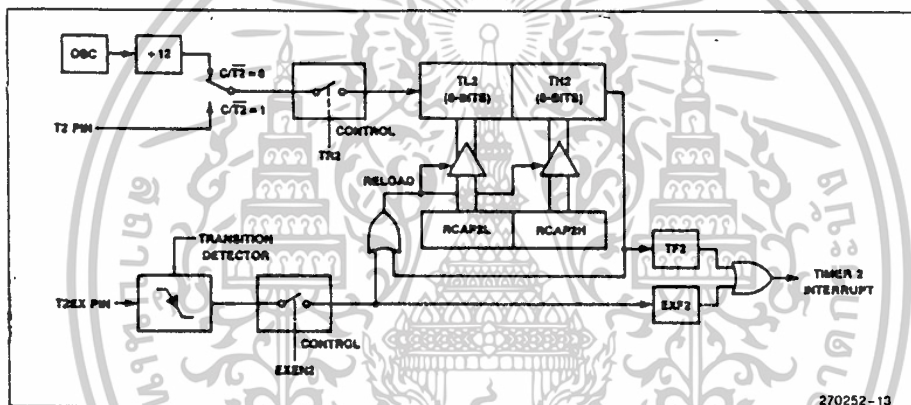
มี 2 Options ให้เลือก สามารถถูกเลือกได้โดยบิต EXEN2 ใน T2CON ซึ่งถ้า EXEN2 = 0 แล้ว Timer2 จะเป็น Timer หรือ Counter ขนาด 16 บิต ซึ่ง Overflow จะไปเซ็ทบิต TF2 (Timer2 Overflow Flag) ส่งผลให้เกิดอินเทอร์รัพต์ได้ แต่ถ้า EXEN2 = 1 Timer2 จะยังคงทำงานเหมือนที่กล่าวมาแล้ว แต่มีคุณสมบัติพิเศษที่เพิ่มเข้ามาคือ สัญญาณที่มีการเปลี่ยนจาก 1 เป็น 0 ที่ขา T2EX จะทำให้ค่าปัจจุบัน (Current Value) ใน Timer2 Register (TL2, TH2) ถูกไหลด (Captured) ลงไปในรีจิสเตอร์ RCAP2L และ RCAP2H ตามลำดับ (RCAP2L และ RCAP2H เป็น SFR ที่มีเพิ่มขึ้นมาใน 8052) นอกจากนี้ การเปลี่ยนระดับ (Transition) ที่ T2EX จะทำให้บิต EXF2 ใน ถูกเซ็ท T2CON และบิต EXF2 สามารถทำให้เกิดอินเทอร์รัพต์ได้เช่นเดียวกับ TF2 (Capture Mode สามารถอธิบายการทำงานได้ดังรูปที่ 3.10)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกที่รูปที่ 3.10 Timer2 Capture Mode เอกสารทุกครั้งที่มีการนำไปใช้

Auto-Reload Mode

มี 2 Options ให้เลือกเช่นเดียวกันซึ่งถูกเลือกโดยบิต EXEN2 ใน T2CON โดยถ้า EXEN2 = 0 แล้ว เมื่อค่าใน Timer2 เปลี่ยนจากเป็น 1 ทั้งหมดไปเป็น 0 ทั้งหมด (นับครบ 16 บิต) ไม่เพียงแต่จะทำให้ TE2 ถูกเซ็ท แต่จะทำให้ค่าในรีจิสเตอร์ของ Timer2 ถูกโหลดอีกครั้งด้วยค่า 16 บิตในรีจิสเตอร์ RCAP2L และ RCAP2H ซึ่งสามารถตั้งไว้ล่วงหน้าได้ด้วยซอฟต์แวร์ ถ้า EXEN2 = 1 Timer2 จะยังคงทำงานเหมือนดังที่กล่าวมา แต่มีคุณสมบัติพิเศษเพิ่มขึ้นคือ การเปลี่ยนสถานะจาก 1 เป็น 0 ที่ขา T2EX จะไป Trig ให้มีการโหลดค่าจาก RCAP2L และ RCAP2H ไปยังรีจิสเตอร์ของ Timer2 ทันที และเซ็ทบิต TF2 เช่นกัน กล่าวคือ การโหลดค่าจาก RCAP2L และ RCAP2H ไปยังรีจิสเตอร์ของ Timer2 จะเกิดขึ้นได้โดยไม่ต้องรอให้ Timer2 เกิด Overflow นั้นเอง แสดงดังรูปที่ 3.11



รูปที่ 3.11 Timer2 in Auto-Reload Mode

T2CON: TIMER/COUNTER 2 CONTROL REGISTER. BIT ADDRESSABLE

8052 Only

TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2
-----	------	------	------	-------	-----	------	--------

TF2	T2CON. 7	Timer 2 overflow flag set by hardware and cleared by software. TF2 cannot be set when either RCLK = 1 or CLK = 1					
EXF2	T2CON. 6	Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX, and EXEN2 = 1. When Timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software.					
RCLK	T2CON. 5	Receive clock flag. When set, causes the Serial Port to use Timer 2 overflow pulses for its receive clock in modes 1 & 3. RCLK = 0 causes Timer 1 overflow to be used for the receive clock.					
TCLK	T2CON. 4	Transmit clock flag. When set, causes the Serial Port to use Timer 2 overflow pulses for its transmit clock in modes 1 & 3. TCLK = 0 causes Timer 1 overflows to be used for the transmit clock.					
EXEN2	T2CON. 3	Timer 2 external enable flag. When set, allows a capture or reload to occur as a result of negative transition on T2EX if Timer 2 is not being used to clock the Serial Port. EXEN2 = 0 causes Timer 2 to ignore events at T2EX.					
TR2	T2CON. 2	Software START/STOP control for Timer 2. A logic 1 starts the Timer.					
C/T2	T2CON. 1	Timer or Counter select. 0 = Internal Timer. 1 = External Event Counter (falling edge triggered).					
CP/RL2	T2CON. 0	Capture/Reload flag. When set, captures will occur on negative transitions at T2EX if EXEN2 = 1. When cleared, Auto-Reloads will occur either with Timer 2 overflows or negative transitions at T2EX when EXEN2 = 1. When either RCLK = 1 or TCLK = 1, this bit is ignored and the Timer is forced to Auto-Reload on Timer 2 overflow.					

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

รูปที่ 3.12 T2CON: Timer/Counter 2 Control Registerที่มีการนำไปใช้

Baud Rate Generator Mode

ถูกเลือกโดยการให้พิก $RCLK = 1$ และ/หรือ $TCLK = 1$ ซึ่งจะอธิบายร่วมกับเรื่อง Serial Port ต่อไป

SERIAL INTERFACE

Microcontroller MCS-51 มี Serial Port ซึ่งสามารถรับและส่งข้อมูลแบบอนุกรมได้ โดยผู้ใช้ไม่ต้องต่อไอซีเพิ่มเติมเข้าไป ทำให้มีความสะดวกในการนำไปประยุกต์ใช้งานที่ต้องการติดต่อข้อมูลแบบอนุกรม Serial Port ที่มีใน MCS-51 สามารถทำงานได้ในแบบ Full Duplex หมายความว่า มันสามารถรับและส่งข้อมูลได้พร้อม ๆ กัน โดยมีการ Buffer ในการรับส่งข้อมูลให้ด้วย กล่าวคือ มันสามารถกำหนดการรับของไบต์ที่สองที่ถูกรับเข้ามา ก่อนที่ไบต์แรกซึ่งได้รับเข้ามาก่อนจะถูกอ่านจาก Receive Register (แต่ถ้าไบต์แรกยังไม่ถูกอ่านเมื่อเวลาที่การรับของไบต์ที่สองสิ้นสุดลง หนึ่งไบต์ในสองไบต์จะหายไป) Serial Port จริง ๆ แล้วประกอบไปด้วยรีจิสเตอร์ขนาด 8 บิต จำนวนสองตัวซึ่งมีชื่อเรียกดังนี้ คือ Receive Register และ Transmit Register ซึ่งรีจิสเตอร์ทั้งสองมีตำแหน่งเดียวกันใน SFR คือรีจิสเตอร์ SBUF โดยการเข้าถึงรีจิสเตอร์แต่ละตัว ผู้ใช้จะรู้เองว่าผู้ใช้ต้องการติดต่อกับรีจิสเตอร์ตัวใด เพราะในการเขียนข้อมูลไปที่รีจิสเตอร์ SBUF จะหมายถึงโหลดค่าไปยัง Transmit Register ส่วนการอ่านข้อมูลในรีจิสเตอร์ SBUF หมายถึงการรับข้อมูลจาก Receive Register

Serial Port ใน MCS-51 สามารถทำงานในโหมดต่าง ๆ กันได้ถึง 4 แบบด้วยกัน เพื่อความเหมาะสมกับงานแต่ละงาน ดังนี้

Mode 0

ข้อมูลแบบอนุกรมถูกรับเข้าและส่งออกผ่านทางขา RXD ส่วนขา TXD จะให้ Shift Clock ในโหมดนี้จะทำการรับส่งข้อมูลแบบ 8 บิต:8 Data Bits โดยรับและส่งบิตต่ำก่อน (LSB First) ส่วนอัตราการส่งข้อมูล (BAUD Rate) ถูกกำหนดไว้แน่นอนที่ $1/12$ ของความถี่ออสซิลเลเตอร์ที่ใช้ในระบบ

Mode 1

ข้อมูลจำนวน 10 บิตถูกส่ง (ผ่านทางขา TXD) หรือถูกรับ (ผ่านทางขา RXD) โดยมี 1 Start Bit (มีค่าเป็น 0) 8 Data Bits (รับและส่งบิตที่ต่ำก่อน) และ 1 Stop Bit (มีค่าเป็น 1) ขณะที่การรับ Stop Bit จะไปอยู่ในบิต RB8 ของ SFR SCON ส่วนค่า BAUD Rate สามารถเปลี่ยนแปลงได้ ดังจะได้กล่าวในรายละเอียดต่อไป

Mode 2

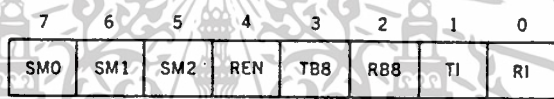
ข้อมูลจำนวน 11 บิตถูกส่ง (ผ่านทางขา TXD) หรือถูกรับ (ผ่านทางขา RXD) มี 1 Start Bit (เป็น 0) 8 Data Bit (รับหรือส่งบิตต่ำก่อน) บิตที่ 9 เป็นบิตที่สามารถโปรแกรมได้ (Programmable 9th Data Bit) และ 1 Stop Bit (เป็น 1) โดยใน

ขณะส่งข้อมูล บิตข้อมูลบิตที่ 9 (The 9th Data Bit) ซึ่งเป็นบิต TB8 ใน SFR SCON สามารถถูกกำหนดให้เป็น 0 หรือ 1 ตัวอย่างเช่น Parity Bit (Bit P ในรีจิสเตอร์ PSW) สามารถนำไปไว้ที่บิต TB8 ส่วนในขณะรับข้อมูล บิตข้อมูลบิตที่ 9 (The 9th Data Bit) จะไปอยู่ในบิต RB8 ของรีจิสเตอร์ SCON โดยไม่สนใจ Stop Bit ค่า Baud Rate สามารถตั้งให้เป็น 1/32 หรือ 1/64 ของความถี่ออสซิลเลเตอร์ที่ใช้

Mode 3

ข้อมูลจำนวน 11 บิตถูกส่ง (ผ่านขา TXD) หรือถูกรับ (ผ่านขา RXD) โดยมี 1 Start Bit (0) 8 Data Bits (LSB First) 1 Programmable 9th Bit และ 1 Stop Bit (1) ซึ่งจะเห็นได้ว่าในโหมด 3 จะเหมือนกับโหมด 2 ทุกอย่าง เว้นเสียแต่ในโหมด 3 นี้สามารถเปลี่ยนแปลงค่า Baud Rate ได้ ดังจะกล่าวถึงรายละเอียดต่อไป

การทำงานของ Serial Port ทั้ง 4 โหมดที่กล่าวมานี้ การส่งข้อมูลถูกทำให้เกิดขึ้นโดยคำสั่งใด ๆ ที่ใช้ SBUF เป็น Destination Register ส่วนการรับข้อมูลถูกทำให้เกิดขึ้น ในโหมด 0 โดยเงื่อนไข RI = 0 และ REN = 1 การรับข้อมูลถูกทำให้เกิดขึ้น ในโหมดอื่น ๆ โดยรับ Start Bit เข้ามา ถ้า REN = 1



THE SERIAL PORT CONTROL (SCON) SPECIAL FUNCTION REGISTER

Bit	Symbol	Function		
7	SM0	Serial port mode bit 0. Set/cleared by program to select mode.		
6	SM1	Serial port mode bit 1. Set/cleared by program to select mode.		
	SM0	SM1	Mode	Description
	0	0	0	Shift register; baud = f/12
	0	1	1	8-bit UART; baud = variable
	1	0	2	9-bit UART; baud = f/32 or f/64
	1	1	3	9-bit UART; baud = variable
5	SM2	Multiprocessor communications bit. Set/cleared by program to enable multiprocessor communications in modes 2 and 3. When set to 1 an interrupt is generated if bit 9 of the received data is a 1; no interrupt is generated if bit 9 is a 0. If set to 1 for mode 1, no interrupt will be generated unless a valid stop bit is received. Clear to 0 if mode 0 is in use.		
4	REN	Receive enable bit. Set to 1 to enable reception; cleared to 0 to disable reception.		
3	TB8	Transmitted bit 8. Set/cleared by program in modes 2 and 3.		
2	RB8	Received bit 8. Bit 8 of received data in modes 2 and 3; stop bit in mode 1. Not used in mode 0.		
1	TI	Transmit interrupt flag. Set to one at the end of bit 7 time in mode 0, and at the beginning of the stop bit for other modes. Must be cleared by the program.		
0	RI	Receive interrupt flag. Set to one at the end of bit 7 time in mode 0, and halfway through the stop bit for other modes. Must be cleared by the program.		

Bit addressable as SCON.0 to SCON.7

รูปที่ 3.13 SCON : Serial Port Control Register

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าการใด ๆ ทั้งสิ้น ยกเว้นให้มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี เป็น SFR ที่เรียกย่อ

ว่า SCON โดยแต่ละบิตในรีจิสเตอร์มีความหมายดังในรูปที่ 3.13 ซึ่งรีจิสเตอร์ตัวนี้ไม่เพียงแต่เก็บข้อมูลเกี่ยวกับโหมดการทำงานของ Serial Port แต่ยังใช้เป็นที่เก็บข้อมูลซึ่งเป็นบิตที่ 9 สำหรับการส่งและการรับ(บิต TB8 และ RB8) รวมทั้งยังมีบิตที่กำหนดอินเทอร์รัพท์ซีพียู(Serial Port Interrupt)คือ บิต TI และ RI อีกด้วย

Baud Rate

Baud Rate หมายถึง อัตราเร็วในการรับส่งข้อมูล ซึ่งจะมีค่าเท่าใดก็ขึ้นอยู่กับการทำงานในแต่ละโหมดของ Serial Port ดังนี้

- ในโหมด 0 Baud Rate = (ความถี่ออสซิลเลเตอร์ที่ใช้)/12
- ในโหมด 2 ค่า Baud Rate ขึ้นอยู่กับค่าของบิต SMOD ซึ่งอยู่ในรีจิสเตอร์ PCON โดยถ้าบิต SMOD = 0 (ซึ่งเป็นค่านี้เมื่อเกิดการรีเซ็ต) Baud Rate จะเป็น 1/64 ของความถี่ออสซิลเลเตอร์ที่ใช้ แต่ถ้าบิต SMOD = 1 Baud Rate จะเป็น 1/32 ของความถี่ออสซิลเลเตอร์ หรืออาจเขียนเป็นสูตรได้ว่า

$$\text{Baud Rate ในโหมด 2} = [2^{(\text{SMOD})} * (\text{Oscillator Frequency})] / 64$$

ส่วน Baud Rate ในโหมด 1 และ 3 จะถูกกำหนดโดยอัตราการเกิด Overflow ใน Timer1 (Timer1 Overflow Rate) แต่ถ้าเป็น 8052 ซึ่งมี Timer/Counter เพิ่มมาอีก 1 ตัว สามารถเอา Timer2 ที่มีเพิ่มมานี้เป็นตัวกำหนด Baud Rate ได้ ทำให้มี Timer จำนวน 2 ตัว (Timer1 และ Timer2) ที่สามารถนำมากำหนด Baud Rate โดยอาจใช้ตัวใดตัวหนึ่งในการกำหนด Baud Rate การรับ และอีกตัวสำหรับการส่งข้อมูล

การใช้ Timer1 เป็นตัวสร้าง Baud Rate

เมื่อ Timer1 ถูกใช้เป็นตัวกำเนิด Baud Rate (Baud Rate Generator) สำหรับการทำงานของ Serial Port ในโหมด 1 และ 3 โดย Baud Rate จะถูกกำหนดด้วยอัตราการเกิด Overflow ของ Timer1 และยังขึ้นอยู่กับบิต SMOD ในรีจิสเตอร์ PCON อีกด้วย ซึ่งเราอาจเขียนเป็นสมการที่ใช้คำนวณหา Baud Rate ได้ดังนี้

$$\text{Baud Rate โหมด 1, 3} = [2^{(\text{SMOD})} * (\text{Timer1 Overflow Rate})] / 32$$

เนื่องจากเมื่อเกิด Overflow ใน Timer ตัวใด จะทำให้ซีพียูถูกอินเทอร์รัพท์ ดังนั้นเมื่อเรานำ Timer1 มาเป็นตัวสร้าง Baud Rate จึงควรที่ห้ามไม่ให้เกิดอินเทอร์รัพท์ขึ้นในระหว่างใช้งาน และตัว Timer เอง ยังสามารถถูกกำหนดให้ทำงานเป็น Timer หรือ Counter อย่างไม่อย่างหนึ่ง ซึ่งมีโหมดการทำงานย่อยลงไปอีก 3 โหมด

ในการใช้งานที่พบบ่อยที่สุดนั้น Timer1 ถูกกำหนดให้อยู่ในโหมด Auto-Reload (ค่า 4 บิตบนของ TMOD คือ 0010B) ในกรณีนี้ Baud Rate จะถูกกำหนดโดยสมการดังนี้

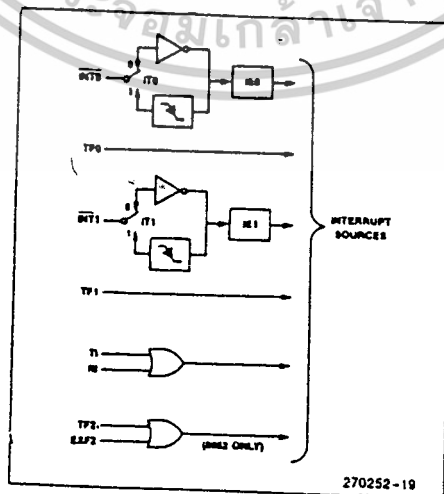
$$\text{Baud Rate โหมด 1, 3} = \frac{2^{SMOD} * (\text{Oscillator Frequency})}{32 * [256 - (TH1)]}$$

เราสามารถสร้าง Baud Rate ค่าต่างๆ ด้วย Timer1 โดยปล่อยให้ Timer 1 ทำการอินเทอร์รัพท์ที่พื้ชู้ได้ และกำหนดให้ Timer ทำงานในแบบ 16 บิต (โหมด 1 ค่า บิตบนของ TMOD คือ 0010B) และใช้ Timer1 อินเทอร์รัพท์ที่พื้ชู้เพื่อทำการโหลดค่าเอง ด้วยซอฟต์แวร์ขณะที่เกิด Overflow เนื่องจากในโหมด 16 บิต ไม่มีการทำงานแบบ Auto-Reload นั้นเอง

รูปที่ 3.14 เป็นค่า Baud Rate ค่าต่าง ๆ ที่ใช้กันมากและบอกว่าสามารถ ใช้ได้จาก Timer1 อย่างไร

Baud Rate	fosc	SMOD	Timer 1		
			C/T	Mode	Reload Value
Mode 0 Max: 1 MHz	12 MHz	X	X	X	X
Mode 2 Max: 375K	12 MHz	1	X	X	X
Modes 1, 3: 62.5K	12 MHz	1	0	2	FFH
19.2K	11.059 MHz	1	0	2	FDH
9.6K	11.059 MHz	0	0	2	FDH
4.8K	11.059 MHz	0	0	2	FAH
2.4K	11.059 MHz	0	0	2	F4H
1.2K	11.059 MHz	0	0	2	E8H
137.5	11.986 MHz	0	0	2	1DH
110	6 MHz	0	0	2	72H
110	12 MHz	0	0	1	FREEBH

รูปที่ 3.14 Timer1 Generated Commonly Used Baud Rate



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น รูปที่ 3.15 MCS-51 Interrupt Sources

INTERRUPTS

8051 มี Interrupt Source ให้ 5 Interrupt ด้วยกัน ส่วนใน 8052 มีให้ 6 Interrupt Source ดังแสดงในรูปที่ 3.15 โดย Interrupt แต่ละตัวมีรายละเอียดดังนี้

External Interrupts

INT0* และ INT1* เป็นอินเทอร์รัพท์ที่เกิดขึ้นภายนอก แต่ละตัวสามารถเลือกให้เป็น Level-Activated หรือ Transition-Activated อย่างใดอย่างหนึ่งขึ้นอยู่กับบิต ITO และ IT1 ในรีจิสเตอร์ TCON

Flag ที่ทำให้เกิดการอินเทอร์รัพท์ในซีพียูจริง ๆ แล้วคือค่าในบิต IE0 และ IE1 ของรีจิสเตอร์ TCON โดยเมื่อเกิดอินเทอร์รัพท์จากภายนอกขึ้น Flag ซึ่งทำให้เกิดอินเทอร์รัพท์จะถูกเคลียร์โดยฮาร์ดแวร์เมื่อซีพียูถูกย้ายไปทำงานที่ Interrupt Service Routine ก็ต่อเมื่ออินเทอร์รัพท์ตัวนั้นเป็นชนิด Transition-Activated แต่ถ้าอินเทอร์รัพท์ตัวนั้นเป็นชนิด Level-Activated แล้ว เมื่อซีพียูถูกย้ายไปทำงานที่ Interrupt Service Routine จะไม่เคลียร์บิต IEX ให้ วงจรภายนอกที่ขออินเทอร์รัพท์ซีพียูจะต้องทำหน้าที่ควบคุมสถานะของสัญญาณที่ขา INTx* ให้กลับสู่สภาพเดิมเอง

Timer0, Timer1 Interrupts

ถูกทำให้เกิดอินเทอร์รัพท์ได้โดยบิต TFO และ TF1 ซึ่งถูกเซตโดยการเปลี่ยนค่าจากเป็น 1 ทั้งหมดเป็น 0 ทั้งหมดใน Timer/Counter Register ของแต่ละตัว (ยกเว้น Timer0 ในโหมด 3) เมื่อ Timer Interrupt เกิดขึ้น Flag ที่ทำให้เกิดอินเทอร์รัพท์จะถูกเคลียร์โดยฮาร์ดแวร์ทันทีเมื่อซีพียูถูกย้ายไปทำงานในส่วน Service Routine

Serial Port Interrupt

สามารถสร้างสัญญาณอินเทอร์รัพท์ได้ โดยสัญญาณที่จะทำให้เกิดอินเทอร์รัพท์ได้มาจากบิต TI หรือ RI โดยนำมาผ่านเกตแบบ OR และ Flag ทั้งสองตัวนี้จะไม่ถูกเคลียร์โดยฮาร์ดแวร์เมื่อซีพียูทำงานในส่วน Service Routine เพราะจริง ๆ แล้วใน Service Routine จะต้องหาว่า RI หรือ TI เป็นตัวทำให้เกิดอินเทอร์รัพท์ขึ้น และบิตนั้นจะถูกเคลียร์ได้โดยซอฟต์แวร์เท่านั้น

ใน 8052 จะมี Timer2 เพิ่มขึ้นมาอีก 1 ตัว ซึ่ง Timer2 สามารถทำให้เกิดการอินเทอร์รัพท์ได้ โดยนำบิต TF2 และ EXF2 มาผ่านเกต OR โดย Flag ทั้งสองตัวนี้จะไม่ถูกเคลียร์โดยฮาร์ดแวร์ เมื่อซีพียูไปทำงานในส่วน Service Routine เหมือนการเกิดอินเทอร์รัพท์ใน Serial Port โดยใน Service Routine จะต้องหาว่า TF2 หรือ EXF2 เป็นตัวทำให้เกิดการอินเทอร์รัพท์และบิตที่ทำให้เกิดอินเทอร์รัพท์จะต้องได้รับ

การเคลียร์โดยซอฟต์แวร์เองด้วย

บิตทั้งหมดที่ทำให้เกิดสัญญาณอินเทอร์รัพท์สามารถถูกเซ็ทหรือเคลียร์โดยซอฟต์แวร์ โดยมีผลเหมือนกับว่ามันถูกเซ็ทหรือถูกเคลียร์โดยฮาร์ดแวร์ นั่นคือ อินเทอร์รัพท์ทั้งหมดที่กล่าวมาสามารถ Enable ให้เกิดสัญญาณอินเทอร์รัพท์ที่กำลังรอการบริการจากซีพียูอยู่สามารถถูกยกเลิกได้ด้วยซอฟต์แวร์

Interrupt Source แต่ละชนิดเหล่านี้ สามารถถูก Enable เป็นตัว ๆ หรือถูก Disable เป็นตัว ๆ โดยการเปลี่ยนแปลงบิตใน SFR IE ซึ่งแต่ละบิตมีความหมายดังในรูปที่ 3.15 และจะเห็นได้ว่าในรีจิสเตอร์ IE จะมีบิตที่เป็น Global Disable Bit (EA) อยู่ด้วย ซึ่งสามารถ Disable Interrupt ทั้งหมดได้ในครั้งเดียว

7	6	5	4	3	2	1	0
EA	—	ET2	ES	ET1	EX1	ET0	EX0

THE INTERRUPT ENABLE (IE) SPECIAL FUNCTION REGISTER

Bit	Symbol	Function
7	EA	Enable interrupts bit. Cleared to 0 by program to disable all interrupts; set to 1 to permit individual interrupts to be enabled by their enable bits.
6	—	Not implemented.
5	ET2	Reserved for future use.
4	ES	Enable serial port interrupt. Set to 1 by program to enable serial port interrupt; cleared to 0 to disable serial port interrupt.
3	ET1	Enable timer 1 overflow interrupt. Set to 1 by program to enable timer 1 overflow interrupt; cleared to 0 to disable timer 1 overflow interrupt.
2	EX1	Enable external interrupt 1. Set to 1 by program to enable $\overline{INT1}$ interrupt; cleared to 0 to disable $\overline{INT1}$ interrupt.
1	ET0	Enable timer 0 overflow interrupt. Set to 1 by program to enable timer 0 overflow interrupt; cleared to 0 to disable timer 0 overflow interrupt.
0	EX0	Enable external interrupt 0. Set to 1 by program to enable $\overline{INT0}$ interrupt; cleared to 0 to disable $\overline{INT0}$ interrupt.

Bit addressable as IE.0 to IE.7

รูปที่ 3.16 IE : Interrupt Enable

สังเกตว่าในรูปที่ 3.16 ตำแหน่งบิต IE6, IE5 ไม่ถูกใช้ใน 8051 เพราะถูกสงวนไว้ใช้ใน Microcontroller MCS-51 ตัวอื่น ซึ่งมีคุณสมบัติเพิ่มขึ้น ดังนั้นซอฟต์แวร์ของผู้ใช้ไม่ควรจะมีคำสั่งเขียนค่า 1 ลงไปในบิตตำแหน่งเหล่านี้ เพื่อให้ซอฟต์แวร์นั้นสามารถใช้กับไอซี Microcontroller ตัวใหม่ ๆ ในตระกูลนี้ได้นั่นเอง

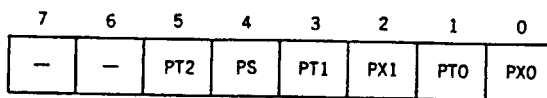
Priority Level Structure

แต่ละ Interrupt Source สามารถถูกเลือกระดับความสำคัญได้ 2 ระดับ โดยการเซ็ทหรือเคลียร์บิตในรีจิสเตอร์ IP ดังในรูปที่ 3.17

เอกสารนี้เป็นเอกสารระดับความสำคัญของการขออินเทอร์รัพท์ที่มีได้ 2 ระดับได้แก่ระโยชน์ด้านการค้าไม่ว่ากรณีใด Low Priority Interrupt สามารถถูก Interrupted ที่โดยนำ High

Priority Interrupt ได้ แต่ไม่สามารถถูก Interrupted โดย Low Priority Interrupted ตัวอื่น ๆ ได้

High Priority Interrupt ไม่สามารถถูก Interrupted โดย Interrupt Source ตัวอื่นได้เลย



THE INTERRUPT PRIORITY (IP) SPECIAL FUNCTION REGISTER

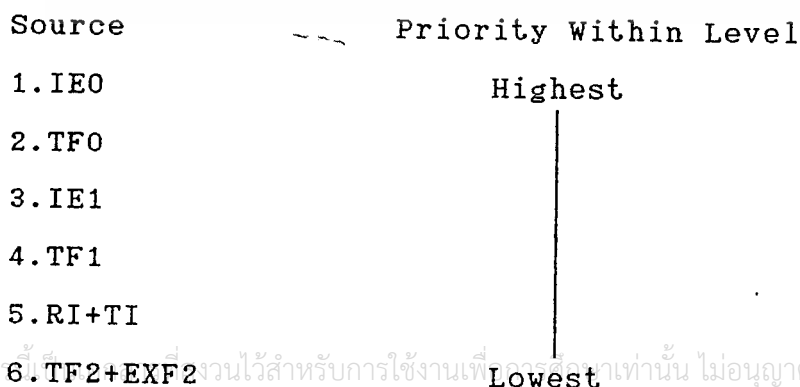
Bit	Symbol	Function
7	—	Not implemented.
6	—	Not implemented.
5	PT2	Reserved for future use.
4	PS	Priority of serial port interrupt. Set/cleared by program.
3	PT1	Priority of timer 1 overflow interrupt. Set/cleared by program.
2	PX1	Priority of external interrupt 1. Set/cleared by program.
1	PT0	Priority of timer 0 overflow interrupt. Set/cleared by program.
0	PX0	Priority of external interrupt 0. Set/cleared by program.

Note: Priority may be 1 (highest) or 0 (lowest)

Bit addressable as IP.0 to IP.7

รูปที่ 3.17 IP : Interrupt Priority Register

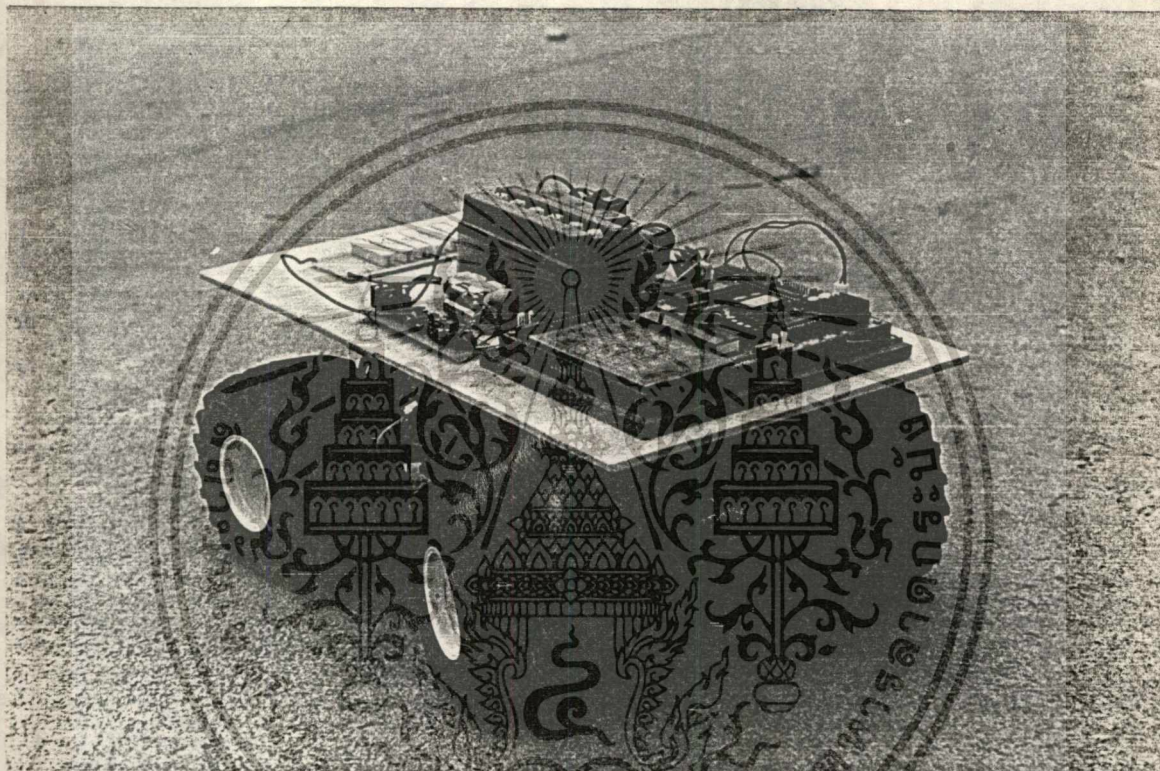
ถ้ามีการขออินเทอร์รัพท์พร้อมกัน 2 ชนิด ซึ่งมีระดับความสำคัญในการขออินเทอร์รัพท์ไม่เท่ากัน อินเทอร์รัพท์ที่มีระดับความสำคัญสูงกว่า (High Priority Interrupt) จะได้รับการบริการก่อน แต่ถ้ามีการขออินเทอร์รัพท์พร้อมกัน 2 ชนิด ซึ่งมีระดับความสำคัญเท่ากัน ลำดับการรับสัญญาณอินเทอร์รัพท์ภายในจะกำหนดเองว่า อินเทอร์รัพท์ชนิดใดจะถูกบริการก่อน ดังนั้นภายในระดับความสำคัญของการขออินเทอร์รัพท์หนึ่ง ๆ จะมี Second Priority Structure (ระดับความสำคัญในการขออินเทอร์รัพท์ย่อย) ดังนี้



เอกสารนี้เป็นทรัพย์สินของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สังเกตว่า Priority Within Level Structure จะถูกใช้เพียงเพื่อแก้ปัญหาการเกิดสัญญาณอินเทอร์รัพท์พร้อมกันขึ้นมาเท่านั้น (Simultaneous Request of the Same Priority Level)

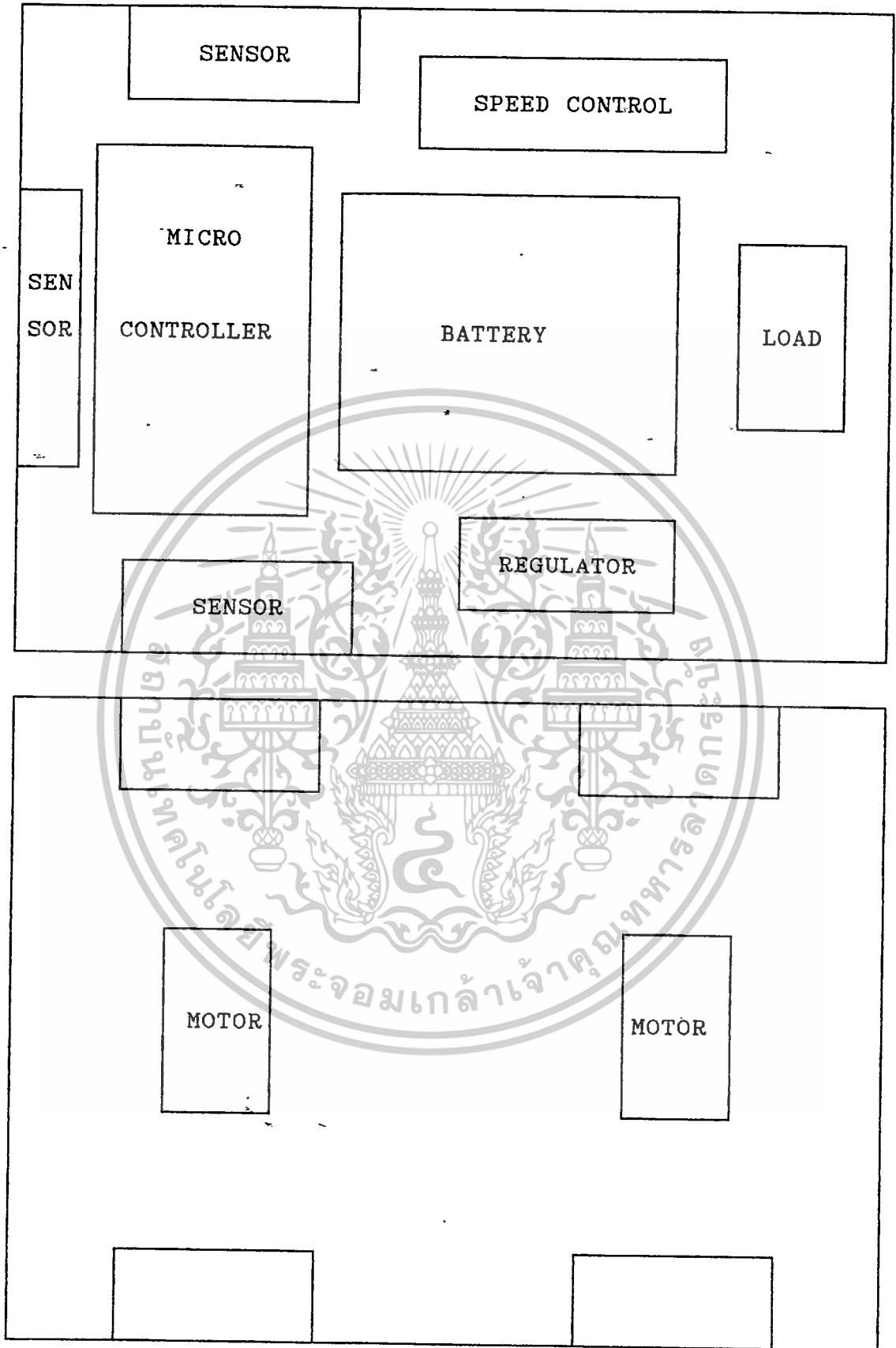
IP Register มีบิตที่ไม่ถูกใช้งานอยู่จำนวนหนึ่ง คือ IP.7 และ IP.6 โดยวางทั้งใน 8051 และ 8052 โดยใน 8051 จะมีบิตที่ว่างเพิ่มมาอีก 1 บิต คือ IP.5 และเช่นเดียวกันซอฟต์แวร์ของผู้ใช้ไม่ควรมีการเขียนค่า 1 ไปที่ตำแหน่งบิตเหล่านี้ เพราะมันอาจถูกนำไปใช้ใน MCS-51 ตัวใหม่ ๆ ในอนาคตต่อไป



รูปที่ 3.18 ภาพถ่ายตัวจริงของโครงงานนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 ตำแหน่งการจัดวางอุปกรณ์



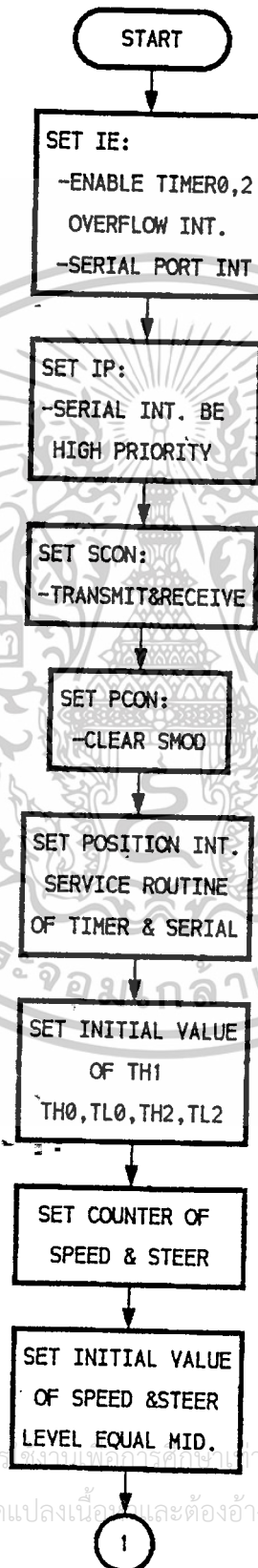
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น รูปที่ 3.19 ตำแหน่งการจัดวางอุปกรณ์ต่าง ๆ เอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

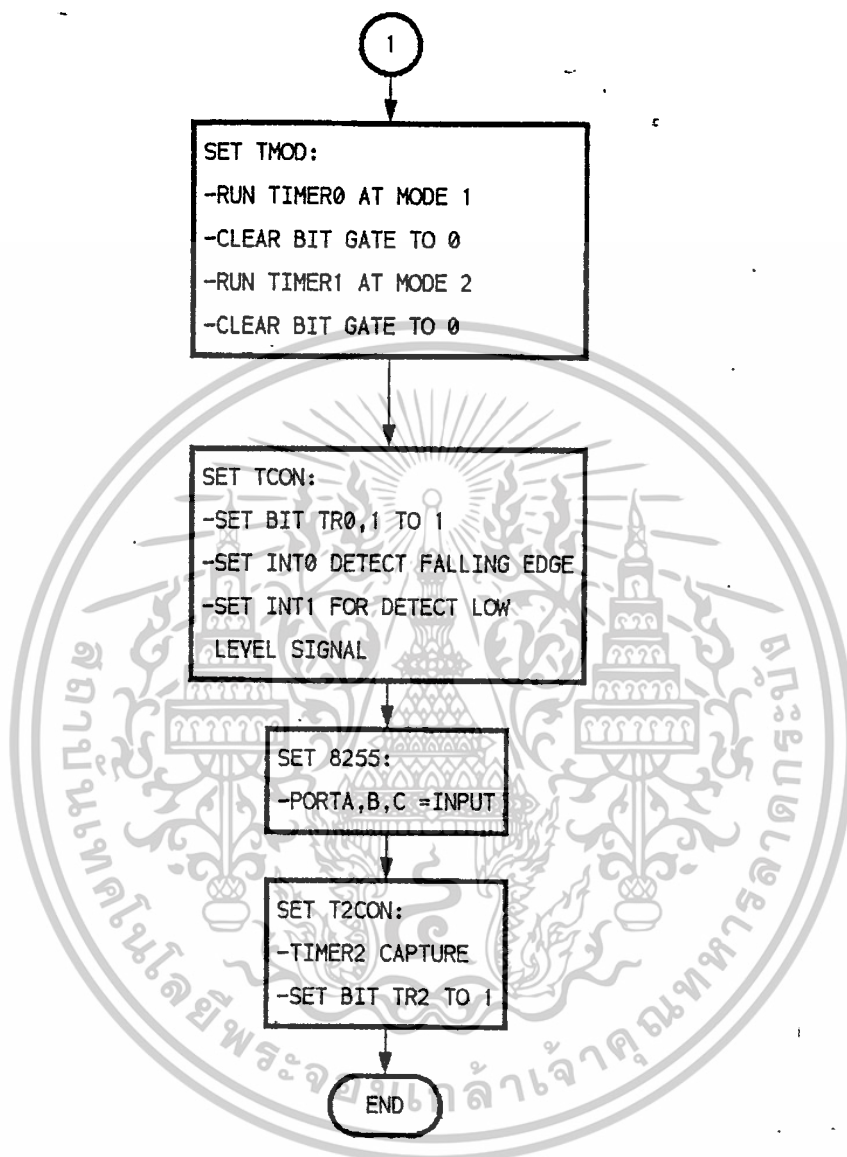
โปรแกรมควบคุมการทำงาน

4.1 Flow Chart

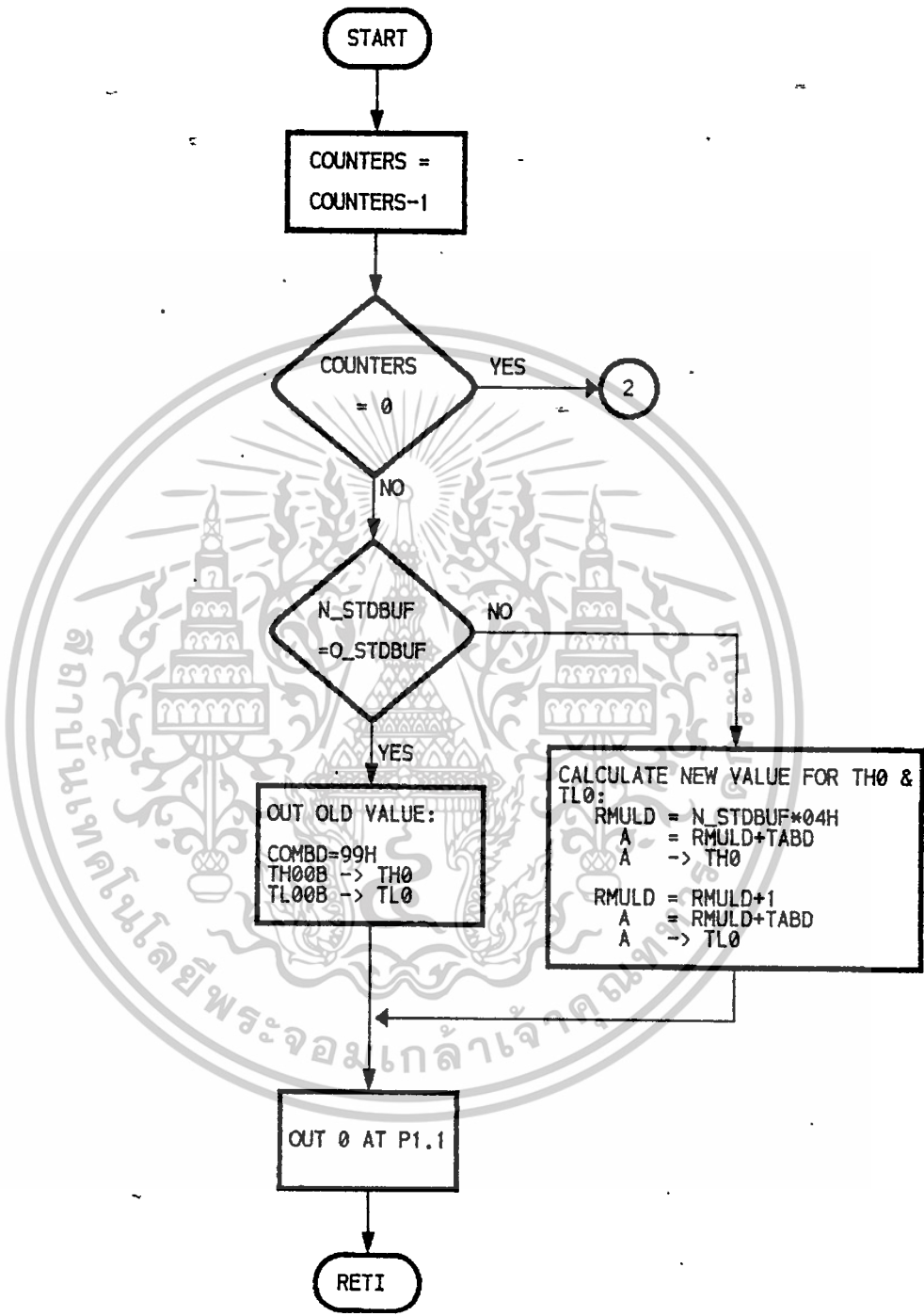
MAIN PROGRAM



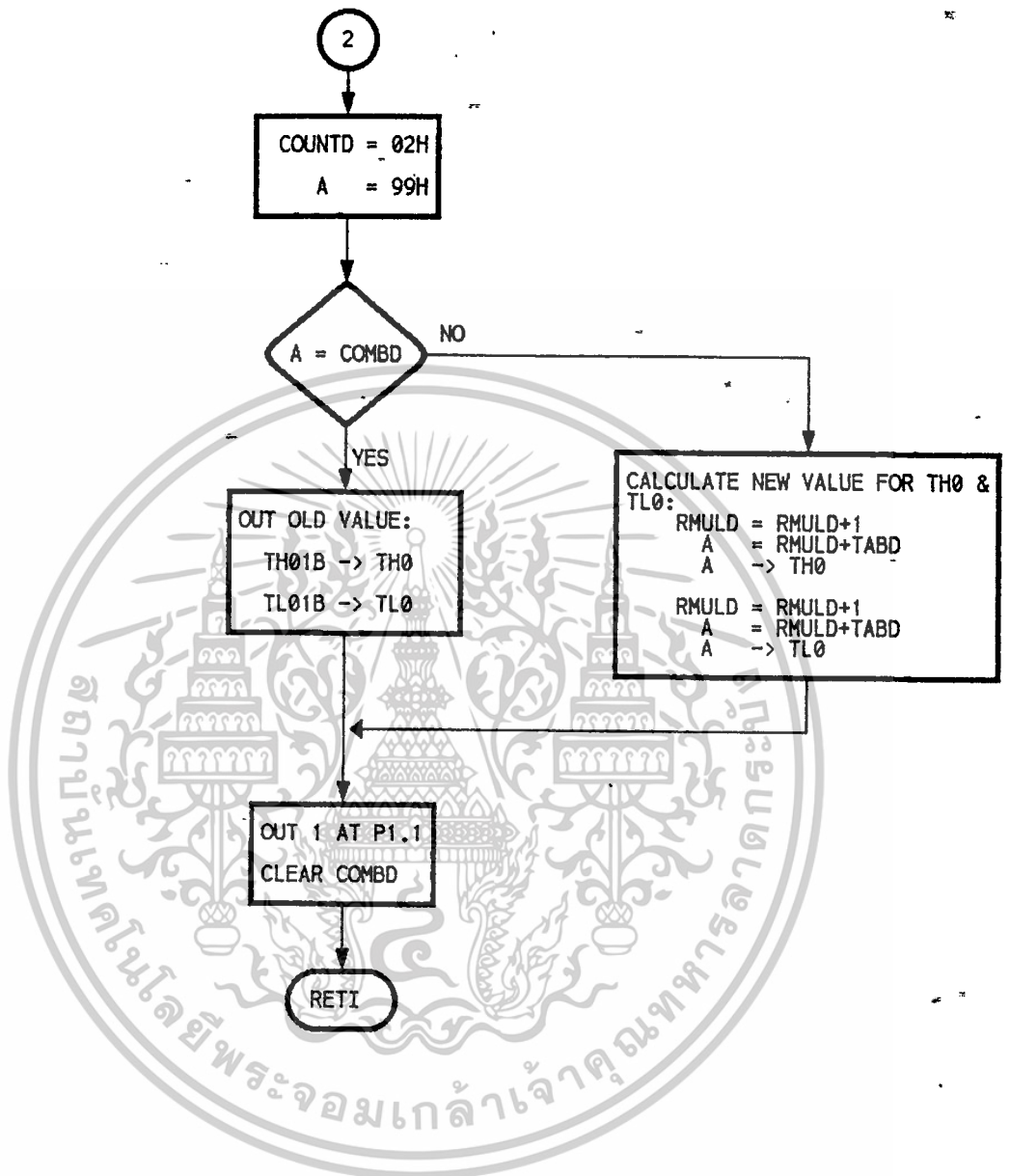
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาระบบและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

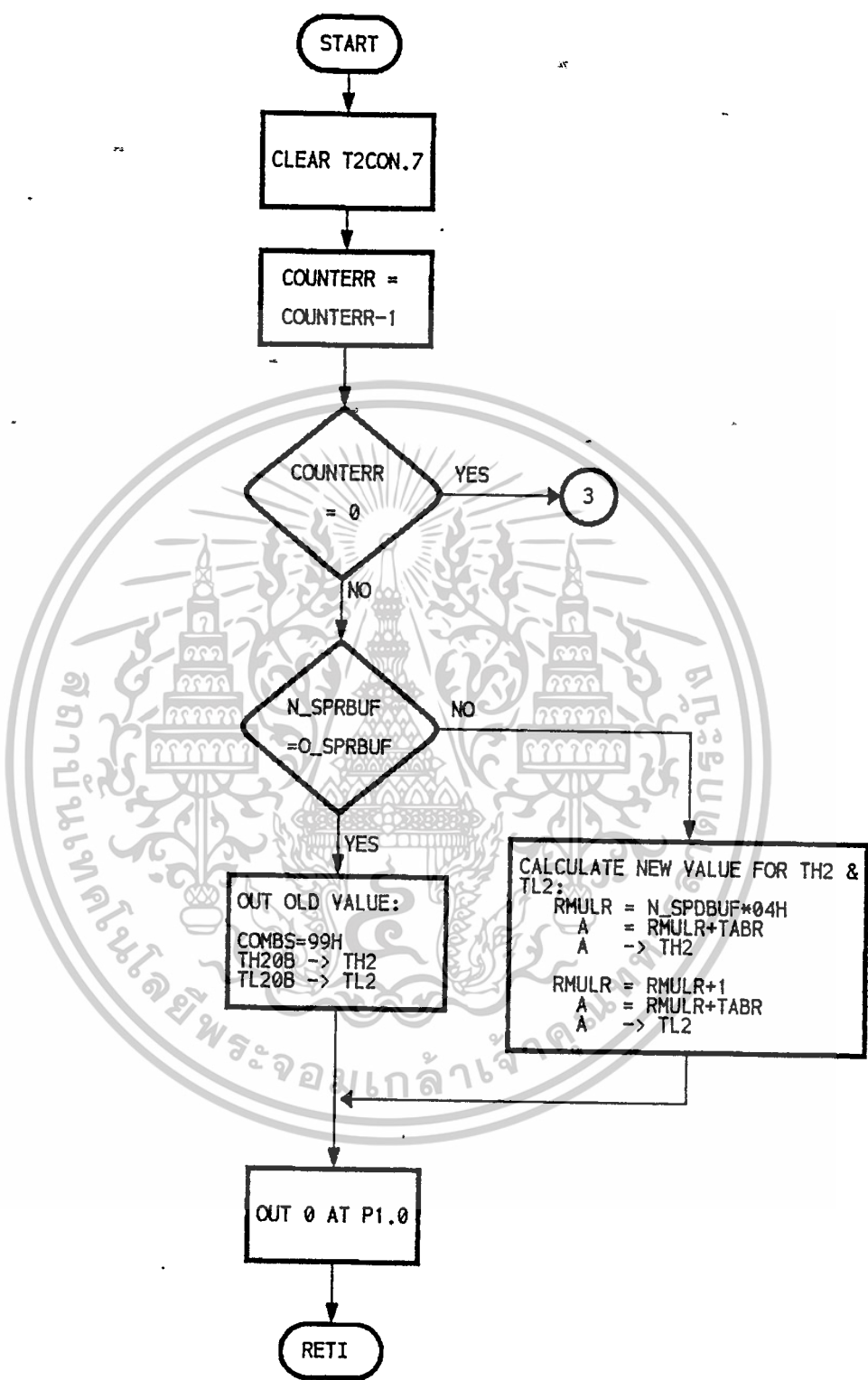


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

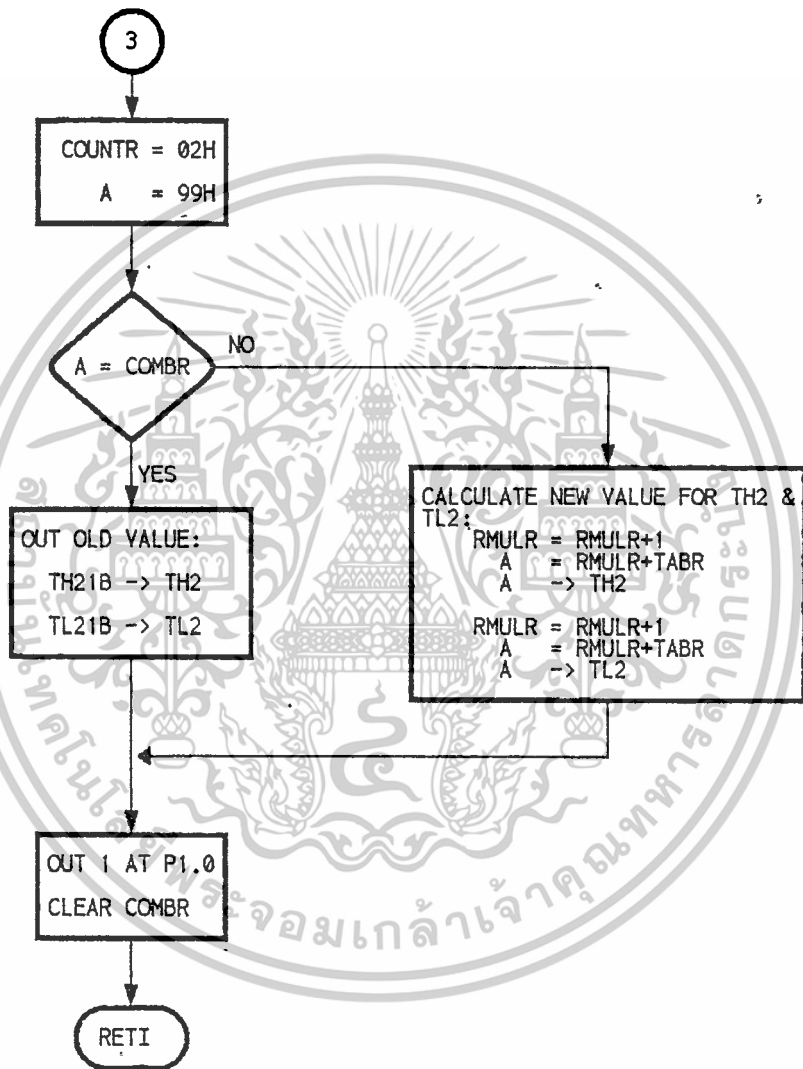


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

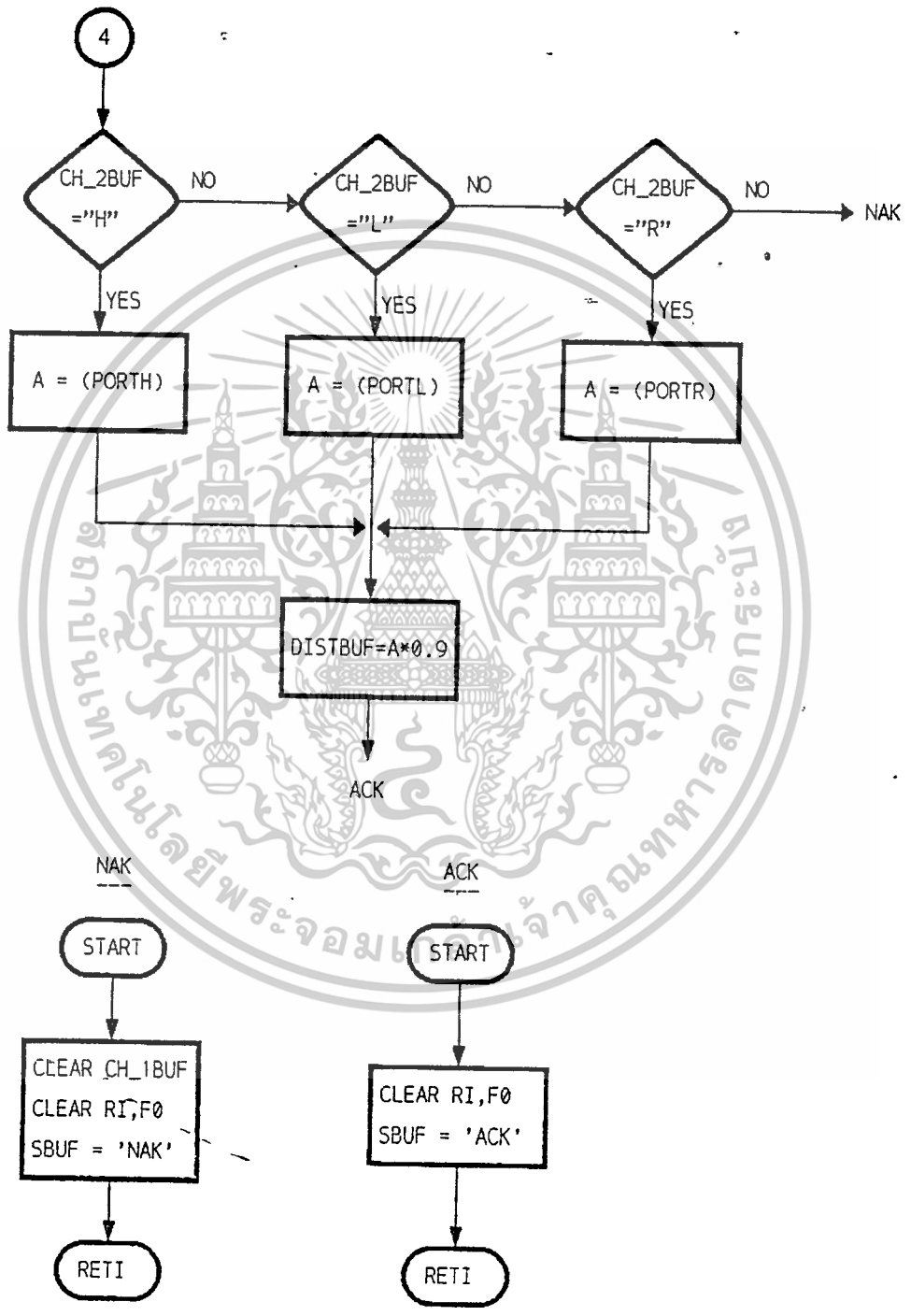
TIMER2 INTERRUPT



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 คำอธิบายโปรแกรม

Main Program

ในส่วนของโปรแกรมหลักนี้จะเป็นการตั้งค่า และกำหนดค่าเริ่มต้นของ พารามิเตอร์ต่าง ๆ ที่จะใช้ในขบวนการรับส่งข้อมูล และใช้ในการทำงานของ Timer0 และ Timer2 โดยค่าพารามิเตอร์ต่าง ๆ แสดงดังนี้

1) ตั้งค่ารีจิสเตอร์ IE กำหนดให้ทำการ Enable สัญญาณ Interrupt จาก Timer0,2 และ Serial Port

2) ตั้งค่ารีจิสเตอร์ IP กำหนดให้ Interrupt จาก Serial Port มีความสำคัญสูงสุด

3) ตั้งค่ารีจิสเตอร์ SCON กำหนดให้ขบวนการรับส่งข้อมูลแบบอนุกรมทำงานในโหมด 1 และ Enable การรับข้อมูล

4) ตั้งค่ารีจิสเตอร์ PCON กำหนดให้บิต SMOD เป็น 0 เพื่อนำไปใช้ในการสร้าง Baud Rate

5) กำหนดค่าตำแหน่ง Interrupt Service Routine ของ Timer 0,2 และ Serial Port

6) ตั้งค่ารีจิสเตอร์ TH1 กำหนด Baud Rate = 2,400 บิตต่อวินาที

7) กำหนดค่าเริ่มต้นให้กับ TH0, TLO, TH2 และ TL2

8) กำหนดให้ตัวนับของความเร็วและมุมเลี้ยว เท่ากับ 2 เพื่อจะได้ทราบว่า จะส่ง 0 หรือ 1 ออก

9) กำหนดค่าเริ่มต้นของระดับความเร็วและมุมเลี้ยว ให้อยู่ที่ค่ากลางคือ รถไม่วิ่งและล้อตรงไม่มีการเลี้ยว

10) ตั้งค่าในรีจิสเตอร์ TMOD กำหนดให้ Timer0 ทำงานในโหมด 1 และ Timer1 ทำงานในโหมด 2 และกำหนดบิต GATE = 0 เพื่อให้ Timer ถูกควบคุมด้วยบิต TR0,1 เท่านั้น

11) ตั้งค่าในรีจิสเตอร์ TCON กำหนดให้ TR0,1 เพื่อ Enable ให้ Timer0, 1 เริ่มทำงานและตั้งค่าบิต ITO = 1 เพื่อให้ Interrupt ทันทีเมื่อมีสัญญาณขอบขาลงมา และ IT1 = 0 เพื่อให้ Interrupt ทันที เมื่อ Low Level เข้ามา

12) กำหนดค่า Control Port ของ 8255 ให้เป็น Input ทั้ง 3 Port

13) ตั้งค่าในรีจิสเตอร์ T2CON กำหนดให้ Timer2 ทำงานใน Capture Mode และให้ TR2 = 1 เพื่อให้ Timer2 เริ่มทำงาน

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด โปรแกรมในส่วนนี้จัดจะเป็นโปรแกรมสร้างสัญญาณพัลส์ที่มีความถี่กว้างแตกต่างกัน

กันขึ้นมา ซึ่งขึ้นอยู่กับระดับความเร็วที่ผู้ใช้ป้อนให้ โดยใช้หลักการให้ Timer0 เป็นตัวตั้ง เวลาที่จะส่งค่า 0 หรือ 1 ออกที่ขาของ P1.1 ซึ่งเราจะเก็บค่าที่จะ Load ให้กับ TH0 และ TLO ไว้ในตาราง ในตอนแรกจะทำการตรวจสอบว่า เป็นการอินเทอร์รัพท์ให้ส่ง 0 หรือส่ง 1 ออกไป ซึ่งเราจะทำการส่ง 0 ออกมาก่อน แล้วจึงส่ง 1 ถ้าเป็นการส่ง 0 ออก จะต้องมีการเปรียบเทียบระดับความเร็วว่ายังคงมีค่าเท่าเดิมหรือไม่ ถ้าไม่เท่าจะมีการคำนวณหาตำแหน่งเริ่มต้นในตาราง ซึ่งเก็บค่าที่จะส่งให้ Timer ใหม่ แต่ถ้าเท่าจะนำค่าเดิมให้กับ Timer0 แล้วจึงส่ง 0 ออก เมื่อ Timer มีสัญญาณอินเทอร์รัพท์มาอีกครั้ง ก็จะเป็นการส่ง 1 ออก ซึ่งการส่ง 1 จะต้องอาศัยข้อมูลที่ได้จากการเปรียบเทียบในตอนแรกมาใช้ในการตัดสินใจว่า จะนำค่าเดิมให้กับ Timer หรือนำค่าใหม่ในตารางให้กับ Timer0 จากนั้นจึงส่ง 1 ออก ทำสลับกันเช่นนี้ไปเรื่อย ๆ

Subroutine Steer

โปรแกรมในส่วนนี้ มีหลักการเช่นเดียวกับการสร้างสัญญาณพัลส์ควบคุมความเร็ว แต่จะมีข้อแตกต่างกันตรงที่โปรแกรมนี้จะใช้ Timer2 เป็นตัวตั้งเวลา และจะส่งค่า 0 และ 1 ออกที่ P1.0

Subroutine SINT(Serial Interrupt)

การรับส่งข้อมูลจากคอมพิวเตอร์นั้น จะทำการรับส่งแบบอนุกรม โดยคำสั่งที่ได้รับจากคอมพิวเตอร์จะมีขนาด 2 ไบต์ โดยในไบต์แรกจะกำหนดชนิดของคำสั่ง ซึ่งมีอยู่ 3 ชนิด คือ ความเร็ว การเลี้ยว และระยะห่างของสิ่งกีดขวาง ส่วนไบต์ที่สองจะเป็นขนาดของความเร็ว ขนาดของวงเลี้ยว ซึ่งค่าทั้ง 2 อย่างนี้จะมี 7 ระดับ และค่าระยะห่างของสิ่งกีดขวางว่า ต้องการอ่านด้านหน้า ด้านซ้าย หรือด้านขวา

เมื่อได้รับคำสั่งและแปลงคำสั่งนั้นแล้ว ถ้าคำสั่งนั้นถูกต้องก็จะส่งสัญญาณ ACK ตอบกลับไปยังคอมพิวเตอร์ ถ้าถามระยะทางมาก็จะส่งขนาดของระยะทางไปด้วย แต่ถ้าคำสั่งที่ส่งมาไม่ถูกต้องก็จะส่งสัญญาณ NAK กลับไปยังคอมพิวเตอร์

โปรแกรมในส่วนนี้ จะเป็นการตอบสนองการอินเทอร์รัพท์ที่เกิดขึ้นเนื่องจากการติดต่อกับภายนอกแบบอนุกรม โดยเมื่อมีการอินเทอร์รัพท์นี้เกิดขึ้นก็จะตรวจสอบว่า เป็นการอินเทอร์รัพท์จากการรับหรือว่าการส่งข้อมูล ถ้าเป็นการรับข้อมูลก็จะตรวจสอบว่าข้อมูลนั้นมาครบ 2 ไบต์หรือยังถ้ายังก็จะเก็บข้อมูลไบต์แรกไว้ก่อน แล้วรอรับข้อมูลไบต์ที่สอง แต่ถ้าครบ 2 ไบต์แล้ว ก็จะนำข้อมูลไบต์แรกมาตรวจสอบว่าเป็นคำสั่ง ความเร็ว การเลี้ยว หรือถามระยะวัตถุ (ถ้านอกเหนือจากนี้ก็จะตอบ NAK กลับไป) แล้วนำข้อมูลไบต์ที่สองมาตรวจสอบว่าอยู่ในขอบเขตที่กำหนดหรือไม่ ถ้าอยู่ก็นำข้อมูลนั้นเก็บลงในหน่วยความจำเพื่อนำไปปฏิบัติตาม ณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

ผลการทดลอง

การควบคุมความเร็ว

1. ขณะที่ไม่มีวัตถุอยู่บนตัวรถสำรวจ แต่มีความต้านทานเท่ากับ 1.00 โห้ท้ม ต่ออยู่และมีแบตเตอรี่ขนาด 12 โวลท์ 9 แอมป์ ทดลองขณะเพิ่งชาร์จกระแสใหม่ คาบ เวลาของสัญญาณมีค่าคงที่เท่ากับ 19 มิลลิวินาที

ความกว้างของสัญญาณพัลส์ที่ป้อนให้ (มิลลิวินาที)	ความเร็วของมอเตอร์ (รอบต่อนาที)	ทิศทางการเคลื่อนที่
1.15	260	ถอยหลัง
1.20	245	ถอยหลัง
1.25	235	ถอยหลัง
1.30	-	-
1.35	-	-
1.40	-	-
1.45	-	-
1.50	-	-
1.55	-	-
1.60	-	-
1.65	-	-
1.70	400	เดินหน้า
1.75	475	เดินหน้า
1.80	515	เดินหน้า

ตารางที่ 5.1 แสดงความเร็วของมอเตอร์ขณะไม่มีวัตถุวางอยู่บนตัวรถ

หมายเหตุ: ที่ความกว้างของสัญญาณพัลส์ 1.3-1.45 และ 1.55-1.6 มิลลิวินาที มอเตอร์ มีเสียงดังแต่ล้อยังไม่หมุน ส่วนที่ 1.5 มิลลิวินาที ไม่มีอะไรเกิดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า จากผลการทดลองดังกล่าวข้างต้น สามารถแบ่งระดับของความเร็วได้เป็น 7 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระดับ เพื่อนำไปทำการทดลองความเร็วในขณะที่มีวัตถุอยู่บนรถต่อไป

2. ขณะมีแบตเตอรี่ขนาด 12 โวลต์ 9 แอมป์ ทดลองขณะเพ็งซาร์จกระแสใหม่ และอุปกรณ์ต่าง ๆ วางอยู่บนตัวรถและมีความต้านทาน 1.25 โอห์ม ต่อกัน และสัญญาณพัลส์มีคาบเวลาคงที่ 19 มิลลิวินาที

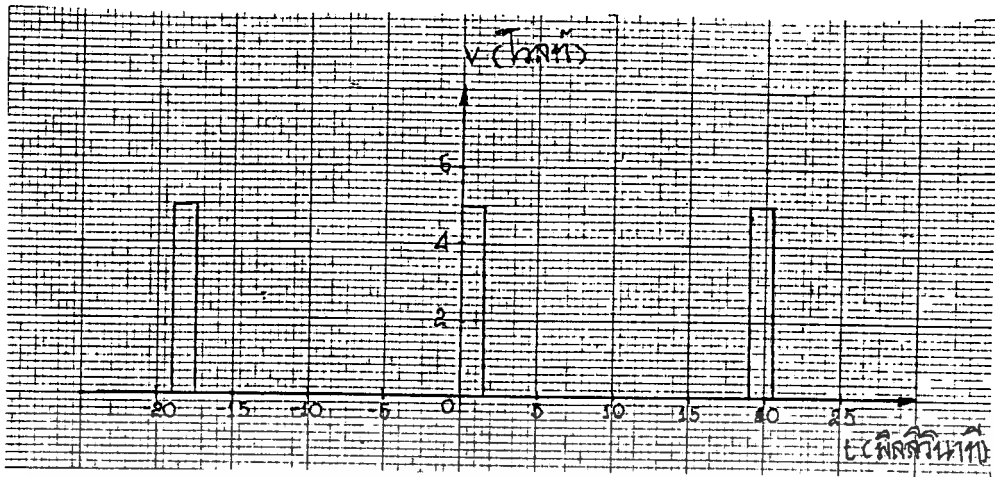
ความกว้างของสัญญาณพัลส์ที่ป้อนให้ (มิลลิวินาที)	ความเร็วของรถ (เมตรต่อวินาที)	ทิศทางการเคลื่อนที่
1.15	0.60	ถอยหลัง
1.20	0.35	ถอยหลัง
1.25	0.20	ถอยหลัง
1.50	0	-
1.70	0.4	เดินหน้า
1.75	0.65	เดินหน้า
1.80	1.00	เดินหน้า

ตารางที่ 5.2 แสดงความเร็วของรถในขณะที่มีน้ำหนักวางอยู่บนตัวรถ

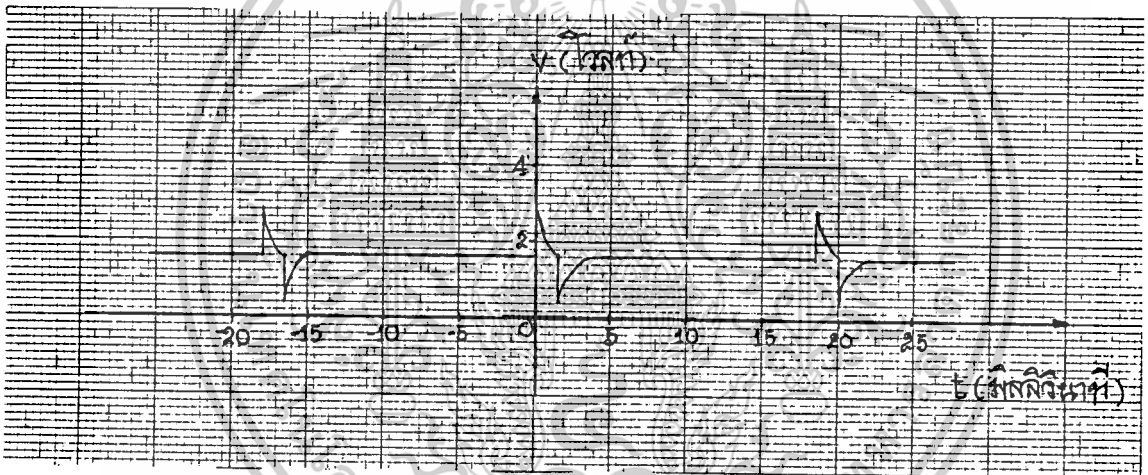
การบังคับเลี้ยว

ความกว้างของสัญญาณพัลส์ที่ป้อนให้ (มิลลิวินาที)	รัศมีของการเลี้ยว (เซนติเมตร)	ทิศทางการเลี้ยว
0.40	57	ซ้าย
0.75	62	ซ้าย
1.10	112	ซ้าย
1.35	0	ตรงไปข้างหน้า
1.60	92	ขวา
1.95	56	ขวา
2.30	44	ขวา

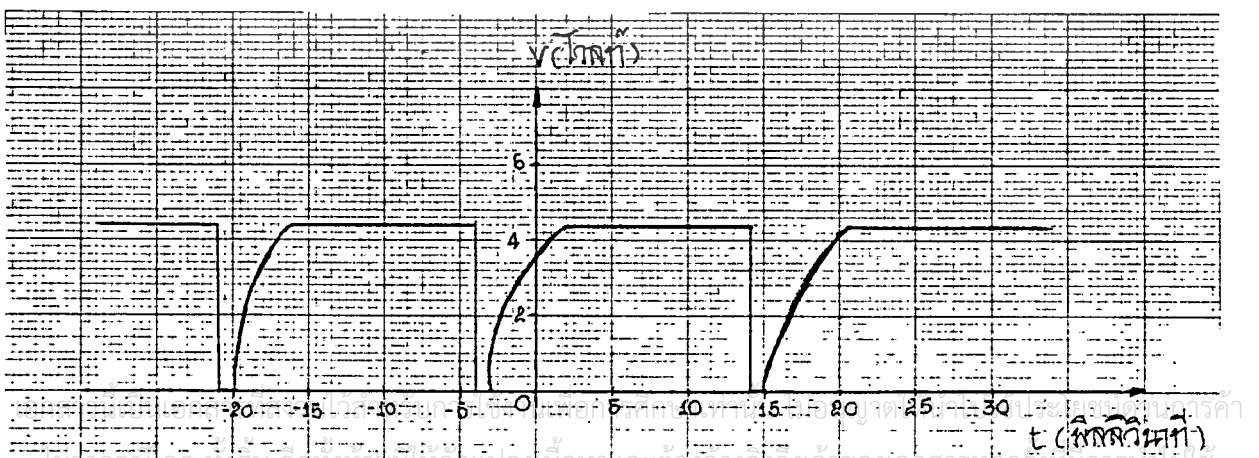
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมออนุญาตเห็นำไปใช้ประโยชน์ด้านการค้า
 ตารางที่ 5.3 แสดงรัศมีของการเลี้ยวที่ค่าความกว้างของสัญญาณพัลส์ต่าง ๆ มีการนำป้ใช้



รูปที่ 5.1 รูปของสัญญาณอินพุท

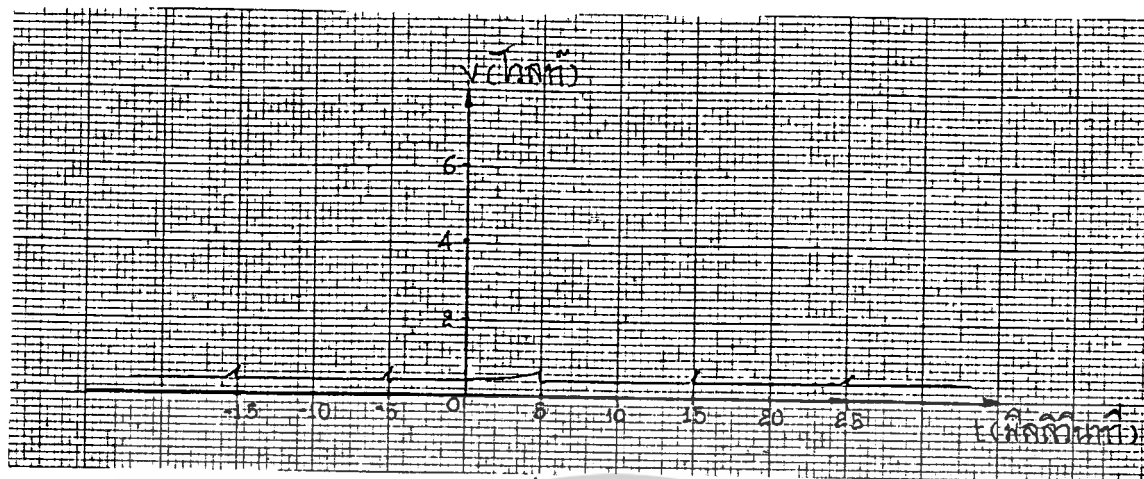


รูปที่ 5.2 รูปของสัญญาณที่ขา 4 ของ IC1

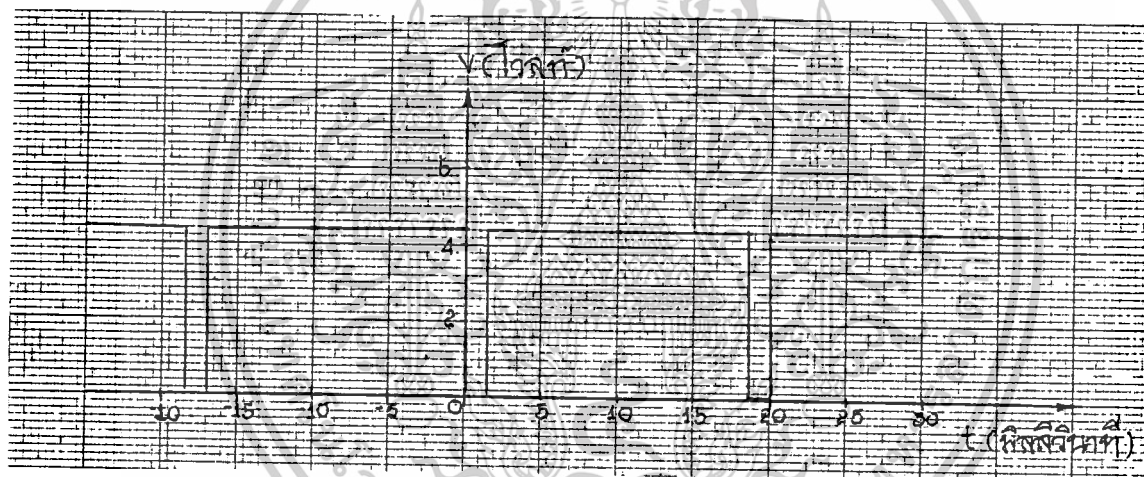


รูปที่ 5.3 รูปของสัญญาณที่ขา 2 ของ IC1

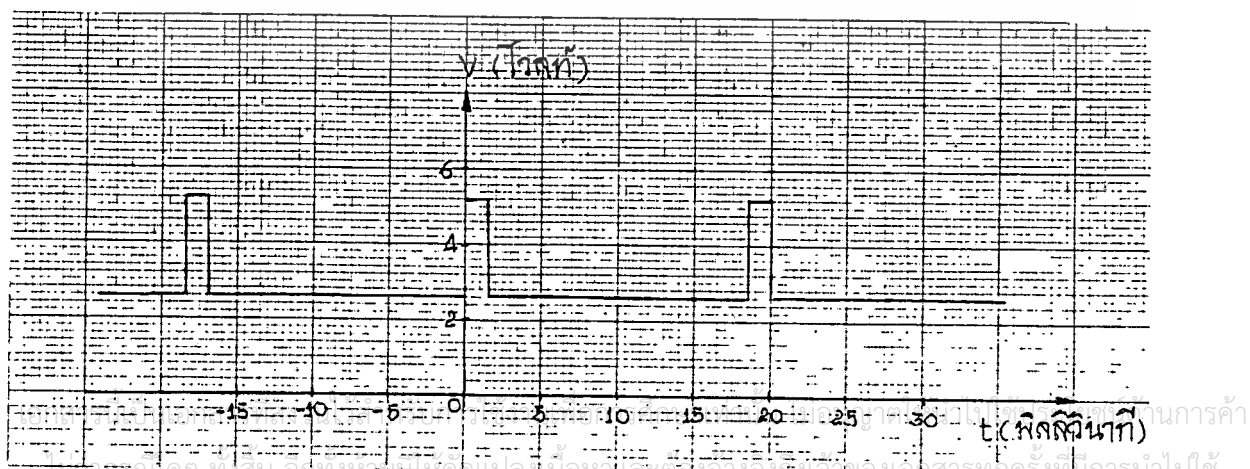
แม้ว่ากรรมใดๆ ทั้งสิ้น อี... ทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



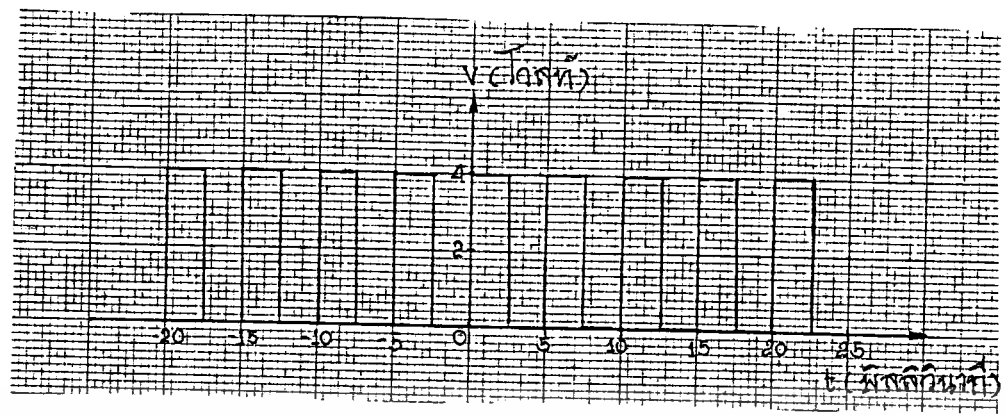
รูปที่ 5.4 รูปของสัญญาณที่ขา 11 ของ IC1



รูปของสัญญาณที่ขา 13 ของ IC1

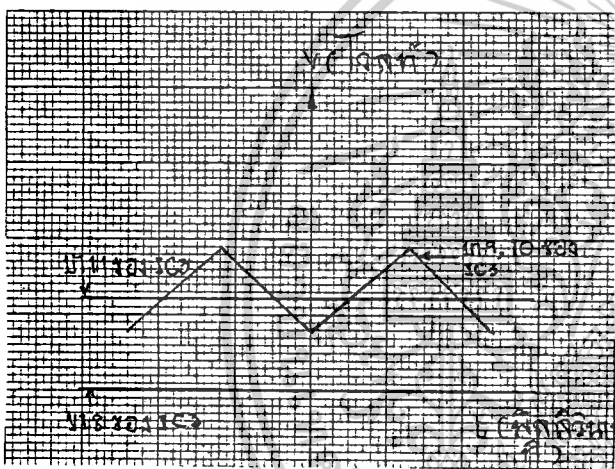


รูปที่ 5.5 รูปของสัญญาณที่ขาคอลเลคเตอร์ของ 2N3906

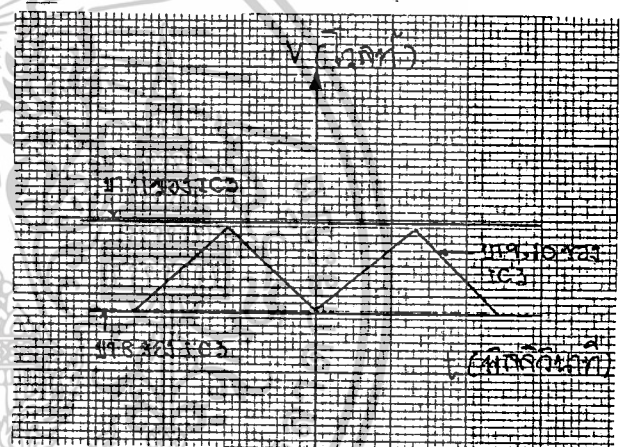


รูปที่ 5.6 รูปของสัญญาณที่ขา 8 ของ IC4

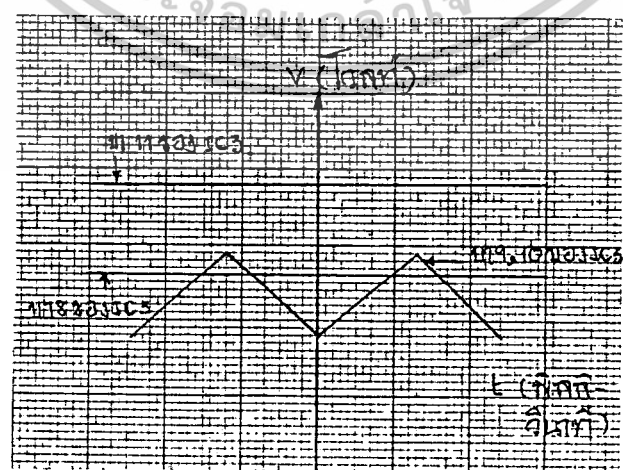
รูปที่ 5.7



รูปที่ 5.8

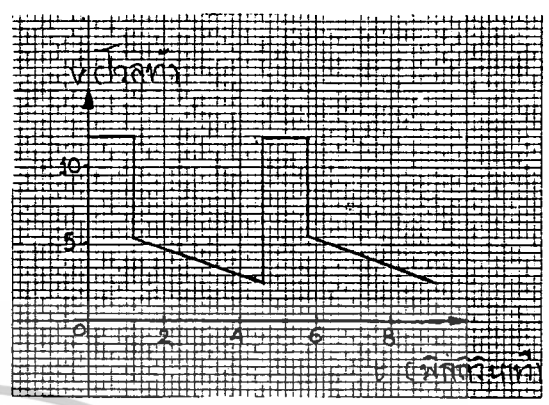
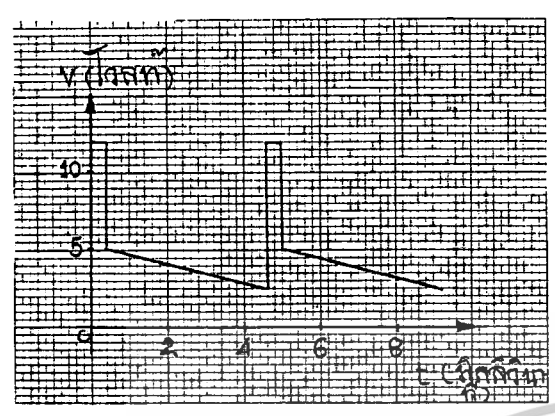


ความกว้างของสัญญาณพัลส์ เท่ากับ 1.30 มิลลิวินาที ความกว้างของสัญญาณพัลส์ เท่ากับ 1.50 มิลลิวินาที

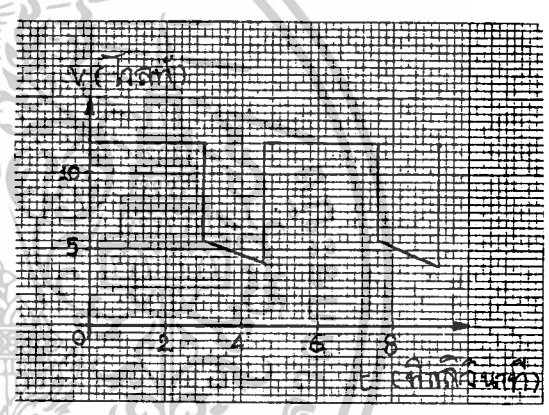
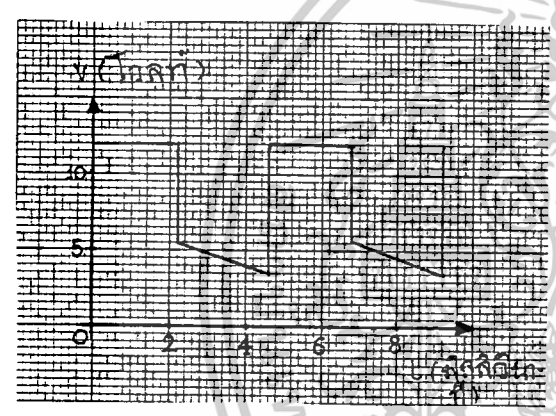


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ารูปที่ 5.9 ความกว้างของสัญญาณพัลส์ เท่ากับ 1.70 มิลลิวินาที ทุกครั้งที่มีการนำไปใช้

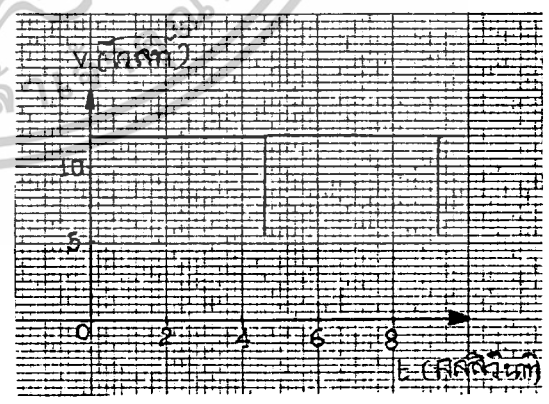
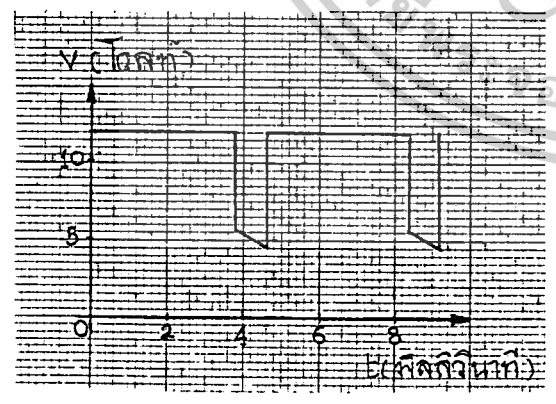
รูปที่ 5.10 รูปของสัญญาณที่ต่อเข้ากับมอเตอร์



ความกว้างของสัญญาณพัลส์ เท่ากับ 1.55 มิลลิวินาที ความกว้างของสัญญาณพัลส์ เท่ากับ 1.60 มิลลิวินาที

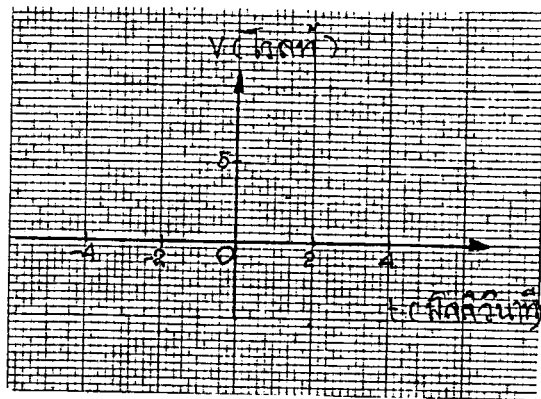


ความกว้างของสัญญาณพัลส์ เท่ากับ 1.65 มิลลิวินาที ความกว้างของสัญญาณพัลส์ เท่ากับ 1.70 มิลลิวินาที

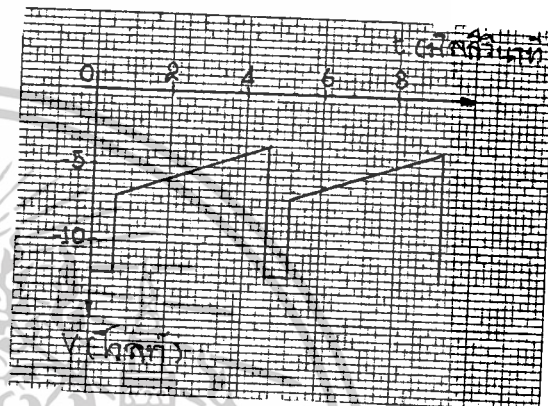
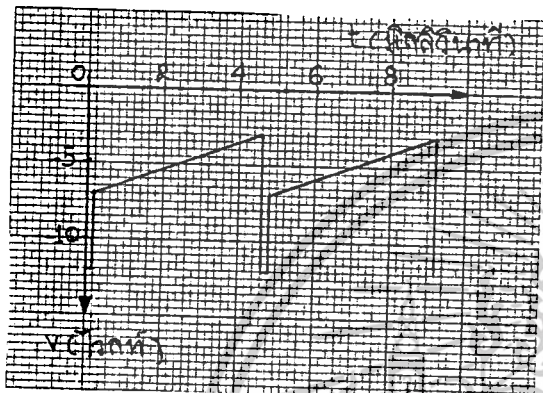


ความกว้างของสัญญาณพัลส์ เท่ากับ 1.75 มิลลิวินาที ความกว้างของสัญญาณพัลส์ เท่ากับ 1.80 มิลลิวินาที

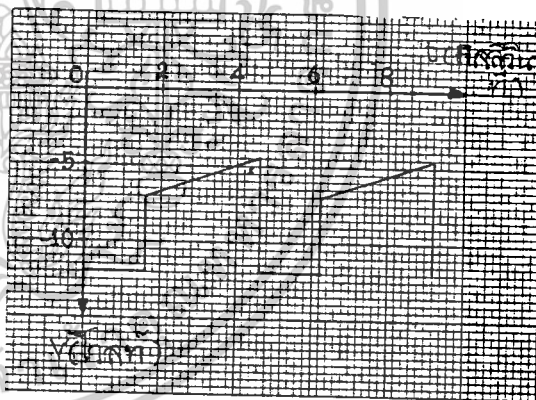
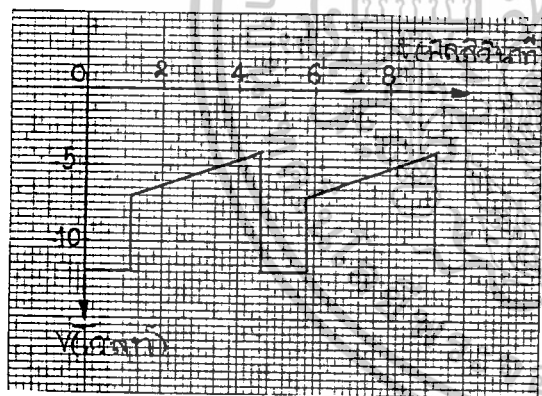
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



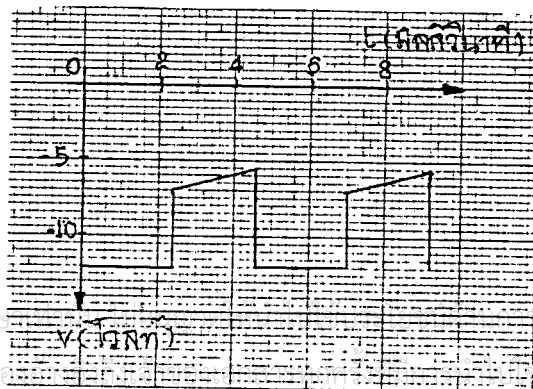
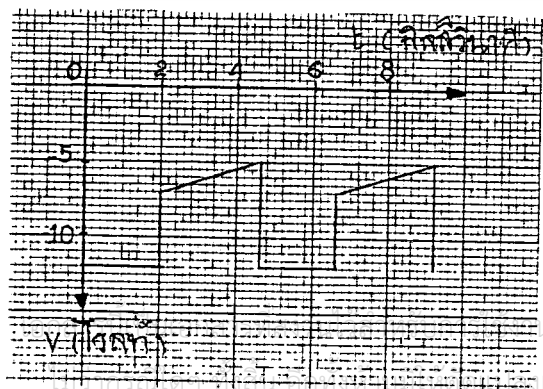
ความกว้างของสัญญาณพัลส์ เท่ากับ 1.50 มิลลิวินาที



ความกว้างของสัญญาณพัลส์ เท่ากับ 1.40 มิลลิวินาที ความกว้างของสัญญาณพัลส์ เท่ากับ 1.35 มิลลิวินาที



ความกว้างของสัญญาณพัลส์ เท่ากับ 1.30 มิลลิวินาที ความกว้างของสัญญาณพัลส์ เท่ากับ 1.25 มิลลิวินาที



ความกว้างของสัญญาณพัลส์ เท่ากับ 1.20 มิลลิวินาที ความกว้างของสัญญาณพัลส์ เท่ากับ 1.15 มิลลิวินาที

บทที่ 6

สรุปผล ปัญหาและแนวทางแก้ไข

1. ตัวตรวจจับสิ่งกีดขวางด้วยคลื่นอัลตราโซนิก จะวัดระยะทางได้ถูกต้องแม่นยำที่ระยะใกล้สุดประมาณ 30 เซนติเมตร และระยะไกลสุดประมาณ 2 เมตร โดยจะมีค่าผิดพลาดประมาณ 1-2 เซนติเมตร ซึ่งตัวตรวจจับสิ่งกีดขวางด้วยคลื่นอัลตราโซนิกนี้จะใช้ได้ดีกับวัตถุที่มีพื้นที่สัมผัสมากและเป็นผิวเรียบ นอกจากนี้ความถูกต้องในการวัดนั้นยังขึ้นอยู่กับ อุณหภูมิห้อง การถ่ายเทอากาศ และอาจถูกรบกวนจากแหล่งกำเนิดเสียงที่มีความถี่สูงด้วย

2. การควบคุมความเร็ว ความเร็วและทิศทางการเคลื่อนที่ของตัวรถ เปลี่ยนแปลงไปตามสัญญาณพัลส์ที่ได้รับ โดยค่ากลางที่รถไม่เคลื่อนที่นั้นอยู่ที่สัญญาณพัลส์ขนาด 1.5 มิลลิวินาที และมีความเร็วเท่ากับ 19 มิลลิวินาที ถ้ามีค่ามากกว่านี้จะทำให้รถเดินหน้า และมีความเร็วเพิ่มขึ้นตามลำดับ แต่ถ้าค่าน้อยกว่านี้ จะทำให้รถถอยหลัง และความเร็วเพิ่มขึ้นตามลำดับเช่นกัน โดยที่ความเร็วเมื่อรถถอยหลัง จะเป็นครึ่งหนึ่งของความเร็วเมื่อรถเดินหน้า

นอกจากนั้น ความเร็วของรถยังขึ้นอยู่กับกระแสจากแบตเตอรี่ที่จ่ายให้กับมอเตอร์ ซึ่งกระแสของแบตเตอรี่ จะลดลงตามระยะเวลาการใช้งาน จึงทำให้ความเร็วของรถที่สัญญาณพัลส์ค่าเดียวกันมีค่าไม่เท่ากัน เนื่องจากกระแสที่จ่ายให้ไม่เท่ากัน

3. การควบคุมการบังคับเลี้ยว การบังคับเลี้ยวของตัวรถนั้นจะขึ้นอยู่กับสัญญาณพัลส์ที่ป้อนให้กับเซอร์โวมอเตอร์ โดยค่ากลางที่ล้อรถไม่มีการเลี้ยว คือ สัญญาณพัลส์ขนาด 1.35 มิลลิวินาที และมีความเร็ว 19 มิลลิวินาที ถ้ามีค่ามากกว่านี้ จะทำให้รถเลี้ยวขวา และรัศมีการเลี้ยวเพิ่มตามลำดับ แต่ถ้าค่าน้อยกว่านี้จะทำให้รถเลี้ยวซ้าย และรัศมีการเลี้ยวเพิ่มขึ้นตามลำดับเช่นกัน

จากการทดลอง รัศมีการเลี้ยวที่ได้มีค่าไม่มาก ซึ่งตรงกับที่ต้องการใช้งาน แต่ถ้าต้องการให้ได้รัศมีการเลี้ยวที่แคบกว่านี้ ต้องปรับปรุงที่ตัวแมคคานิกส์ของตัวรถ และจะเห็นได้ว่า การเลี้ยวไปทางซ้ายและขวา จะมีรัศมีการเลี้ยวไม่ใกล้เคียงกันเท่าใดนัก ทั้งนี้เนื่องมาจากว่าในการปรับ Servo Motor ไม่สามารถปรับให้แกนหมุนของ Servo Motor อยู่ตรงกลาง จึงทำให้การเลี้ยวซ้ายและเลี้ยวขวาได้รัศมีการเลี้ยวไม่เท่ากัน เมื่อค่าของพัลส์ห่างจากค่ากลางเท่า ๆ กัน

ปัญหาและแนวทางแก้ไข

1. ปัญหา: ตัวตรวจจับสิ่งกีดขวางด้วยคลื่นอัลตราโซนิก จะใช้ได้ดีกับวัตถุที่มีพื้นที่สัมผัสมากและเป็นผิวเรียบ นอกจากนี้ ความถูกต้องในการวัดยังขึ้นกับอุณหภูมิห้อง การถ่ายเทอากาศ และอาจถูกรบกวนจากแหล่งกำเนิดเสียงความถี่สูงด้วย จึงได้มีการนำไปใช้

แนวทางแก้ไข: ใช้แสงเป็นตัวช่วยในการตรวจจับวัตถุ (Optical Sensor) ในระยะใกล้ เพราะแสงสามารถตรวจจับวัตถุขนาดเล็กหรือใหญ่ก็ได้ อย่างแม่นยำ แต่จะใช้ได้ดีในระยะใกล้เท่านั้น

2. ปัญหา: ความเร็วที่ได้ในการป้อนสัญญาณพัลส์ค่า นั้น ๆ จะได้ความเร็วแต่ละครั้งไม่คงที่ เพราะระบบที่ใช้เป็นแบบลูปเปิดไม่มีการป้อนกลับความเร็ว ทำให้ความเร็วที่ได้ไม่คงที่ เนื่องจากการลดลงของกระแสจากแบตเตอรี่ที่จ่ายให้มอเตอร์

แนวทางแก้ไข: ทำระบบให้เป็นลูปปิด เพื่อให้ได้ความเร็วคงที่ และติดตั้งทาโคเอนโคดเดอร์ เพื่อทำการป้อนความเร็วกลับ

3. ปัญหา: แบตเตอรี่ที่ใช้เป็นแบบเปียก มีน้ำหนักมากทำให้มีปัญหาในตอนเริ่มวิ่ง ต้องใช้แรงบิดที่มีค่าสูงมาก

แนวทางแก้ไข: เปลี่ยนมาใช้แบตเตอรี่แบบแห้งแทน

4. ปัญหา: ความสามารถของไมโครคอนโทรลเลอร์มีจำกัด ทำให้การควบคุมการเคลื่อนที่ลำบาก

แนวทางแก้ไข: เปลี่ยนไปใช้หน่วยประมวลผลที่มีความสามารถสูงกว่านี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

โปรแกรมควบคุมการทำงาน

CH_1BUF	EQU	1BH	;1ST BYTE BUFFER
CH_2BUF	EQU	1CH	;2ND BYTE BUFFER
DISTBUF	EQU	08H	;DISTANCE BUFFER
N_SPDBUF	EQU	09H	;NEW SPEED LEVEL
O_SPDBUF	EQU	0AH	;OLD SPEED LEVEL
N_STRBUF	EQU	0BH	;NEW STEER LEVEL
O_STRBUF	EQU	0CH	;OLD SPEED LEVEL
PASS	EQU	06H	;ASCII OF ACK
FAULT	EQU	15H	;ASCII OF NAK
PORTH	EQU	0FC00H	;FRONT SENSOR
PORTL	EQU	0FC01H	;LEFT SENSOR
PORTR	EQU	0FC02H	;RIGHT SENSOR
TH00B	EQU	0DH	;TH0 BUFFER, OUT 0
TLO0B	EQU	0EH	;TLO BUFFER, OUT 0
TH01B	EQU	0FH	;TH0 BUFFER, OUT 1
TL01B	EQU	10H	;TLO BUFFER, OUT 1
TH20B	EQU	11H	;TH2 BUFFER, OUT 0
TL20B	EQU	12H	;TH2 BUFFER, OUT 0
TH21B	EQU	13H	;TH2 BUFFER, OUT 1
TL21B	EQU	14H	;TH2 BUFFER, OUT 1
COUNTD	EQU	15H	;COUNTER OF SPEED
COUNTR	EQU	16H	;COUNTER OF STEER
COMBD	EQU	17H	;SPEED COMPARE BUFFER
RMULD	EQU	18H	;SPEED RESULT OF MULTIPLICATION
COMBR	EQU	19H	;STEER COMPARE BUFFER
RMULR	EQU	1AH	;SPEED RESULT OF MULTIPLICATION
TOIH	EQU	68H	;TO INT. ROUTINE HIGH BYTE
TOIL	EQU	69H	;TO INT. ROUTINE LOW BYTE
T2IH	EQU	6CH	;T2 INT. ROUTINE HIGH BYTE
T2IL	EQU	6DH	;T2 INT. ROUTINE LOW BYTE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้หรือเผยแพร่โดยไม่ได้รับอนุญาต
 ไม่ว่าการแก้ไข ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องสงวนลิขสิทธิ์ไว้

```

SIH      EQU      6AH  ;SERIAL INT. ROUTINE HIGH BYTE
SIL      EQU      6BH  ;SERIAL INT. ROUTINE LOW BYTE

      ORG 8000H

MAIN:    MOV IE, #F2H      ;กำหนดค่าให้กับ IE
      MOV IP, #F0H      ;กำหนดค่าให้กับ IP
      MOV SCON, #50H     ;กำหนดค่าให้กับ SCON
      MOV PCON, #00H     ;กำหนดค่าให้กับ PCON
      MOV TOIH, #81H     ;ตำแหน่งของ timer0 inter-
      MOV TOIL, #00H     ;rupt service routine
      MOV T2IH, #82H     ;ตำแหน่งของ timer2 inter-
      MOV T2IL, #00H     ;rupt service routine
      MOV SIH, #85H      ;ตำแหน่งของ serial inter-
      MOV SIL, #00H      ;rupt service routine
      MOV TH1, #F4H      ;กำหนดค่า TH ของ timer1
      MOV TH00B, #BEH    ;กำหนดค่าเริ่มต้นของ TH0, TLO,
      MOV TLO0B, #DOH    ;TH2, TL2
      MOV TH01B, #FAH    ;
      MOV TLO1B, #ACH    ;
      MOV TH20B, #BEH    ;
      MOV TL20B, #70H    ;
      MOV TH21B, #FBH    ;
      MOV TL21B, #3AH    ;
      MOV COUNTS, #02H   ;กำหนดค่าให้กับ counter
      MOV COUNTR, #02H   ;
      MOV TMOD, #21H     ;กำหนดค่าให้กับ TMOD
      MOV TCON, #51H     ;กำหนดค่าให้กับ TCON
      MOV N_SPDBUF, #03H ;กำหนดค่าเริ่มต้นของ SPDBUF
      MOV O_SPDBUF, #03H ;ให้อยู่ที่ค่ากลาง
      MOV N_SPRBUF, #03H ;กำหนดค่าเริ่มต้นของ SPRBUF
      MOV O_SPRBUF, #03H ;ให้อยู่ที่ค่ากลาง

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น ห้ามเผยแพร่โดยไม่ได้รับอนุญาต
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้า MOV DPTR, #CONTROLPORT; กำหนดใน control port ของ ถ้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้า MOV A, #8BH; กำหนดค่า A และต้องอ้างอิง; 8255 องเอกสารทุกครั้งที่มีมีการนำไปใช้

```

MOVX @DPTR,A ;
LOOP: MOV T2CON,#05H ;กำหนดค่าให้กับ T2CON
      SJMP $ ;ให้วนทำงานอยู่ที่เดิม

ORG 8100H
TOINT: DJNZ COMPD ;ลดค่า counter ลง 1 ถ้ายังไม่
      ;เป็น 0 แสดงว่าให้ out 0 ออก
      SJMP OUT1D ;ถ้าเป็น 0 ให้ out 1 ออก
COMPD: MOV A,N_STDBUF ;
      CJNE A,O_STDBUF,CALNOD;เปรียบเทียบค่าเดิมกับค่าใหม่
      ;ถ้าไม่เท่าให้ไปคำนวณตำแหน่ง
      ;เริ่มต้นของตาราง
      MOV COMBD,#99H ;เป็นตัวเปรียบเทียบในการตัดสินใจ out 1 ออก
      MOV THO,THO0B ;โหลดค่าเดิมให้กับ THO
      MOV TLO,TLO0B ;โหลดค่าเดิมให้กับ TLO
      LJMP OUTOD ;กระโดดไป out0
CALNOD: MOV B,#04H ;
      MUL AB ;ค่า speed level คูณกับ 04H
      MOV RMULD,A ;
      MOV DPTR,#TABD ;
      MOVC A,@A+DPTR ;เท่ากับตำแหน่งเริ่มต้นที่จะโหลด
      ;ค่าให้ THO,TLO
      MOV THO0B,A ;
      MOV THO,A ;โหลดค่าใหม่ให้ THO
      INC RMULD ;เพิ่มตำแหน่งในตาราง
      MOV A,RMULD ;
      MOVC A,@A+DPTR ;
      MOV TLO0B,A ;
      MOV TLO,A ;โหลดค่าใหม่ให้ TLO
      CLR P1.1 ;out 0 ออกขา p1.1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในงานเพื่อการศึกษาเท่านั้น;ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

OUT1D:      MOV COUNTD,#02H      ;
            MOV A,#99H           ;
            CJNE A,COMBD,CALN1D  ;ตัดสินใจว่าจะใช้ค่าเก่าหรือใหม่
            ;โดยการเปรียบเทียบเทอมแรก
            MOV TH0,TH01B       ;โหลดค่าเดิมให้ TH0
            MOV TLO,TLO1B       ;โหลดค่าเดิมให้ TLO
            LJMP OUT11D         ;กระโดดไป out1
CALN1D:     MOV DPTR,#TABD      ;โหลดค่าใหม่ให้กับ TH0,TLO
            INC RMULD           ;
            MOV A,RMULD         ;
            MOVC A,@A+DPTR      ;
            MOV TH01B,A         ;
            MOV TH0,A           ;
            INC RMULD           ;
            MOV A,RMULD         ;
            MOVC A,@A+DPTR      ;
            MOV TLO1B,A        ;
            MOV TLO,A           ;
OUT11D:     SETB P1.1           ;out 1 ออกที่ขา P1.1
            MOV COMBD,#00H      ;clear COMBD
            RETI                ;
            ORG 8200H
T2INT:     CLR T2CON.7          ;clear timer2 overflow
            DJNZ COMPR          ;ลดค่า counter ลง 1 ถ้ายังไม่
            ;เป็น 0 ให้ out 0 ออก
            SJMP OUT1R         ;ถ้าเป็น 0 ให้ out 1 ออก
COMPR:     MOV A,N_SPRBUF      ;
            CJNE A,O_SPRBUF,CALNOR;เปรียบเทียบค่าเก่ากับใหม่ ถ้า
            ;ไม่เท่ากันให้หาตำแหน่งในตาราง
            MOV COMBR,#99H     ;
            MOV TH2,TH20B      ;โหลดค่าเดิมให้กับ TH2
            MOV TL2,TL20B      ;โหลดค่าเดิมให้กับ TL2

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการศึกษาเท่านั้น มิใช่ให้เผยแพร่โดยไม่ได้รับอนุญาต
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามให้ข้อมูลนี้แก่บุคคลอื่นและต้องอ้างอิงถึงแหล่งที่มาของเอกสารนี้

```

                                LJMP OUTOR                ;กระโดดไป out0
CALNOR:                        MOV B,#04H                ;
                                MUL AB                    ;ค่า steer level คูณกับ 04H
                                MOV RMULR,A              ;
                                MOV DPTR,#TABR            ;
                                MOVC A,@A+DPTR            ;ตำแหน่งเริ่มที่จะโหลดค่าให้ TH0,
                                ;TLO
                                MOV TH20B,A              ;
                                MOV TH2,A                 ;โหลดค่าใหม่ให้ TH2
                                INC RMULR                 ;เพิ่มตำแหน่งในตาราง
                                MOV A,RMULR               ;
                                MOVC A,@A+DPTR            ;
                                MOV TL20B,A               ;
                                MOV TL2,A                 ;โหลดค่าใหม่ให้ TL2
OUTOR:                          CLR P1.0                ;out 0 ออกขา P1.0
                                RETI                      ;
OUT1R:                          MOV COUNTR,#02H          ;
                                MOV A,#99H               ;
                                CJNE A,COMBR,CALN1R       ;เลือกว่าจะใช้ค่าเก่าหรือใหม่จาก
                                ;ผลของการเปรียบเทียบ
                                MOV TH2,TH21B            ;โหลดค่าเดิมให้ TH2
                                MOV TL2,TL21B            ;โหลดค่าเดิมให้ TL2
                                LJMP OUT11R              ;กระโดดไป out1
CALN1R:                          MOV DPTR,#TABR          ;โหลดค่าใหม่ให้กับ TH2, TL2
                                INC RMULR                ;
                                MOV A,RMULR              ;
                                MOVC A,@A+DPTR            ;
                                MOV TH21B,A              ;
                                MOV TH2,A                ;
                                INC RMULR                ;
                                MOV A,RMULR              ;
                                MOV A,@A+DPTR            ;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV TL21B,A      ;
MOV TL2,A        ;
OUT11R:          SETB P1.0      ;out 1 ออกที่ขา P1.0
                  MOV COMBR,#00H ;clear COMBR
                  RETI          ;

                  ORG 8500H

SRINT:           JB RI,RECEIVE   ;ตรวจสอบว่าเป็นการรับหรือส่ง
                  JB TI,TRANSMIT ;
                  RETI          ;

TRANSMIT:        MOV A,#'D'      ;ตรวจว่าค่าสิ่งนั้นถามระยะทาง
                  CJNE A,CH_1BUF,RETURN;ถ้าไม่ใช่ให้ไป return
                  MOV CH_1BUF,#00H ;ถ้าใช่ให้ clear buffer
                  CLR TI         ;clear TI
                  MOV SBUF,DISTBUF ;ส่งระยะทางออกไป
                  RETI          ;
RETURN:          CLR TI         ;clear TI
                  RETI          ;

RECEIVE:         JB FO,RECEIVE_2ND ;ตรวจว่าเป็น byte ที่เท่าไร
                  MOV CH_1BUF,SBUF ;เก็บ byte 1ไว้
                  SETB FO        ;set flag
                  CLR RI        ;clear RI
                  RETI          ;

RECEIVE_2ND:     MOV CH_2BUF,SBUF ;เก็บ byte ที่ 2 ไว้
                  MOV A,#'R'    ;ตรวจว่าเป็นค่าสิ่งความเร็ว
                  CJNE A,CH_1BUF,STEER;
                  MOV A,CH_2BUF ;ถ้าใช่แปลงระดับความเร็ว
                  ANL A,#0FH     ;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับศึกษาเท่านั้น; ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้าม JZ NAK ปลงเนื้อหาและต้องอ้างอิง;ถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ANL A, #0F8H ;
JNZ NAK ;
MOV O_SPDBUF, N_SPDBUF; เก็บข้อมูลเก่าไว้
MOV N_SPDBUF, CH_2BUF; เก็บข้อมูลใหม่แทนที่
SJMP ACK ;
STEER: MOV A, #'S' ; ตรวจสอบว่าเป็นคำสั่งเลข
CJNE A, CH_1BUF, DISTANCE;
MOV A, CH_2BUF ; ถ้าใช่แปลงระดับวงเลข
ANL A, #0FH ;
MOV CH_2BUF, A ;
JZ NAK ;
ANL A, #0F8H ;
JNZ NAK ;
MOV O_STRBUF, N_STRBUF; เก็บข้อมูลเก่าไว้
MOV N_STRBUF, CH_2BUF; เก็บข้อมูลใหม่แทนที่
SJMP ACK ;
DISTANCE: MOV A, #'D' ; ตรวจสอบเป็นการถามระยะทาง
CJNE A, CH_1BUF, NAK ;
AHEAD: MOV A, #'H' ; ตรวจสอบเป็นด้านหน้า
CJNE A, CH_2BUF, LEFT ;
MOV DPTR, #PORTH ; เก็บระยะทางด้านหน้าไว้
MOVX A, @DPTR ;
SJMP CAL ;
LEFT: -MOV A, #'L' ; ตรวจสอบเป็นด้านซ้าย
CJNE A, CH_2BUF, RIGHT;
MOV DPTR, #PORTL ; เก็บระยะทางด้านซ้ายไว้
MOVX A, @DPTR ;
SJMP CAL ;
RIGHT: MOV A, #'R' ; ตรวจสอบเป็นด้านขวา
CJNE A, CH_2BUF, NAK ;
MOV DPTR, #PORTR ; เก็บระยะทางด้านขวาไว้
MOVX A, @DPTR ;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับศึกษาเท่านั้น; ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่า CAL: ใดๆ ทั้งสิ้น อีกทั้งห้ามมิได้เผยแพร่หาและต้องอ้างอิง; คำว่าขนาดระยะทางที่อ่านมาใช้

```

MUL AB ;
MOV B, #0AH ;
DIV AB ; ทำให้เป็นเซนติเมตร
MOV DISTBUF, A ;
SJMP ACK ;
NAK: MOV CH_1BUF, #00 ; เมื่อคำสั่งไม่ถูกต้อง
CLR RI ;
CLR FO ;
MOV SBUF, #FAULT ;
RETI ;
ACK: CLR RI ; เมื่อคำสั่งถูกต้อง
CLR FO ;
MOV SBUF, #PASS ;
RETI ;
ORG 9000H
TABSPEED: DB BDH, FOH, FBH, FOH
DB BEH, 00H, FBH, COH
DB BEH, 10H, FBH, 95H
DB BEH, DOH, FAH, ACH
DB BFH, COH, F9H, EOH
DB COH, 00H, F9H, COH
DB COH, 30H, F9H, 90H

ORG 9200H
TABSTEER: DB BAH, EOH, FEH, BOH
DB BCH, 50H, FDH, 6AH
DB BDH, BOH, FCH, 20H
DB BEH, 70H, FBH, 3AH
DB BFH, 30H, FAH, 50H
DB COH, AOH, F9H, 10H
DB C1H, AOH, F7H, DOH

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

END

ภาคผนวก ข.

แนวทางพัฒนา

1. ทำตัวรถขึ้นมาเอง เพื่อที่จะได้ไม่มีข้อจำกัดด้านแมคคานิกส์ และสามารถทำให้การควบคุมการเคลื่อนที่เป็นแบบ ลูปปิด เพื่อให้ง่ายแก่การควบคุม และสามารถออกแบบให้ใช้เทคนิคการเลี้ยวแบบ Differential Speed ซึ่งจะช่วยให้วงเลี้ยวแคบขึ้น และมีเสถียรดีขึ้น ซึ่งเทคนิคการเลี้ยวแบบนี้ใช้หลักการป้องกันความเร็วให้กับล้อทั้งสองข้างไม่เท่ากัน

2. นำระบบตรวจจับต่าง ๆ มาติดตั้งบนตัวรถ เพื่อที่จะเพิ่มความสามารถของรถสำรวจ เช่น ตัวตรวจจับกลิ่น ตัวตรวจจับความร้อน ตัวตรวจจับเสียง เป็นต้น

3. ใช้ ซีพียู เบนซ์ที่มีความสามารถสูงติดตั้งบนตัวรถ เพื่อที่ตัวรถจะสามารถเคลื่อนที่ไปได้เองโดยไม่ต้องมีคนคอยควบคุมอยู่หน้าจอ และการเคลื่อนที่ของตัวรถสามารถทำได้หากคอมพิวเตอร์หรือระบบการสื่อสารด้วยคลื่นวิทยุเสีย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

1. บัญชา ตั้งตระกูล, "กลศาสตร์ยานยนต์", บริษัท ซีเอ็ดยูเคชั่น จำกัด, 2532
2. ยืน ภู่วรรณ, "ทฤษฎีและการประยุกต์ไมโครโปรเซสเซอร์ Z-80", บริษัท ซีเอ็ดยูเคชั่น จำกัด, 2533
3. กฤษดา วิศวธีรานนท์, "ไมโครโปรเซสเซอร์", สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น), 2528
4. ผศ.ดร.โยธิน เปรมปราณีรัตน์, "วิเคราะห์และออกแบบระบบการควบคุมมอเตอร์", สถาบันเทคโนโลยี พระจอมเกล้า เจ้าคุณทหาร ลาดกระบัง, 2526
5. ผศ.ดร.โยธิน เปรมปราณีรัตน์, "ระบบเซอร์โวและอิเล็กทรอนิกส์คอนโทรลมอเตอร์", สถาบันเทคโนโลยี พระจอมเกล้า เจ้าคุณทหาร ลาดกระบัง, 2533
6. อนุตรชัย ธี กลาง, "การพัฒนาระบบควบคุมของยานขนส่งแบบนำร่องอัตโนมัติ", วิทยานิพนธ์ สาขาวิศวกรรมไฟฟ้า สถาบันเทคโนโลยี พระจอมเกล้า เจ้าคุณทหาร ลาดกระบัง, 2535
7. "การพัฒนาหุ่นยนต์เคลื่อนที่ขั้นที่ 2", วิทยานิพนธ์ ภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม สถาบันเทคโนโลยี พระจอมเกล้า เจ้าคุณทหาร ลาดกระบัง, 2534
8. Kenneth J. Ayala, "The 8051 Microcontroller Architecture Programming and Application", West Publishing Company, 1991
9. "MCS-51 Microcontroller", ETT CO., LTD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้