สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

รถสำรวจไร้สายควบคุมด้วยคอมพิวเตอร์

นางสาวสุขศิริ กาญจนสำราญวงศ์
นายสุนทร แซ่ตั้ง
นายสุเมธา นามศิริเลิศ

เลขหมู่..............
เลขทะเบียน 62316
วัน,เดือน,ปี - 8 ส.ค. 2549

b 11620213
i ....

โครงงานพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
ภาควิชาฟิสิกส์ประยุกต์
คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา พ.ศ. 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Computer-Controlled Wireless Rover

Miss. Suksiri Kanchanasamranvong
Mr. Soontorn Saetang
Mr. Sumetha Namsirilert

A Special Project Submitted in Partial Fulfillment of the Requirement for the Degree of Bachelor of Science

Department of Applied Physics
Faculty of Science
King Mongkut's Institute of Technology Ladkrabang

Academic Year 2005

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงงานพิเศษเรื่อง	รถสำรวจไร้สายควบคุมด้วยคอมพิวเตอร์
นักศึกษา	นางสาวสุขศิริ กาญจนสำราญวงศ์
	นายสุนทร แซ่ตั้ง
	นายสุเมธา นามศิริเลิศ
ภาควิชา	ฟิสิกส์ประยุกต์
สาขาวิชา	ฟิสิกส์ประยุกต์-เครื่องมือวิทยาศาสตร์อุตสาหกรรม
อาจารย์ที่ปรึกษา	รศ.วิชิต ศิริโชติ

ภาควิชาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้โครงงานพิเศษฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

คณะกรรมการตรวจสอบ		ลายมือชื่อ
ประธานกรรมการ	อ.ภูมินทร์ จินดาจิธาวัฒน์	ภูมินทร์ จินดาจิธาวัฒน์
กรรมการ	อ.เบญจพล ตันฮู้	
กรรมการ	ดร.ปิติพร ถนอมงาม	
กรรมการที่ปรึกษา	รศ.วิชิต ศิริโชติ	

....

(รองศาสตราจารย์วิชาญ เตชิตธีระ)

หัวหน้าภาควิชาฟิสิกส์ประยุกต์

ลิขสิทธิ์ของภาควิชาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงงานพิเศษ เรื่อง **รถสำรวจไร้สายควบคุมด้วยคอมพิวเตอร์**

นักศึกษา นางสาวสุขศิริ กาญจนสำราญวงศ์
นายสุนทร แซ่ตั้ง
นายสุเมธา นามศิริเลิศ

ภาควิชา ฟิสิกส์ประยุกต์
สาขาวิชา ฟิสิกส์ประยุกต์-ฟิสิกส์เครื่องมือวิทยาศาสตร์และอุสาหกรรม
ปีการศึกษา 2548
อาจารย์ที่ปรึกษา รศ. วิชิต ศิริโชติ

บทคัดย่อ

โครงงานพิเศษนี้ได้ออกแบบและสร้างหุ่นยนต์สำรวจควบคุมด้วยคลื่นวิทยุความถี่ 2.4 GHz หุ่นยนต์สามารถปฏิบัติงานที่ระยะห่างไกลและเป็นอันตรายได้ บนหุ่นยนต์ได้ติดตั้งกล้องวงจรปิดไร้สายเพื่อส่งสัญญาณภาพไปยังส่วนควบคุม แผงวงจรคอมพิวเตอร์สร้างด้วยไมโครคอนโทลเลอร์ MCS – 51 ได้พัฒนาโปรแกรมติดต่อผู้ใช้แบบกราฟิกด้วยโปรแกรม Visual Basic นอกจากที่ติดตั้งกล้องวีดิโอไร้สายแล้ว หุ่นยนต์ยังมีเซ็นเซอร์วัดอุณหภูมิและอินฟราเรดเซ็นเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Special Project Title Computer-Controlled Wireless Rover

Name Ms. Suksiri Kanchanasamranvong
Mr. Soontorn Saetang
Mr. Sumetha Namsirilert

Department Applied Physics
Program Applied Physics Instrumentation and factory
Academic year 2548
Special Project Advisor Assoc.Prof. Wichit Sirichote

ABSTRACT

A robot controlled by a 2.4 GHz radio wave has been designed and built. The robot can be controlled for remote and hazardous areas. The close circuit video camera was installed. The video signal was modulated and sent to the receiver. The onboard computer was built using MCS51 microcontroller. The GUI remote control software was developed with Visual Basic program. In addition to the built – in video camera , the robot also provides temperature and infrared sensors

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบคุณอาจารย์ที่ปรึกษา รศ.วิชิต ศิริโชติ เป็นอย่างสูงที่ได้ให้คำแนะนำและคำปรึกษาเป็นอย่างสูงที่ทำให้โครงงานพิเศษนี้สำเร็จลุล่วงไปได้ดี

ขอขอบพระคุณคุณพ่อคุณแม่และอาจารย์ทุกท่านที่ให้ความสนับสนุนและเป็นกำลังใจเสมอมาขอบคุณ ครอบครัว นามศิริเลิศที่ ให้การอุปการะและเอื้อเฟื้อสถานที่ในการทำโครงงานพิเศษตลอดมา จนกระทั่งโครงงานนี้เสร็จสมบูรณ์

นางสาวสุขศิริ กาญจนสำราญวงศ์
นายสุนทร แซ่ตั้ง
นายสุเมธา นามศิริเลิศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญตาราง	ช
สารบัญรูป	ซ
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการ	3
2.1 รหัสแอสกี้ ASCII Code	3
2.2 ลักษณะเฉพาะของไมโครคอนโทรลเลอร์ MCS-51	4
2.3 การทำงานของไมโครคอนโทรลเลอร์	5
2.3.1 หน้าที่ขาต่างๆของ MCS-51	5
2.3.2 การจัดหน่วยความจำของ MCS-51	6
2.3.3 Timer/Counter	11
2.3.4 วงจรนับ/จับเวลา 2 (Timer2)	13
2.3.5 การ interupt ใน 8051	13
2.4 การรับส่งข้อมูลแบบอนุกรม	14
2.5 RF2.4GHz	19
2.6 วีดีโอเซ็นเดอร์ (VIDEO SENDER)	22
2.7 ส่วนตัวรถ	23
2.7.1 การขับมอเตอร์ไฟตรง	23
2.7.2 การขับมอเตอร์ไฟตรงแบบ H-Bridge	24
2.7.3 การควบคุม Stepping Motor แบบยูนิโพลาร์	25
2.8 อินฟราเรด(Infrared)	29
2.9 Ds 1820 1-Wire Thermometer	31
2.9.1 หลักการเบื้องต้นของไอซี DS1820	31
2.9.2 การอินเตอร์เฟสผ่านสายเส้นเดียว	31
2.9.3 คุณสมบัติ	31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
บทที่ 3 วิธีดำเนินการวิจัย	36
3.1 ส่วนการออกแบบตัวรถและระบบควบคุม	36
3.1.1 ส่วนการรับ-ส่งสัญญาณควบคุมจากคอมพิวเตอร์	39
3.1.2 ส่วนควบคุมการหมุนของกล้อง	39
3.1.3 ส่วนการขับมอเตอร์ไฟตรง	40
3.1.4 ส่วนไมโครคอนโทรลเลอร์ 8051	40
3.1.5 ส่วนกล้องและส่วนรับส่งสัญญาณภาพไร้สาย	41
3.2 ตัวตรวจรู้	41
3.2.1 Infrared	41
3.2.3 DS1820 1-Wire Thermometer	42
3.3 ส่วนซอฟต์แวร์ควบคุมการทำงานของรถ	42
3.3.1 ส่วนโปรแกรมควบคุมบน 8051 จะใช้โปรแกรม Micro C และภาษา Assembly	42
3.3.2 ส่วนโปรแกรมติดต่อกับผู้ใช้จะใช้ Visual Basic ในการสร้างโปรแกรม	46
บทที่ 4 การทดลองและผลการทดลอง	55
4.1 ผลการทดลองในส่วนรับส่งข้อมูลระหว่างคอมพิวเตอร์กับตัวรถสำรวจ	55
4.2 ผลการทดลองในส่วนรับส่งข้อมูลระหว่างรีโมทคอนโทลกับตัวรถ	56
4.3 ผลการทดลองในส่วนของการขับเคลื่อน	57
4.3.1 ผลการทดลองขับสเต็ปปิ้งมอเตอร์	57
4.3.2 ผลการทดลองขับมอเตอร์กระแสตรง	58
4.4 ผลการทดลองในส่วนของระบบตรวจรู้	58
4.4.1 ผลการทดลองการวัดระยะทางการเคลื่อนที่ของรถสำรวจ	58
4.4.2 ผลการทดลองการวัดมุมของการตั้งศูนย์กล้อง	59
บทที่ 5 สรุป ปัญหาและข้อเสนอแนะ	61
5.1 สรุปผลการทดลอง	61
5.2ปัญหาที่พบ	61

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บรรณานุกรม

ภาคผนวก ก.

ภาคผนวก ข.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 2.1 ตารางรหัสแอสกี้(เลขฐาน 16) 3

ตารางที่ 2.2 ตารางรหัสแอสกี้ที่เป็นรหัสควบคุม 3

ตารางที่ 2.3 ตารางแสดงลักษณะเฉพาะของไมโครคอนโทรเลอร์ตระกูล MCS-51 ของบริษัท Intel 4

ตารางที่ 2.4 แสดงลำดับการทำงานของขดลวดในแต่ละเฟสของมอเตอร์ เมื่อได้รับการกระตุ้นแบบหนึ่งเฟส 24

ตารางที่ 2.5 แสดงลำดับการทำงานของขดลวดในแต่ละเฟสของมอเตอร์ เมื่อได้รับการกระตุ้นแบบสองเฟส 25

ตารางที่ 2.6 แสดงลำดับการทำงานของขดลวดในแต่ละเฟสของมอเตอร์ เมื่อได้รับการกระตุ้นแบบครึ่งสเต็ป

ตารางที่ 3.1 แสดงการตอบสนองการทำงานเมื่อคลิก Mouse และกดคีย์บอร์ด 47

ตารางที่ 4.1 การวัดระยะทางการเคลื่อนที่ 59

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่ 2.1 ไมโครคอนโทรลเลอร์ 40 ขา แบบ PLCC และ PDIP พร้อมชื่อของขาต่างๆ 5

รูปที่ 2.2 การจัดหน่วยความจำของ MCS-51 6

รูปที่ 2.3 หน่วยความจำโปรแกรม 7

รูปที่ 2.4 แสดงหน่วยความจำข้อมูลที่เข้าถึงได้โดยตรงและโดยอ้อม 8

รูปที่ 2.5 แสดงส่วนของเรจิสเตอร์ R0-R7 ในแต่ละ Bank รวมทั้งส่วนที่เข้าถึงได้แบบบิท (เข้าถึงได้ทั้งแบบโดยตรงและโดยอ้อม) 8

รูปที่ 2.6 แสดงส่วนของเรจิสเตอร์หน้าที่พิเศษหรือ Special Function Registers (เข้าถึงแบบ โดยตรงเท่านั้น) 9

รูปที่ 2.7 เป็นการรับส่งข้อมูลแบบขึ้นกับเวลา Synchronous Serial Communication 16

รูปที่ 2.8 เป็นการรับส่งข้อมูลแบบไม่ขึ้นกับเวลา Asynchronous Serial Communication 16

รูปที่ 2.9 แสดงขาต่างๆ ของ MAX232 17

รูปที่ 2.10 แสดงตำแหน่งขา TXD และ RXD ของ 8051 17

รูปที่ 2.11 แสดงการเชื่อมต่อ MAX232 กับ 8051 18

รูปที่ 2.12 รูปแสดงการรับส่งข้อมูลแบบจุดต่อจุด 22

รูปที่ 2.13 แสดงการกระตุ้นด้วยเทคนิค PWM ที่ความเร็วรอบต่างๆ 23

รูปที่ 2.14 แสดงวงจรขับมอเตอร์ไฟตรง H-Bridge

รูปที่ 2.15 แสดงวงจรสมมูลย์ H-Bridge 24

รูปที่ 2.16 Full Bridge Driver Circuit 24

รูปที่ 2.17 แสดงการวางของขดลวดแต่ละเฟสของ stepping motor 26

รูปที่ 2.18 แสดงการหมุนของแกนมอเตอร์เมื่อได้รับการกระตุ้นแบบหนึ่งเฟส 26

รูปที่ 2.19 แสดงการหมุนของแกนมอเตอร์เมื่อได้รับการกระตุ้นแบบสองเฟส 27

รูปที่ 2.20 แสดงการหมุนของแกนมอเตอร์เมื่อได้รับการกระตุ้นแบบครึ่งสเต็ป 29

รูปที่ 2.21 วงจรอินฟราเรดเซ็นเซอร์ 30

รูปที่ 2.22 แผนผังของระบบบัสแบบ 1- Wire Bus 32

รูปที่ 2.23 จังหวะเวลาในการทำกระบวนการตรวจสอบว่ามีอุปกรณ์อยู่บนบัส 33

รูปที่ 2.24 จังหวะเวลาที่ไมโครคอนโทรลเลอร์ใช้เขียนข้อมูล '0' หรือ '1' ไปยังอุปกรณ์บนบัส 34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

หน้า

รูปที่ 2.25 จังหวะเวลาที่ไมโครคอนโทรลเลอร์ใช้อ่านบิตข้อมูล ‘0’ หรือ ’1’ จากอุปกรณ์บนบัส 35
รูปที่ 3.1 บล็อกไดอะแกรมของรถสำรวจ 36
รูปที่ 3.2 ภาพแสดงตัวอย่างโครงสร้างของตัวรถ 37
รูปที่ 3.3 ภาพแสดงตัวอย่างการเคลื่อนที่ของรถสำรวจ 38
รูปที่ 3.4 แสดงวงจรขับ Stepping Motor โดยใช้ไอซี Driver เบอร์ ULN2003 39
รูปที่ 3.5 วงจรขับมอเตอร์ไฟตรง 40
รูปที่ 3.6 วงจร อินฟราเรดเซ็นเซอร์ 42
รูปที่ 3.7 รูปแสดงแผนผังการส่วนควบคุมด้วยโปรแกรม Micro C 43
รูปที่ 3.8 โปรแกรมที่ใช้ในการควบคุม 46
รูปที่ 3.9 แสดงการ Add Component MSComm 48
รูปที่ 3.10 แสดงการนำ Control MSComm มาใช้ใน Form ของโปรแกรม 48
รูปที่ 3.11 การลงทะเบียนFile*.ocx 50
รูปที่ 3.12 แสดง การ Add Control Component ezVidCap 51
รูปที่ 3.13 แสดงการนำ Control ezVidCap มาวางบน Form ของโปรแกรม 51
รูปที่ 3.14 แสดงแผนผังการทำงานของโปรแกรม VB ในส่วนการควบคุมหลัก 52
รูปที่ 3.15 แสดงแผนผังการทำ งานของโปรแกรม VB เมื่อในตัวโปรแกรมเป็นเกิด Timer Interrupt 53
รูปที่ 3.16 แสดงแผนผังการทำงานของโปรแกรม VB เมื่อมีการกดปุ่ม U บนคีย์บอร์ด 54
รูปที่ 4.1 แสดงหน้าตาโปรแกรม Visual Basic ที่ติดต่อระหว่างผู้ควบคุมรถสำรวจกับรถสำรวจ 55
รูปที่ 4.2 แสดงหน้าตารีโมทคอนโทล 56
รูปที่ 4.3 รูปวงจรขับสเต็ปเปอร์มอเตอร์ 57
รูปที่ 4.4 รูปวงจร DC Motor 58
รูปที่ 4.5 แสดงตำแหน่งของการตั้งศูนย์กล้อง 60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

ในอดีตการสำรวจหรือตรวจตราในบริเวณต่างๆมนุษย์อาจจะต้องเป็นคนลงมือเดินไปสำรวจเองจนกระทั่งได้มีการพัฒนาระบบกล้องวีดีโอวงจรปิดเพื่อที่จะสามารถส่งภาพบริเวณๆหนึ่งได้ ในอีกด้านก็มีผู้ที่ได้หุ่นยนต์ขึ้นเหมือนกันซึ่งในช่วงแรกการควบคุมหุ่นยนต์นั้นจะต้องมีสายเชื่อมต่อระหว่างหุ่นยนต์และตัวควบคุมอยู่ รวมทั้งการส่งภาพของกล้องวีดีโอวงจรปิดก็จะต้องใช้สายเชื่อมต่อระหว่างตัวกล้องและส่วนรับภาพอยู่เหมือนกัน จนต่อมาทั้งเทคโนโลยีการควบคุมหุ่นยนต์และการส่งสัญญาณภาพวีดีโอได้มีการพัฒนาให้เป็นระบบแบบไร้สายทั้งคู่จึงเป็นแนวคิดที่จะนำทั้งสองอย่างมาไว้ด้วยกัน โดยใช้ประโยชน์จากการเคลื่อนที่ของหุ่นยนต์และการส่งภาพของกล้องวีดีโอก็กลายมาเป็นกล้องวีดีโอที่เคลื่อนที่ได้เพื่อการยืดหยุ่นในการสำรวจและอำนวยความสะดวกแก่มนุษย์โดยที่มนุษย์สามารถมองภาพจากหน้าจอคอมพิวเตอร์และควบคุมการเดินของหุ่นยนต์ด้วยคอมพิวเตอร์เช่นกัน

1.1 วัตถุประสงค์

- ศึกษาระบบไมโครคอนโทรเลอร์ที่นำมาใช้งานเพื่อใช้ในการสั่งงานและควบคุมการทำงานของรถ โดยไมโครคอนโทรลเลอร์ที่เลือกใช้คือ AT89C51
- ศึกษาระบบสื่อสารแบบไร้สายด้วยการเชื่อมต่อระหว่างคอมพิวเตอร์และระบบไมโครคอนโทรลเลอร์ ทั้งทางด้านฮาร์ดแวร์ ซอฟต์แวร์ และระบบการรับส่งภาพของกล้อง
- เขียนโปรแกรมรับภาพจากกล้องวีดีโอที่ติดกับรถเพื่อแสดงภาพบนหน้าจอคอมพิวเตอร์
- เพื่อใช้เป็นแนวทางการพัฒนารถสำรวจควบคุมแบบไร้สายต่อไปในอนาคต

1.2 ขอบเขตของโครงงาน

เนื่องจากในระบบรักษาความปลอดภัยในตัวอาคารหรือบริเวณต่างๆอาจจะมีกล้องวีดีโอวงจรปิดติดไว้ตามจุดต่างๆซึ่งกล้องเหล่านั้นต้องอยู่ประจำจุดใดจุดหนึ่งซึ่งจากการทำเช่นนั้นอาจจะเกิดมุมอับของกล้องที่มองไม่เห็นเพราะตัวกล้องเคลื่อนที่ไม่ได้ดังนั้นจะเป็นการดีที่จะมีกล้องวีดีโอที่สามารถเคลื่อนที่ไปตามซอกต่างๆ เพื่อสำรวจจุดต่างได้อย่างครบถ้วนหรือถ้ามีวัตถุเล็กๆอยู่บนพื้นก็สามารถที่จะเข้าไปดูใกล้ๆให้เห็นว่าวัตถุนั้นเป็นอะไร โดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทั้งหมดนี้จะต้องอยู่บนพื้นฐานของการควบคุมแบบไร้สาย เพื่อที่จะอำนวยความสะดวกต่อมนุษย์ คือมนุษย์ควบคุมอยู่ที่หน้าจอคอมพิวเตอร์เพียงอย่างเดียว

1.3 ขั้นตอนการดำเนินโครงงาน

ประกอบด้วยส่วนสำคัญ 2 ส่วนใหญ่ๆคือ

ส่วนที่เป็นซอฟต์แวร์ เป็นส่วนที่ทำหน้าที่ในการติดต่อกับผู้ใช้เพื่อควบคุมตัวรถให้วิ่งไปในทิศทางต่างๆตามต้องการและประมวลผลรับภาพจากกล้องวีดีโอที่อยู่ติดกับตัวรถเพื่อแสดงภาพบนหน้าจอคอมพิวเตอร์

ส่วนที่เป็นฮาร์ดแวร์ เป็นส่วนที่ใช้ติดต่อระหว่างคอมพิวเตอร์และหุ่นยนต์ไร้สาย โดยจะประกอบด้วยส่วนต่างๆ 6 ส่วนใหญ่ๆ

1. วงจรควบคุม 1 ทำหน้าที่ในการเตรียมข้อมูลที่จะรับส่งระหว่าง คอมพิวเตอร์ (ซอร์ฟแวร์) กับเครื่องรับ/ส่งสัญญาณ (RF)

2. วงจรควบคุม2 ทำหน้าที่ในการเตรียมข้อมูลที่จะใช้รับส่งระหว่างเครื่องรับ/ส่งสัญญาณอาร์เอฟ (RF) กับตัวรถ

3. เครื่องรับ/ส่งสัญญาณ RF แปลงข้อมูลเป็นสัญญาณดิจิตอล ให้เป็นสัญญาณอนาลอก และสัญญาณอนาลอก ไปเป็นสัญญาณดิจิตอล และเครื่องรับ/ส่งข้อมูลด้วยระบบสัญญาณไร้สายอาร์เอฟ

4. กล้องวีดีโอแบบไร้สาย เป็นกล้องที่จะนำมาติดกับตัวรถที่จะส่งสัญญาณภาพที่บริเวณนั้นๆไปที่ตัวรับสัญญาณ

5. ตัวรถ เป็นตัวรถที่สามารถเคลื่อนที่สำรวจเส้นทางและจุดต่างๆที่สามารถควบคุมแบบไร้สายจากคอมพิวเตอร์ที่อยู่ไกลออกไป

6. อินฟาเรด เป็นตัววัดระยะทางการเคลื่อนที่ของรถว่าเดินทางไปได้ระยะเท่าไร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

2.1 รหัสแอสกี้ ASCII Code

ในการแทนค่าในระบบคอมพิวเตอร์นอกจากเป็นจำนวนตัวเลข แล้วจำนวนที่เป็นตัวอักษร เราจะแทนด้วยการเก็บข้อมูลขนาด 1 Byte หรือ 8 Bit โดยมีรหัสมาตรฐานกลางคือ ASCII (American Standard Code for Information Interchange)

ตารางที่ 2.1 ตารางรหัสแอสกี้(เลขฐาน 16)

Hex	Char.	Hex	Char.	Hex	Char.	Hex	Char.	Hex	Char.
00	NUL	1A	SUB	34	4	4E	N	68	h
01	SOH	1B	ESV	35	5	4F	O	69	i
02	STX	1C	FS	36	6	50	P	6A	j
03	ETX	1D	GS	37	7	51	Q	6B	k
04	EOT	1E	RS	38	8	52	R	6C	l
05	ENQ	1F	US	39	9	53	S	6D	m
06	ACK	20	SP	3A	:	54	T	6E	n
07	BEL	21	!	3B	;	55	U	6F	o
08	BS	22	"	3C	<	56	V	70	p
09	HT	23	#	3D	=	57	W	71	q
0A	LF	24	$	3E	>	58	X	72	r
0B	VT	25	%	3F	?	59	Y	73	s
0C	FF	26	&	40	@	5A	Z	74	t
0D	CR	27	'	41	A	5B	[	75	u
0E	SO	28	(	42	B	5C	\	76	v
0F	SI	29	)	43	C	5D	]	77	w
10	DLE	2A	*	44	D	5E	^	78	x
11	DC1	2B	+	45	E	5F	_	79	y
12	DC2	2C	,	46	F	60		7A	z
13	DC3	2D	-	47	G	61	a	7B	{
14	DC4	2E	.	48	H	62	b	7C	|
15	NAK	2F	/	49	I	63	c	7D	}
16	SYN	30	0	4A	J	64	d	7E	~
17	ETB	31	1	4B	K	65	e	7F	DEL
18	CAN	32	2	4C	L	66	f		
19	EM	33	3	4D	M	67	g		

ตารางที่ 2.2 ตารางรหัสแอสกี้ที่เป็นรหัสควบคุม

Code		Code		Code	
NUL	All zero	FF	Form feed	CAN	Cancel
SOH	Start of heading	CR	Carriage return	EM	End of medium
STX	Start of text	SO	Shift out	SUB	Substitute
ETX	End of text	SI	Shift in	ESC	Escape
EOT	End of transmission	DLE	Data link escape	FS	File separator
ENQ	Enquiry	DC1	Device control	GS	Group separator
ACK	Acknowledge	DC2	Device control	RS	Record separator
BEL	Audible signal	DC3	Device control	US	Unit separator
BS	Backspace	DC4	Device control	SP	Space
HT	Horizontal tabulation	NAK	Negative acknowledge	DEL	Delete
LF	Line feed	SYN	Synchronous idle		
VT	Vertical tabulation	ETB	End of transmission block		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ลักษณะเฉพาะของไมโครคอนโทรลเลอร์ MCS-51

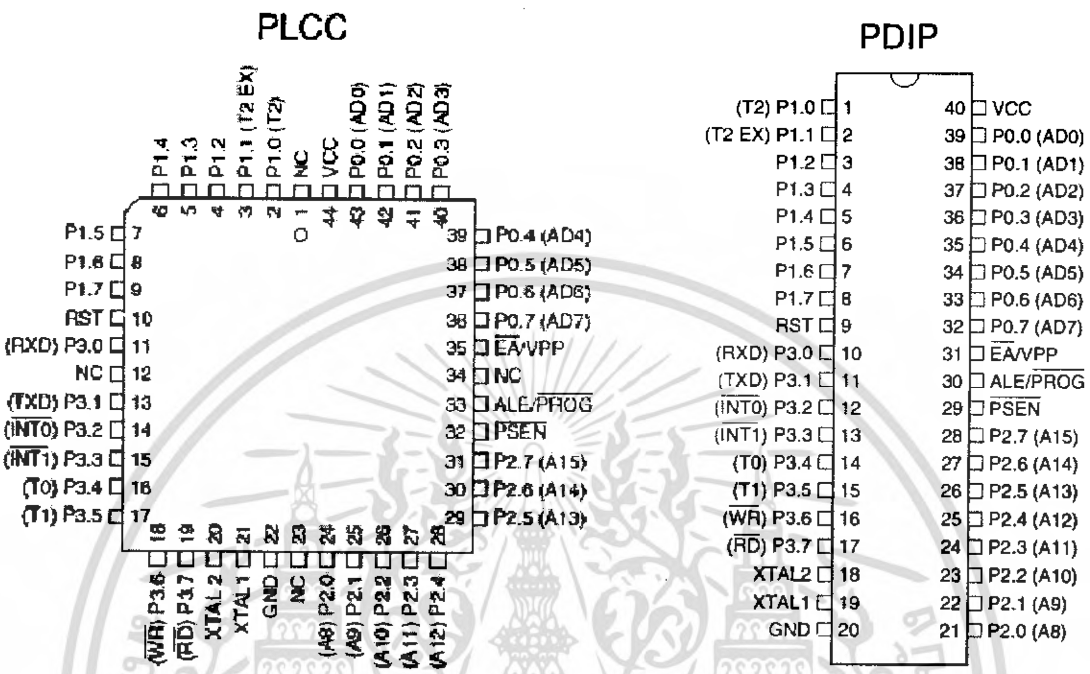
ไมโครคอนโทรเลอร์ 8051 ได้มีการผลิตครั้งแรกโดยบริษัท Intel และใช้ชื่อว่า MCS-51 ซึ่งปรากฏว่านิยมใช้กันอย่างแพร่หลาย

ตารางที่ 2.3 ตารางแสดงลักษณะเฉพาะของไมโครคอนโทรเลอร์ตระกูล MCS-51 ของบริษัท Intel

EMBEDDED CONTROLLERS										
Feature	8051AH	8031AH	8751H	80C51BH	80C31BH	87C51	8052AH	8032AH	8752	8044H
Program Memory (Bytes)	4K	-	4K	4K	-	4K	8K	-	8K	4K
RAM Memory (Bytes)	128	128	128	128	128	128	256	256	256	192
Program Memory Expansion (Off Chip) (Bytes)	64k									
Data Memory Expansion (Off Chip) (Bytes)	64K									
Max Clock Frequency (MHz)	12	12	12	16	16	16	16	12	12	12
Typical instruction Time (uS)	1	1	1	0.75	0.75	0.75	1	1	1	1
16-Bit Timer/Counter	2	2	2	2	2	2	3	3	3	2
Serial Communications	Synchronous Mode, Asynchronous Mode, 8 or 10 -Bit Programmable									HDLC/SDLC
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
No. of I/O Lines Interrupt Sources	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
(Two Priority Levels) Power Requirements 125	[illegible]	250	[illegible]	24	24	175	175	175	200	[illegible]
(ICC Max. mA) Programmable Power Modes Idle Power Down				4.0 mA 50 uA	4.0 mA 51 uA	4.0 mA 52 uA				30 mA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 การทำงานของไมโครคอนโทรลเลอร์



รูปที่ 2.1 ไมโครคอนโทรลเลอร์ 40 ขา แบบ PLCC และ PDIP พร้อมชื่อของขาต่างๆ

2.3.1 หน้าที่ขาต่างๆของ MCS-51

Vcc ไฟเลี้ยงสำหรับไอซีปกติแล้วจะอยู่ในช่วง 4.5 – 5.5 V แต่ก็ขึ้นอยู่กับไอซีเบอร์นั้นๆ ด้วย

GND ขากราวด์ของระบบ

EA External Access เมื่อป้อนไฟบวกเข้าที่ขานี้หมายความว่า CPU จะใช้ส่วนของ Program

Memory จากภายใน ถ้าเป็น GND หมายความว่าจะใช้ Program Memory จากภายนอก

Reset เป็นขา Reset สำหรับสั่งให้ CPU เริ่มทำงานใหม่เหมือนตอนที่เปิดเครื่อง(Power On Reset)

X1,X2 เป็นขาสำหรับป้อนสัญญาณนาฬิกาให้กับระบบ

ALE Address Latch Enable ขาสัญญาณนี้จะส่งสัญญาณนาฬิกาความถี่ 1/6 ของสัญญาณจาก Oscillator ขานี้จะใช้ในการ Latch สัญญาณ Address ในขณะที่ติดต่อหน่วยความจำภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PSEN Program Store Enable ปกติขานี้จะมีสถานะเป็น 1 แต่เมื่อ CPU ต้องการอ่านคำสั่ง (Fetch Instruction) จะส่งสัญญาณ 0 โดยขานี้สามารถนำไปต่อขาอินพุตของ TTL แบบ LS ได้ 8 อินพุต

WR Write ขาควบคุมการเขียนข้อมูลไปยังหน่วยความจำภายนอก

RD Read ขาควบคุมการอ่านข้อมูลสำหรับหน่วยความจำภายนอก

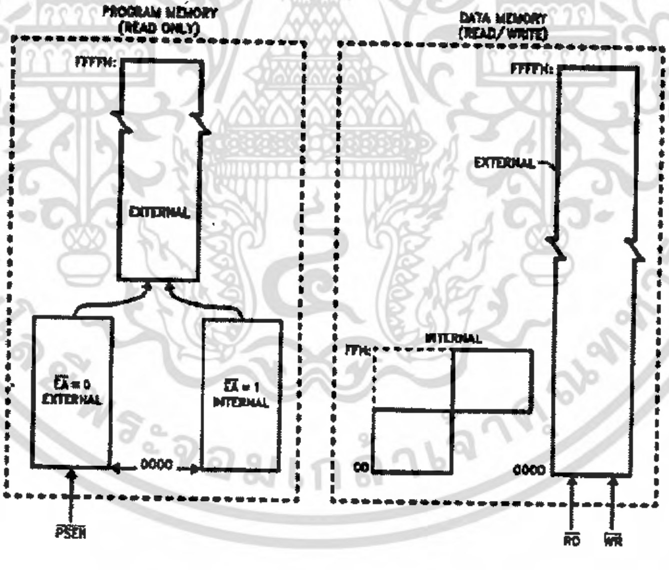
TxD , RxD เป็นขาสำหรับการรับส่งข้อมูลแบบอนุกรม

INT0 , INT1 เป็นขาสำหรับ External Interrupt

T0 , T1 เป็นขาสำหรับรับสัญญาณนาฬิกาภายนอกสำหรับเป็นอินพุตของวงจร Timer/Counter

2.3.2 การจัดหน่วยความจำของ MCS-51

MCS – 51 จะใช้สถาปัตยกรรมภายในแบบ Harvard ที่จะแยก program memory กับ data memory ออกจากกัน



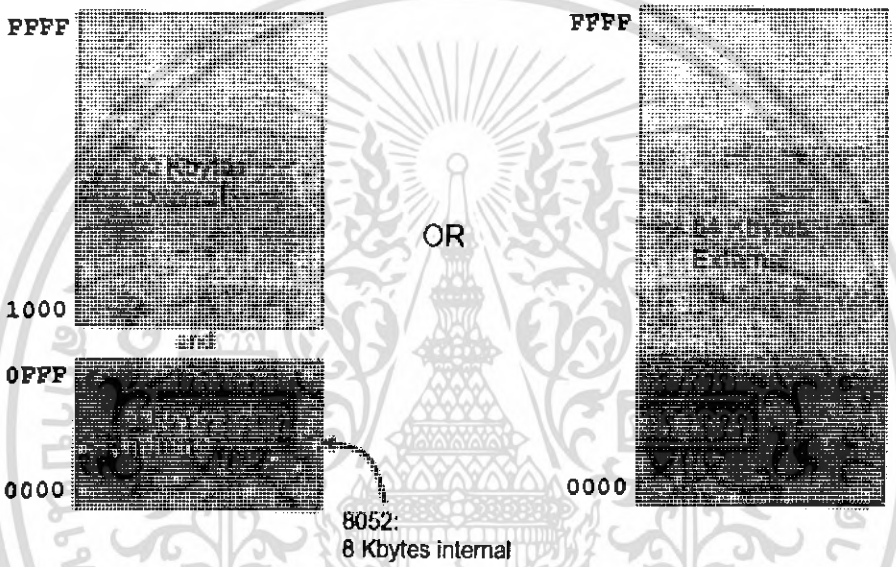
รูปที่ 2.2 การจัดหน่วยความจำของ MCS-51

หน่วยความจำโปรแกรม

89C52 มีการจัดหน่วยความจำ โดยแยกพื้นที่ของโปรแกรมและข้อมูลออกจากกัน กล่าวคือหน่วยความจำโปรแกรมจะไม่สามารถเขียนข้อมูลลงได้ใช้เป็นพื้นที่เก็บรหัสคำสั่ง ในขณะที่หน่วยความจำข้อมูลจะใช้เก็บตัวแปร และข้อมูลต่างๆ สามารถเขียนและอ่านได้ โครงสร้างของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยความจำโปรแกรม แสดงดังรูปที่ 2.4 EA เป็นขาสัญญาณควบคุมการอ่านรหัสคำสั่ง ถ้าเราให้ EA ต่อเข้ากับ +Vcc ตำแหน่งที่ CPU เริ่มอ่านคำสั่งแรกจะเป็นตำแหน่ง 0000H โปรแกรมเคาน์เตอร์จะเพิ่มค่าเพื่ออ่านคำสั่งต่อไปจนถึงตำแหน่ง 1FFFH ซึ่งเป็นพื้นที่ของหน่วยความจำแฟลชขนาด 8 กิโลไบต์นั่นเอง กรณีที่โปรแกรมมีขนาดใหญ่กว่าหน่วยความจำโปรแกรมภายใน CPU จะอ่านรหัสคำสั่งที่เก็บอยู่ในพื้นที่ส่วนที่เหลือคือตำแหน่ง 2000H ถึง FFFFH แต่ถ้าเราให้ EA ต่อลงกราวด์หรือเป็นลอจิกศูนย์ CPU จะอ่านรหัสคำสั่งจากหน่วยความจำโปรแกรมภายนอกตำแหน่ง 0000H ถึง FFFFH โดยใช้สัญญาณ PSEN ควบคุมการอ่าน

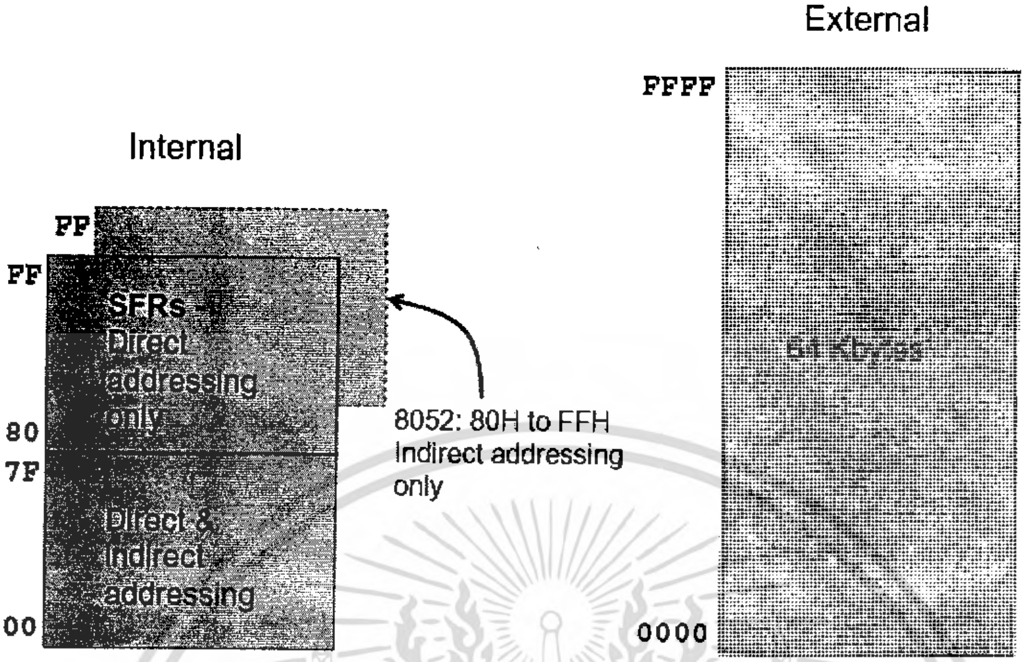


รูปที่ 2.3 หน่วยความจำโปรแกรม

หน่วยความจำข้อมูล

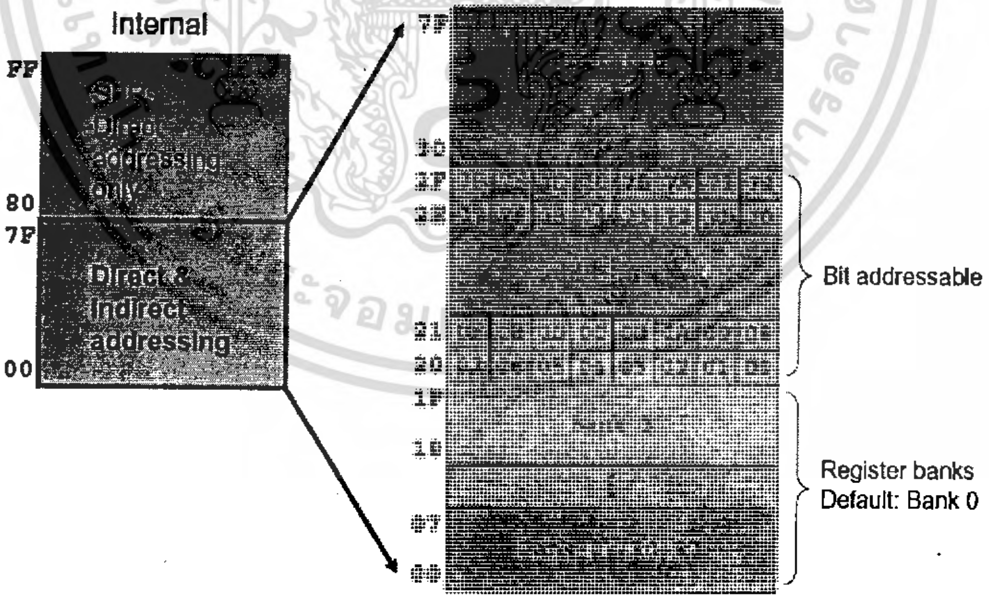
89C52 แบ่งหน่วยความจำข้อมูลออกเป็นสองพื้นที่ คือ พื้นที่ของหน่วยความจำข้อมูลบนชิพ ขนาด 256 ไบต์ตำแหน่ง 00H ถึง FFH และหน่วยความจำข้อมูลภายนอก ตำแหน่ง 0000H ถึง FFFFH หรือมีขนาด 64 กิโลไบต์ การอ่านหน่วยความจำข้อมูลภายนอก CPU จะกำเนิดสัญญาณ RD ส่วนการเขียนข้อมูล CPU จะกำเนิดสัญญาณ WR พื้นที่ของหน่วยความจำข้อมูลบนชิพ 89C52 จะพิเศษกว่าหน่วยความจำข้อมูลภายนอกกล่าวคือ ตำแหน่ง 00H ถึง 1FH จะเป็นพื้นที่ของเรจิสเตอร์ใช้งานทั่วไป R0, R1, R2, R3, R4, R5, R6,R7 จำนวน 4 ชุด ตำแหน่ง 20H - 2FH เป็นพื้นที่ที่ใช้คำสั่งประมวลผลแบบบิท โดยเราสามารถระบุตำแหน่งข้อมูลเป็นบิทได้มีค่าตำแหน่งบิทเป็น 00H - 7FH เช่นที่ตำแหน่งไบต์ 20H ภายในมีตำแหน่งบิทเป็น 00H -07H เป็นต้น พื้นที่ตำแหน่ง 30H - 7FH เป็นพื้นที่ใช้งานทั่วไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



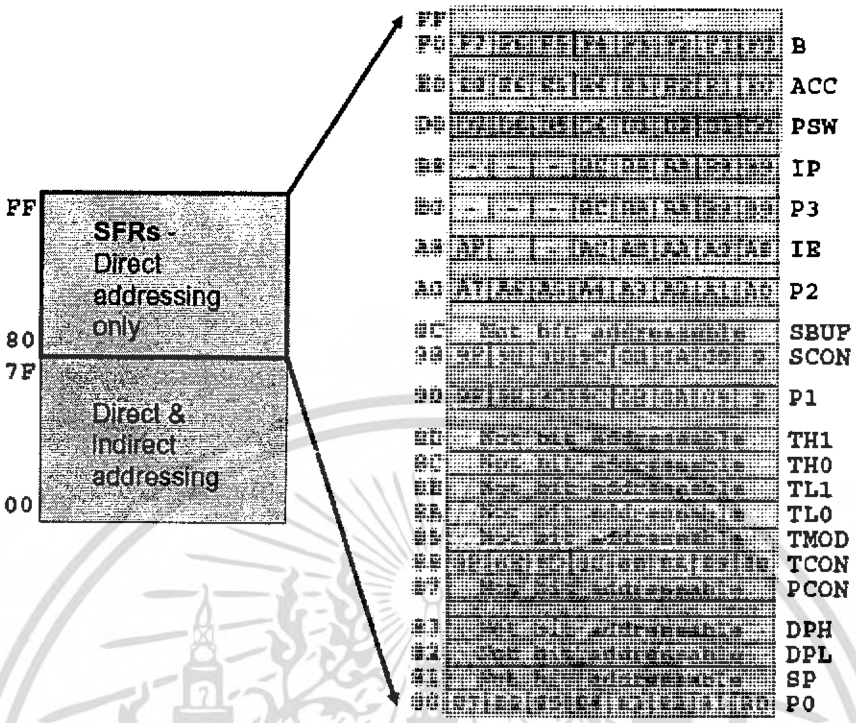
Program Memory of AT89C52

รูปที่ 2.4 แสดงหน่วยความจำข้อมูลที่เข้าถึงได้โดยตรงและโดยอ้อม



รูปที่ 2.5 แสดงส่วนของรีจิสเตอร์ R0-R7 ในแต่ละ Bank รวมทั้งส่วนที่เข้าถึงได้แบบบิท
(เข้าถึงได้ทั้งแบบโดยตรงและโดยอ้อม)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 แสดงส่วนของรีจิสเตอร์หน้าที่พิเศษหรือ Special Function Registers (เข้าถึงแบบโดยตรงเท่านั้น)

ACC (Accumulator) อยู่ที่ตำแหน่ง 0E0H เป็นรีจิสเตอร์ขนาด 8 บิตที่ใช้มาก คำสั่งที่จะอ่านหรือเก็บข้อมูลภายนอกจะต้องกระทำผ่านรีจิสเตอร์นี้เท่านั้น

B อยู่ที่ตำแหน่ง 0F0H เป็นรีจิสเตอร์ที่ใช้ในการคูณและหาร โดย รีจิสเตอร์ B จะเก็บตัวคูณและผลลัพธ์บิต 8 ถึงบิต 15 ในส่วนของคำสั่งหาร รีจิสเตอร์ B จะเก็บตัวหารและผลการหาร

SP (Stack Pointer) : อยู่ที่ตำแหน่ง 081H รีจิสเตอร์นี้จะใช้ในการชี้ตำแหนหน่วยความจำภายในที่ใช้เก็บตำแหน่ง(Address) เดิมของโปรแกรมก่อนทำคำสั่ง CALL หรือตำแหน่งที่จะใช้เก็บข้อมูลด้วยคำสั่ง PUSH และ ตำแหน่งที่จะอ่านข้อมูลออกมาในคำสั่ง POP

DPTR (Data Pointer Register) : อยู่ที่ตำแหน่ง 082H และ 081H DPTR เป็นริจิสเตอร์ขนาด 16 บิต ที่เกิดจากการนำรีจิสเตอร์ขนาด 8 บิต 2 ตัวมาต่อกันคือ DPL กับ DPH โดย DPTR จะใช้ในการชี้ตำแหน่งหน่วยในความจำ

PC (Program Counter) : PC เป็นเรจิสเตอร์ขนาด 16 บิท ใช้เป็นตัวชี้ตำแหน่งของรหัสคำสั่งจะมีค่าเป็น 0000H เมื่อรีเซ็ท CPU

PSW (Program Status Word) :อยู่ที่คำแหน่ง D0 จะใช้ในการเก็บสถานะการทำงานต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CY	AC	F0	RS1	RS0	OV	-	P

- CY : Carry Flag ใช้สำหรับบอกสถานะในการประมวลผลทางคณิตศาสตร์ว่ามีตัวทดเกิดขึ้นหรือไม่ ถ้ามีจะทำให้บิตนี้เป็น 1

- AC : Auxiliary Carry เป็นบิตที่ใช้เก็บตัวทดที่เกิดจากการคำนวณจากบิตที่ 3 ไปยังบิตที่ 4

- F0 : เป็นบิตอเนกประสงค์ที่ผู้ใช้สามารถนำไปใช้ได้

- OV : Over Flow เป็นบิตที่บอกว่ามีตัวทดเกิดขึ้นจากบิตที่ 6 ไปยังบิตที่ 7 จะใช้ในกรณีที่เราทำการคำนวณแบบ Signed Integer

- P : Parity Bit จะใช้ในกรณีที่บอกว่าค่าใน Register A เมื่ออยู่ในรูปเลขฐานสองจะมีค่าเป็น 1 เป็นคู่หรือคี่ โดยถ้าเป็นคี่ จะทำให้บิตนี้เป็น 1 แต่ถ้าเป็น คู่จะทำให้บิตนี้เป็น 0

Port0 : อยู่ที่ตำแหน่ง 080H พอร์ทนี้จะใช้ในการรับส่งตำแหน่งและข้อมูลกับหน่วยความจำหรือใช้รับ-ส่งข้อมูลก็ได้ โครงสร้างของ พอร์ท 0 จะเป็นแบบ Open Collector Bi-directional

Port1 : อยู่ที่ตำแหน่ง 090H จะคล้ายกับพอร์ท 0 เพียงแต่จะมีตัวต้านทานต่อเป็น Internal Pull Up พอร์ท 1 นี้จะใช้รับส่งข้อมูลเท่านั้น ข้อมูลที่ส่งออกมาทางพอร์ทจะถูก Latch ไว้แล้วส่งออกไปแต่ละขา ก่อนที่จะอ่านข้อมูลจะต้องเขียน 1 ไปยังบิตที่ต้องการอ่านข้อมูล เพื่อให้ FET อยู่ในสภาวะ OFF เสียก่อน

Port2: อยู่ที่ตำแหน่ง 0A0H โครงสร้างของ พอร์ท 2 จะคล้ายกับพอร์ท 0 แต่ที่แตกต่างกันคือภาค Driver ของพอร์ท 2 จะใช้งานเพียง 2 ลักษณะคือ

- ใช้ส่งค่าตำแหน่งหน่วยความจำภายนอกที่เป็น High Byte
- ใช้รับส่งข้อมูลภายนอก

Port3: อยู่ที่ตำแหน่ง 0B0H จะประกอบด้วยส่วน latch ข้อมูลที่เขียนเข้ามาทางพอร์ท 3 ทาง Internal Bus เหมือนกับพอร์ทอื่น ๆ และ พอร์ท 3 นี้จะมี Internal Pull Up ต่ออยู่ทุกบิต แต่พอร์ท 3 นี้จะใช้ทำหน้าที่อื่นด้วยโดยใช้สัญญาณจากคำสั่งควบคุมการทำงาน ในส่วน Alternative Output Function จะเป็นสัญญาณที่สร้างมาจากส่วน Timing and Control โดยจะส่งสัญญาณออกเมื่อต้องการสั่งให้ พอร์ท 3 ทำงานในฟังก์ชันอื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.3 Timer/Counter

Timer คือวงจรที่นับสัญญาณนาฬิกาจากวงจร Oscillator ที่อยู่ภายใน Counter คือวงจรที่นับสัญญาณนาฬิกาจากภายนอก การเลือกว่าจะใช้งานเป็น Timer หรือ Counter จะต้องพิจารณางานที่เราใช้ เช่น ถ้าเราต้องการใช้งานนาฬิกา หรือฐานเวลา เราต้องใช้งานเป็น Timer แต่ถ้าเราต้องการนับความถี่หรือสัญญาณนาฬิกา เราต้องใช้งานเป็น Counter

รีจิสเตอร์ที่เกี่ยวข้องกับ Timer/Counter

Timer Register : TH0 , TL0 , TH1 , TL1

ตำแหน่งหน่วยความจำภายในคือ 8CH , 8AH , 8DH , 8BH

ใน 8051 ปกติจะมีวงจร Timer/Counter อยู่ 2 ชุดโดยเป็นรีจิสเตอร์ขนาด 8 บิตสองตัวต่อเป็นขนาด 16 บิตทำให้ค่าที่นับได้สูงสุดคือ 16 บิต ซึ่งค่าที่เก็บจะอยู่ในรีจิสเตอร์ TH0 , TL0 สำหรับ Timer0 High Byte , Low Byte ตามลำดับ และรีจิสเตอร์ TH1 , TL1 สำหรับ Timer1 High Byte , Low Byte ตามลำดับ การใช้งานเราสามารถใช้งานทั้งเป็นขนาด 16บิต หรือใช้งานทีละ 8 บิตได้

TMOD : Timer/Counter Mode Register

ตำแหน่งหน่วยความจำภายในคือ 89H

เป็นรีจิสเตอร์ขนาด 8 บิตที่ควบคุม mode ในการทำงานของ Timer โดยมีรายละเอียดคือ

GA	C/T	M1	M0	GA	C/T	M1	M0
Timer1				Timer0			

Gate : เป็นบิตที่ควบคุมให้ Timer ทำงาน โดยถ้าบิตของ Timer x นี้ถูกกำหนดให้เป็น 1 จะทำให้ Timer x ทำงานเมื่อขา INT x มีลอจิกเป็นหนึ่ง และบิต TR x มีลอจิกเป็น 1 ด้วย

C/T : จะเป็นการกำหนดว่าจะให้วงจร Timer ทำงานเป็น Timer หรือ Counter โดยถ้าถูกกำหนดเป็น 1 จะทำงานเป็น Counter ถ้าถูกกำหนดเป็น 0 จะทำงานเป็น Timer M1 , M0 จะเป็นบิตที่ใช้ในการกำหนด Mode ของ Timer/Counter โดยจะมี Mode ให้เลือกใช้งานได้ทั้งหมด 4 mode ตามตารางต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TCON :Timer Control Register

ตำแหน่งหน่วยความจำภายในคือ 88H

จะทำหน้าที่ควบคุมการทำงานและคือค่าสภาวะการทำงานต่าง ๆ ของ Timer/Counter

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

IT0 : Interrupt 0 เป็นบิตที่ใช้สำหรับกำหนดวิธีการขัดจังหวะโปรแกรม ของสัญญาณจาก External Interrupt0(INT0) โดยถ้าเป็น 1 จะเกิดการขัดจังหวะเมื่อสัญญาณที่ขา INT0 เปลี่ยนจาก 1 เป็น 0 ถ้าเป็น 0 จะเกิดการขัดจังหวะเมื่อสัญญาณที่ขา INT0 เปลี่ยน เป็น 0

IE0 : บิตนี้จะเป็น 1 เมื่อสัญญาณที่ขา INT0 เป็นไปตามที่กำหนดใน IT0 แล้วทำให้เกิดการขัดจังหวะแล้วโปรแกรมตอบสนองการขัดจังหวะเริ่มทำงาน จะทำให้บิตนี้เป็น 0

IT1 : Interrupt 1 เป็นบิตที่ใช้สำหรับกำหนดวิธีการขัดจังหวะโปรแกรม ของสัญญาณจาก External Interrupt0(INT1) โดยถ้าเป็น 1 จะเกิดการขัดจังหวะเมื่อสัญญาณที่ขา INT1 เปลี่ยนจาก 1 เป็น 0 ถ้าเป็น 0 จะเกิดการขัดจังหวะเมื่อสัญญาณที่ขา INT1 เปลี่ยน เป็น 0

IE1 : บิตนี้จะเป็น 1 เมื่อสัญญาณที่ขา INT1 เป็นไปตามที่กำหนดใน IT1 แล้วทำให้เกิดการขัดจังหวะแล้วโปรแกรมตอบสนองการขัดจังหวะเริ่มทำงาน จะทำให้บิตนี้เป็น 0

TR0 : Timer0 Run Control เป็นบิตสำหรับสั่งให้ Timer เริ่มทำงานโดยเมื่อบิตนี้เป็น 0 Timer ไม่ทำการนับไม่ว่าจะอยู่ในสถานะใดๆ แต่เมื่อเป็น 1 การทำงานของ Timer จะขึ้นอยู่กับขา INT0 และบิต GATE

TF0 : Timer0 Overflow เป็นบิตสำหรับแสดงว่าการนับ Timer นับได้ถึงค่าสูงสุดที่สามารถนับได้ เช่นใน Mode 2 ที่มีการนับขนาด 16 บิตจะนับได้สูงสุด ถึง FFFFH เมื่อทำการนับครั้งจะเกิดการ Overflow คือบิต TF0 จะเป็น 1 และการนับจะกลับไปเริ่มต้นที่ 0 เมื่อโปรแกรมกระโดดไปไปทำงานที่โปรแกรมตอบสนอง Interrupt จะทำให้บิตนี้เป็น 0

TR1 : Timer1 Run Control จะเหมือนกับ TR0 แต่บิตนี้จะใช้งานสำหรับ Timer1

TF1 : Timer1 Overflowจะเหมือนกับ TR0 แต่บิตนี้จะใช้งานสำหรับ Timer1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.4 วงจรนับ/จับเวลา 2 (Timer2)

mode การทำงานของ Timer2 ประกอบด้วย

-Capture mode: สามารถเลือกใช้งานได้ 2 ลักษณะ ด้วยการกำหนดให้กับ bit EXEN2 ของ register T2CON ดังนี้

1. เมื่อกำหนด bit EXEN2 เป็น 0 Timer 2 ยังทำงานเป็นวงจรนับ / ตรวจจับเวลา เมื่อมีการ overflow ขึ้น bit ใน register TF2 จะถูกเซต และสามารถนำไปสร้างการ interrupt ขึ้นได้

2. เมื่อกำหนดค่า bit EXEN2 เป็น 1 การทำงานจะครอบคลุมการทำงานลักษณะข้างต้น แต่จะเพิ่มเมื่อมีการเปลี่ยนแปลงระดับสัญญาณทางขาสัญญาณ T2EX จาก logic สูง ไปเป็น logic ต่ำ จะมีผลทำให้ค่าข้อมูลภายใน register ของ Timer2 คือ TL2 และ TH2 ถูกนำไปใส่(Capture) ให้กับ register RCAP2L และ RCAP2H ซึ่งเป็น register หน้าที่พิเศษ หรือ SFR ที่มีใน Microcontroller เบอร์ 8052 เท่านั้น นอกจากนี้จะมีผลทำให้ bit EXF2 ภายใน register T2CON มีค่าเป็น 1 สามารถนำไปใช้งานในการ interrupt ได้เช่นกัน

2.3.5 การ interupt ใน 8051

ประเภทของการ interrupt

- External interrupt การตรวจสอบสัญญาณที่มา interrupt นี้จะสามารถกำหนดให้มีการตรวจสอบในลักษณะเมื่อได้มีการเปลี่ยนแปลงระดับสัญญาณ (Level-sensitive) ไปแล้ว หรือในช่วงเวลาขณะเริ่มมีการเปลี่ยนแปลงสัญญาณจาก logic สูงไปต่ำ (Edge-sensitive)

- Internal inetrrupt แหล่งกำเนิดสัญญาณนี้จะเป็นวงจรภายใน Microcontroller เอง เช่น วงจรนับ/จับเวลา วงจรเชื่อต่อสัญญาณอนุกรม เป็นต้น

โครงสร้างการ interrupt เกิดได้ 5 ลักษณะ คือ

- INT0 สัญญาณ interrupt จากภายนอก ทางขาสัญญาณ P3.2 โดย 8051 จะทำการสุ่มตัวอย่าง สัญญาณเมื่อสิ้นสุดทุก Machine Cycle

- INT1 สัญญาณ interrupt จากภายนอกทางขาสัญญาณ P3.3 โดย 8051 จะทำการสุ่มตัวอย่าง สัญญาณเมื่อสิ้นสุดทุก Machinc Cycle

- Timer0 สัญญาณการเกิด Overflow ของ Timcr 0

- Timer1 สัญญาณการเกิด Overflow ของ Timcr 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Serial Port การเกิด interrupt ที่เกิดขึ้นจากการรับ/ส่งข้อมูลอนุกรม ทำให้มีผลต่อ flag interrupt RI และ TI ตามลำดับเมื่อเกิดการ interrupt สัญญาณ ต่างๆขึ้น
จะส่งผลให้มีการควบคุมเพื่อสั่งให้ processor กระโดดไปทำงานที่ตำแหน่ง Address ต่างๆ ตามประเภทของแหล่งกำเนิดสัญญาณ interrupt ที่เกิดขึ้น ซึ่งปรกติควรมีการสร้าง program เหล่านี้ไว้เพื่อทำหน้าที่ย่อยบริการ interrupt

การกำหนดให้ 8051 สามารถตอบรับการ interrupt แต่ละประเภท ทำได้โดยกำหนด bit ของ ข้อมูลที่เกี่ยวข้องซึ่งมักอยู่ภายใน register TCON และ SCON หากได้ว่ามีการกำหนดค่าของ bit ซึ่งอยู่ภายใน register IE (Intereupt Enable Register) ด้วยแล้ว ก็สามารถตอบรับการ interrupt ของสัญญาณนั้นๆ ได้ นอกจากหนี้ตามแผนภาพในรูป ยังแสดงให้เห็นว่าสัญญาณ interrupt แต่ละประเภท ยังสามารถกำหนด piority ของการ interrupt ได้ 2 ลักษณะ คือ High Low
piority กล่าว คือขณะที่ประมวลผลอยู่ภายในส่วนของ program ย่อย บริเวณ interrupt ของสัญญาณที่มีระดับความ สำคัญต่ำอยู่ ก็สามารถถูก interrupt ที่มี piorrity สูงกว่าได้ แต่หากว่าเป็นสัญญาณ interrupt ที่มี piority เดียวกันหรือต่ำกว่าแล้วก็จะต้องรอให้เสร็จสิ้นการประมวลผลที่ดำเนินอยู่ก่อน

การควบคุม interrupt ตามโครงสร้างที่ด้านการจัดการ interrupt ของ 8051 สามารถกำหนดเรียกเพื่อยินยอมหรือไม่ยินยอม (Enable/Disabble) ให้มีการ interrupt แต่ละสัญญาณได้โดยใช้วิธีการกำหนดค่าของ bit ภายใน register IE

2.4 การรับส่งข้อมูลแบบอนุกรม

การรับส่งข้อมูลถ้าแบ่งตามรูปแบบการรับส่งจะแบ่งได้เป็น 3 ประเภทคือ

- Simplex
- 2Half Duplex
- Full Duplex

การส่งผ่านแบบทิศทางเดียว (Simplex)

หมายถึง รูปแบบการส่งสัญญาณให้ด้านรับได้ฝ่ายเดียว โดยไม่สามารถโต้ตอบผ่านทางการติดต่อได้เช่น การกระจายเสียงของวิทยุหรือสัญญาณโทรทัศน์ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การส่งผ่านแบบสองทิศทางแต่ต่างเวลากัน (Half-Duplex)

หมายถึง รูปแบบสัญญาณที่สถานีทั้งสองฝ่ายสามารถรับและส่งสัญญาณระหว่างกันได้โดยกำหนดว่าต้องมีด้านใดด้านหนึ่งเป็นตัวรับเสมอเช่น การใช้วิทยุสมัครเล่นในการติดต่อสื่อสารเป็นต้น

การส่งผ่านแบบสองทิศทางในเวลาเดียวกัน (Full-Duplex)

หมายถึง รูปแบบการส่งสัญญาณที่ทั้งด้านส่งและด้านรับสามารถที่จะส่งสัญญาณในเวลาเดียวกันได้โดยไม่จำเป็นต้องสลับด้านกันด้วย เช่นการสนทนาทางโทรศัพท์ เป็นต้น

การรับส่งข้อมูล (Data Communication)

ปกติแล้วในการสื่อสารสำหรับรับส่งข้อมูลโดยทั่วไปจะแบ่ง ออกเป็น 2 ประเภทคือ

- แบบขนาน (Parallel Communications)
- แบบอนุกรม (Serial Communications)

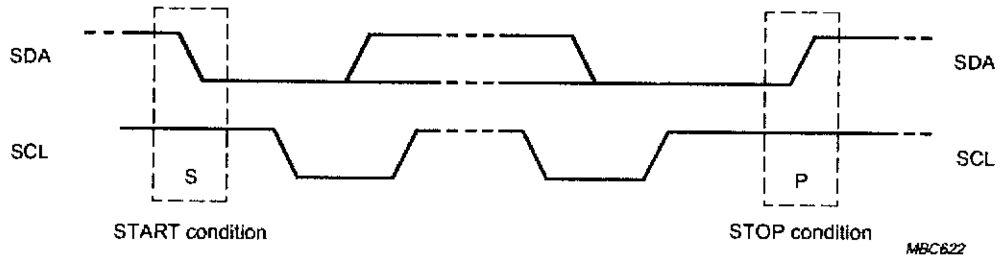
การรับส่งข้อมูลแบบอนุกรม

สำหรับการรับส่งข้อมูลแบบอนุกรม จะมีการรับส่งข้อมูลทีละบิต แล้วเวียนจนครบจำนวนบิตที่ต้องการรับส่ง ซึ่งข้อดีของระบบการรับส่งข้อมูลแบบอนุกรมคือจะใช้จำนวนสายน้อยกว่าการรับส่งแบบขนาน และสามารถสายสัญญาณรับส่งข้อมูลได้ระยะทางไกลมากกว่าแบบขนาน โดยทั่วไปการรับส่งข้อมูลแบบอนุกรมแบ่งออกเป็น 2 ประเภทคือ

- Synchronous Serial Communication เป็นการรับส่งข้อมูลแบบขึ้นกับเวลา

-Asynchronous Scrial Communication เป็นการรับส่งข้อมูลแบบไม่ขึ้นกับเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 เป็นการรับส่งข้อมูลแบบขึ้นกับเวลา Synchronous Serial Communication

รูปที่ 2.8 เป็นการรับส่งข้อมูลแบบไม่ขึ้นกับเวลา Asynchronous Serial Communication

RS-232 EIA

เป็นมาตรฐานในการติดต่ออนุกรมแบบ Asynchronous ตามมาตรฐานของ Electronic Industries Association ซึ่งจะใช้เป็นมาตรฐานสำหรับการติดต่ออุปกรณ์โดยทั่วไป

ลักษณะเฉพาะของการส่งแบบ RS-232 EIA

- ใช้การรับส่งแบบ Common
- ใช้ระดับในการส่งข้อมูล Space หรือ Logic 0 จะอยู่ในช่วง +3 ถึง +25 V
- ใช้ระดับในการส่งข้อมูล Mark หรือ Logic 1 จะอยู่ในช่วง -3 ถึง -25 V
- ช่วงระดับสัญญาณ +3 ถึง -3V จะเป็น Undefined
- Open circuit voltage จะต้องไม่เกิน 25 V เมื่อเทียบกับ GND
- Short circuit current จะต้องไม่เกิน 500 mA
- สามารถรับ-ส่งได้ระยะทางไกลสูงสุดคือ 45 ฟุต
- อัตราการส่งข้อมูลสูงสุดคือ 20kbps (ถ้ามี driver จะสามารถเพิ่มได้เป็น 115kbps)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

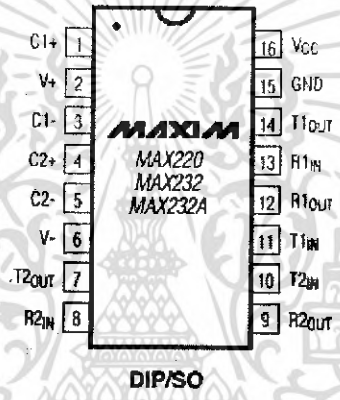
สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

UART

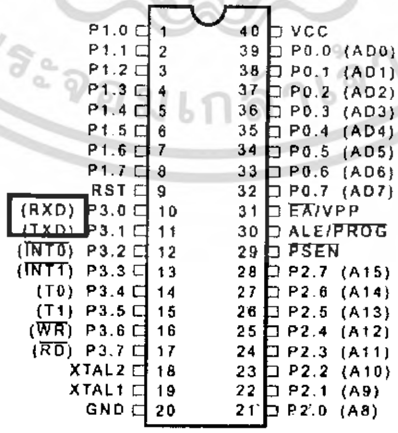
UART : Universal Asynchronous Receiver / Transmitter จะทำหน้าที่แปลงจากข้อมูลที่เป็นแบบขนาน ให้เป็นข้อมูลที่เป็นแบบอนุกรม ซึ่งการทำงานจะคล้ายกับว่าเป็น PISO (Parallel In , Serial Out)ตัวอย่างของ ไอซี UART เช่น 8250 , 16450 , 16550 เป็นต้น

RS-232 Driver

เนื่องจากในมาตรฐานของ RS-232 จะใช้การรับส่งที่มีระดับแรงดันคือ +3 ถึง +25V และ -3 ถึง -25V ซึ่งจะแตกต่างกับระบบ Microprocessor ที่ใช้ระดับแรง 0 ถึง +5V ดังนั้นจะต้องมีไอซีที่ทำหน้าที่แปลงระดับแรงดันดังกล่าว ตัวอย่างของไอซีประเภทนี้เช่น DS275 , MAX232

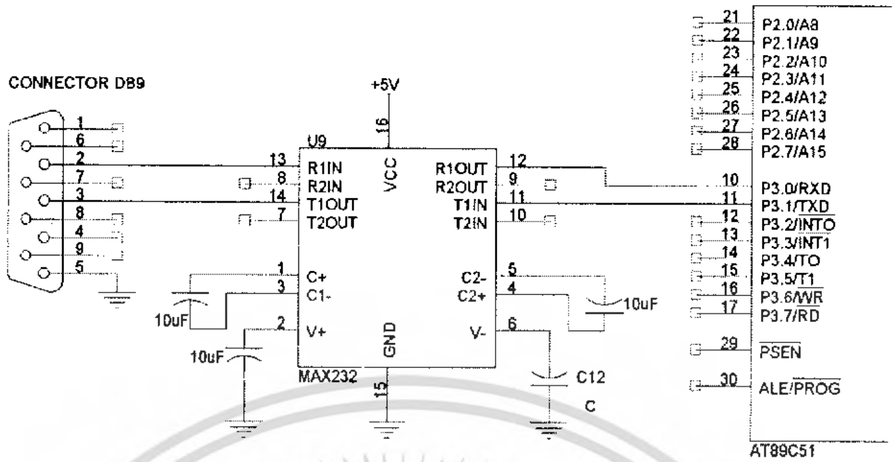


รูปที่ 2.9 แสดงขาต่างๆ ของ MAX232



รูปที่ 2.10 แสดงตำแหน่งขา TXD และ RXD ของ 8051

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
62316



รูปที่ 2.11 แสดงการเชื่อมต่อ MAX232 กับ 8051

MCS-51 และ RS-232

สำหรับ Microprocessor เบอร์ 8051 จะมีวงจร UART บรรจุอยู่ภายในตัว อยู่แล้ว ซึ่งถ้าเราต้องการใช้งาน RS-232 ก็เพียงต่อไอซี Driver เช่น DS275 , MAX232 เท่านั้นก็จะสามารถใช้งานได้ Register ที่เกี่ยวข้อง

SCON : Serial Control มีตำแหน่งหน่วยความจำภายในเท่ากับ 98H

RI : Receive Interrupt Flag

บิตนี้จะ set โดยฮาร์ดแวร์ให้มีค่าเป็น 1 หรือ 0 โดยใน Mode 0 บิตนี้จะมีค่าเป็น 1 เมื่อข้อมูลเข้ามาครบ 8 บิต ส่วนใน Mode อื่นๆบิตนี้จะมีค่าเป็น 1 เมื่อข้อมูลเข้ามาถึงครึ่งหนึ่งของ stop bit สำหรับการ clear จะสามารถทำได้โดยโปรแกรมบิตนี้จะใช้บอกว่าข้อมูลรับเข้ามาอยู่ใน SBUF เรียบร้อยแล้ว

TI : Transmit Interrupt Flag

บิตนี้จะ set โดยฮาร์ดแวร์ให้มีค่าเป็น 1 หรือ 0 โดยใน Mode 0 บิตนี้จะมีค่าเป็น 1 เมื่อข้อมูลส่งออกครบ 8 บิต ส่วนใน Mode อื่นๆบิตนี้จะมีค่าเป็น 1 เมื่อส่งข้อมูลที่เป็น stop bit ออกไป สำหรับการ clear จะสามารถทำได้โดยโปรแกรม บิตนี้จะใช้บอกว่าได้ส่งข้อมูลที่อยู่ใน SBUF เรียบร้อยแล้ว

RB8:

บิตนี้จะใช้ในการเก็บข้อมูลที่เป็นบิตที่ 9 สำหรับการรับส่งใน Mode 2,3 ส่วนใน Mode 1 จะใช้ในการเก็บ Stop bit ซึ่งคือค่า 1 สำหรับใน Mode 0 จะไม่มีการใช้งานบิตนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TB8:

ใช้สำหรับเก็บค่าบิตที่ 9 ที่ใช้ในการส่งใน mode 2,3 ส่วนใน mode อื่น จะไม่มีการใช้งานบิตนี้

REN : Receive Enable

ใช้สำหรับกำหนดให้ทำการรับข้อมูลจากพอร์ทอนุกรม โดยถ้าบิตนี้เป็น 1 ก็จะทำการรับข้อมูล ถ้าบิตนี้เป็น 0 ก็จะไม่มีการรับข้อมูล บิตนี้เราสามารถ set และ clear ได้โดยโปรแกรม

Mode ในการทำงานของพอร์ทอนุกรม

Mode 0

ใน mode นี้จะรับส่งข้อมูลผ่านทางขา RxD ส่วนขา TxD ใช้สำหรับการส่งสัญญาณ Clock สำหรับการเลื่อนข้อมูล (Shift) โดยข้อมูลจะมีขนาด 8 บิตเท่านั้น และอัตราการส่งข้อมูลจะเท่ากับสัญญาณนาฬิกาหารด้วย 12

Mode 1

ใน mode นี้ชุดข้อมูลที่รับ ส่งจะมีขนาด 10 คือ 1 start bit 8 data bit และ 1 stop bit ข้อมูลที่รับส่งจะต้องผ่านทางรีจิสเตอร์ SBUF สำหรับความเร็วของการรับส่งใน mode นี้จะสามารถกำหนดได้จาก Timer1

Mode 2

ใน mode นี้จะมีการรับส่งข้อมูลทั้งหมด 11 บิตโดยประกอบด้วย 1 start bit , 9 data bit และ 1 stop bit โดยบิตข้อมูลที่ 9 ที่เพิ่มเข้าไปปกติใช้เป็นบิตสำหรับตรวจสอบ (Parity bit)ว่าข้อมูลที่รับส่งถูกต้องหรือเปล่า สำหรับความเร็วในการรับส่งจะมีค่าคือถ้าบิต SMOD ในรีจิสเตอร์ PCON มีค่าเป็น 0 อัตราการส่งข้อมูลจะเท่ากับ สัญญาณนาฬิกา หารด้วย 64 และถ้ามีค่าเป็น 1 อัตราการส่งข้อมูลจะเท่ากับ สัญญาณนาฬิกา หารด้วย 32

Mode 3

ใน mode นี้จะคล้ายกับใน mode 2 แต่อัตราการรับส่งข้อมูลจะขึ้นอยู่กับการ overflow ของ Timer1

2.5 การรับส่งข้อมูลด้วยชุดรับส่งข้อมูล RS232 แบบไร้สาย

เป็นชุด Signal Converter สำหรับใช้แปลงสัญญาณระหว่าง RS232 และ RF-Wireless โดยในโหมดการทำงานของการส่งข้อมูล (Transmitter) จะทำหน้าที่รอรับข้อมูลจากพอร์ตสื่อสารอนุกรม RS232จากขา RX แล้วแปลงเป็นสัญญาณความถี่ (GFSK) ส่งออกไปในอากาศ และในทางกลับกันในโหมดการทำงานแบบรับ (Receiver) ชุด ET-RF2.4G V1.0 ก็จะทำหน้าที่คอยตรวจจับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลที่อยู่ในรูปของสัญญาณความถี่ (GFSK)จากด้าน RF เพื่อแปลงกลับเป็นข้อมูลแบบ RS232 ส่งออกไปทางขา TX ได้ด้วย

ซึ่งจะเห็นได้ว่าชุดแปลงสัญญาณ ET-RF24G V1.0 นั้น สามารถนำไปต่อใช้งานร่วมกับพอร์ตสื่อสารอนุกรม แบบ RS232 เพื่อใช้งานในลักษณะของการสื่อสารอนุกรมแบบไร้สาย (Wireless Transceiver) ได้โดยตรงโดยจะมีข้อดีกว่าคือสามารถรับส่งข้อมูลกันได้ในระยะทางที่ไกลกว่า RS232 หลายเท่าตัว และประการสำคัญ คือ ไม่จำเป็นต้องใช้สายสัญญาณที่เป็นตัวนำสัญญาณทางไฟฟ้าในการสื่อสารข้อมูลกันทำให้สามารถเปลี่ยนแปลงหรือเคลื่อนย้ายจุดรับส่งข้อมูลได้ตลอดเวลา ซึ่งถ้าเป็นการรับส่งข้อมูลด้วยระบบ RS232 แบบที่ใช้สายสัญญาณนั้นจะเกิดความยุ่งยากในการติดตั้งสายสัญญาณเป็นอย่างมาก

• **RF Data Rate** ใช้สำหรับกำหนดความเร็วในการรับส่งข้อมูลทางด้าน RF ของ ET-RF24G V1.0 ซึ่งจะต้องกำหนดให้เครื่อง ET-RF24G V1.0 ทุกๆตัว ที่จะนำมาใช้ติดต่อสื่อสารกันมีค่าอัตราความเร็วในการรับส่งข้อมูลด้าน RF หรือ RF Data Rate นี้มีค่าเท่ากันทั้งหมดซึ่งถ้ากำหนดค่าความเร็วต่างกันจะไม่สามารถรับส่งข้อมูลกันได้ซึ่งค่าอัตราความเร็วในการส่งข้อมูลนี้จะมีผลต่อระยะทางการรับส่งข้อมูลด้วยซึ่งถ้าใช้ความเร็วในการส่งสูง (1Mbps) จะทำให้รัศมีการรับส่งข้อมูลได้ระยะทางสั้นลง แต่ถ้าใช้ความเร็วในการรับส่งข้อมูลที่ช้าลง (250Kbps) จะทำให้ได้รัศมีการรับส่งไกลขึ้นโดยค่า RF Data Rateสามารถกำหนดได้ 2 ค่าคือ

- 250 Kbps
- 1 Mbps

• **RF Operation Mode** ใช้สำหรับกำหนดโหมดการทำงานของ ET-RF24G V1.0 ซึ่งสามารถกำหนดหน้าที่การทำงานได้ 3 แบบด้วยกันคือ

- RF Receive Only เป็นการกำหนดให้ ET-RF24G V1.0 ทำหน้าที่เป็นฝ่ายรอรับข้อมูลทางด้านRF เพื่อเปลี่ยนเป็นข้อมูลแบบ RS232 และส่งออกไปทางด้านขา TX ของ RS232 ตลอดเวลา

- RF Transmit Only เป็นการกำหนดให้ ET-RF24G V1.0 ทำหน้าที่เป็นฝ่ายรอรับข้อมูลทางด้าน RS232 จากขา RX เพื่อเปลี่ยนเป็นข้อมูลแบบ GFSK และส่งออกไปทางด้าน RF ตลอดเวลา

- RF Auto Direction เป็นการกำหนดโหมดการทำงานแบบ Half Duplex 2 ทิศทาง ซึ่งสามารถสลับโหมดการทำงานระหว่างการรับและส่งข้อมูลได้เองโดยอัตโนมัติ โดยในโหมดการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำงานนี้เครื่อง ET-RF24G V1.0 จะรอตรวจสอบข้อมูลทั้งจากด้าน RS232 และด้าน RF อยู่ตลอดเวลาโดยถ้าได้รับข้อมูลจากด้าน RS232 ก็จะทำการแปลงแล้วส่งออกทางด้าน RF จากนั้นก็จะกำหนดให้ด้าน RF กลับมาเป็นฝ่ายรอรับข้อมูลตามเดิม และเมื่อได้รับข้อมูลจากด้าน RF ก็จะแปลงเป็นข้อมูลแล้วส่งออกไปทางด้าน RS232 โดยอัตโนมัติ

• **RF Power Gain** เป็นการกำหนดกำลังส่งของวงจร RF Power ที่ใช้ในการส่งข้อมูล โดยค่า +0dBmเป็นค่ากำลังส่งสูงสุด ส่วน –20dBm เป็นค่ากำลังส่งต่ำสุด โดยสามารถกำหนดได้ 4 ระดับ คือ

- -20dBm (กำลังส่งต่ำสุด)
- -10dBm
- -5dBm
- +0dBm (กำลังส่งสูงสุด)

• **RXD ID Code** เป็นรหัส ID Code ของเครื่อง ET-RF24G V1.0 ในโหมดของการรับข้อมูลจาก RF โดยเมื่อเครื่อง ET-RF24G V1.0 ด้านส่งจะทำการส่งข้อมูลออกไปทาง RF นั้นจะมีการระบุหมายเลข ID Code ของด้านรับรวมไปกับชุดข้อมูลด้วยเสมอ u3650 โดยเมื่อเครื่อง ET-RF24G V1.0 ที่อยู่ทางด้านรับทำการรับข้อมูลจากด้าน RF ได้ อันดับแรกมันจะทำการเปรียบเทียบรหัส ID Code ที่รวมมากับข้อมูลที่รับมาได้ว่าตรงกับรหัสของ RXD ID Code ที่กำหนดไว้ในตัวมันหรือไม่ ซึ่งถ้าถูกต้องก็จะแยกเอาเฉพาะส่วนของข้อมูลที่รับเข้ามาได้เพื่อเปลี่ยนเป็นข้อมูลแบบ RS232 แล้วส่งออกไปทางด้าน TX ของ RS232แต่ถ้ารหัส ID Code ที่รับมาได้ไม่ตรงกับรหัส RXD ID Code ที่กำหนดไว้ เครื่อง ET-RF24G V1.0 จะทิ้งข้อมูลชุดนั้นไปทันทีโดยค่า RXD ID Code นั้นสามารถกำหนดได้ 256 ค่าในรูปแบบของเลขฐานสิบหก (00H-FFH)

• **TXD ID Code** เป็นรหัส ID Code ปลายทางที่จะส่งข้อมูลไปหา โดยที่เครื่อง ET-RF24G V1.0 ที่ถูกกำหนดให้ทำหน้าที่เป็นฝ่ายส่งข้อมูลนั้น เมื่อมันสามารถรับข้อมูลจาก RS232 ได้แล้ว มันจะทำการนำเอาข้อมูลนั้นไปเข้ารหัสรวมกับ TXD ID Code ที่กำหนดไว้ แล้วส่งออกไปทางด้าน RF โดยรหัสของTXD ID Code นี้หมายถึงรหัส RXD ID Code ของฝ่ายรับที่ต้องการส่งข้อมูลไปหานั่นเอง โดยค่า TXD ID Code นั้นสามารถกำหนดได้ 256 ค่าในรูปแบบของเลขฐานสิบหก (00H-FFH)

• **RF Frequency Channel** เป็นการกำหนดค่าของช่องความถี่ที่จะใช้ในการรับส่งข้อมูลกัน โดยสามารถเลือกกำหนดช่องความถี่ได้สูงสุดมากถึง 125 ช่อง (0-124) โดยการที่เครื่อง ET-RF24G V1.0 จะทำการรับส่งข้อมูลกันได้นั้นจะต้องกำหนดช่องความถี่ที่ตรงกันและใช้อัตราความเร็ว RF Data Rate ที่เท่ากันด้วย ซึ่งที่สามารถเลือกกำหนดช่องความถี่ RF Frequency Channel ได้นั้น จะมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประโยชน์เป็นอย่างมากในกรณีที่มีการใช้งานเครื่อง ET-RF24G V1.0 จำนวนหลายๆกลุ่ม ในบริเวณพื้นที่ใกล้เคียงกัน โดยให้กำหนดช่องความถี่ของ ET-RF24G V1.0 กลุ่มที่จะสื่อสารข้อมูลร่วมกันไว้ที่ช่องความถี่เดียวกันส่วนกลุ่มอื่นๆก็ให้เลือกกำหนดช่องความถี่ที่แตกต่างกันออกไปเพื่อลดปัญหาการรบกวนกัน

การรับส่งข้อมูล 2 ทิศทาง (Half Duplex) แบบ จุดต่อจุด (Point-to-Point)

รูปที่ 2.12 รูปแสดงการรับส่งข้อมูลแบบจุดต่อจุด

สำหรับตัวอย่างนี้จะเป็นการรับส่งข้อมูลระหว่างอุปกรณ์ที่มีการสื่อสารอนุกรมแบบ RS232 จำนวน 2 ชุดโดยต้องใช้รูปแบบการสื่อสารแบบ Half Duplex หรือผลัดกันรับผลัดกันส่ง กล่าวคือ ด้านรับจะต้องทำการรอรับข้อมูลจากด้านส่งจนครบทั้งหมด แล้วจึงจะส่งข้อมูลตอบกลับไปได้ ซึ่งจะไม่สามารถส่งข้อมูลสวนทางกลับไปในขณะที่กำลังรับข้อมูลอยู่ได้โดยการสื่อสารแบบนี้ฝ่ายรับข้อมูลจะต้องรอให้รับข้อมูลได้ครบทั้งหมดเสียก่อน จากนั้นจึงจะส่งข้อมูลตอบกลับไปได้

2.6 วีดีโอเซ็นเดอร์ (VIDEO SENDER)

การต่อเครื่องวีดีโอเซ็นเดอร์

1.ต่อสายสัญญาณวีดีโอจากเครื่องเล่นวีดีโอที่เขียนว่า VIDEO OUT มาเข้าจุดVIDEO IN ของเครื่องส่งสัญญาณวีดีโอ

2. ต่อสัญญาณเสียงจากเครื่องเล่นวีดีโอที่เขียนว่า AUDIO OUT เข้ามาจุด AUDIO IN ของเครื่องส่งสัญญาณวีดีโอ

3.เปิดไฟเข้าเครื่องวีดีโอและเครื่องส่งสัญญาณวีดีโอ แล้วกดปุ่มเพลย์ของเครื่องเล่นวีดีโอ เพื่อให้มีสัญญาณส่งออกอากาศไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.เปิดเครื่องรับโทรทัศน์แล้วปรับเครื่องรับให้ช่องรับ UHF ถ้าเป็นเครื่องรับที่ปรับคลื่นเป็นแบบอัตโนมัติเพียงกดปุ่มปรับภาพเพียงครั้งเดียวเครื่องจะตรวจหาความถี่เองจนเจอสัญญาณที่จะรับซึ่งได้ทั้งภาพและเสียงที่ชัดเจน

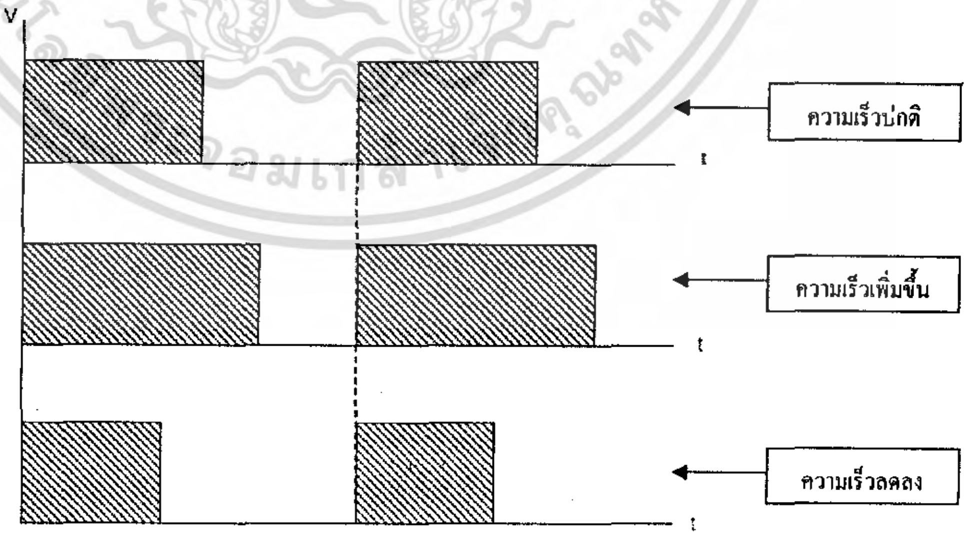
5.ถ้าภาพและเสียงชัดเจนไม่พร้อมกันให้ปรับ FREQUENCY ADJUST ในกล่องของเครื่องส่งสัญญาณวีดีโอ ในบางครั้งอาจจำเป็นต้องปรับกระป๋อง IF ของเสียงด้วย ปกติจะไม่มีปัญหาเพราะปรับแต่งมาจากโรงงานแล้วนอกจากปรับแต่งแรงดันของภาพและเสียงจนเป็นที่พอใจ

6. เครื่องรับโทรทัศน์ที่ใช้สายอากาศภายในแบบหูกระต่ายอาจจำเป็นต้องปรับความยาวของหูกระต่ายช่วยด้วยจนกว่าภาพจะชัดเจน ระการรับจะยิ่งไกลถ้าเครื่องรับใช้สายอากาศภายนอกแบบ UHF

2.7 ส่วนตัวรถ

2.7.1 การขับมอเตอร์ไฟตรง

ในการขับมอเตอร์ไฟตรงนั้นถ้าต้องการให้มอเตอร์ปรับความเร็วรอบได้เราจะต้องใช้เทคนิค PWM (Pulse Width Modulations) โดยเราจะป้อนสัญญาณกระตุ้นให้มอเตอร์หมุนเป็นPulse ที่มีความถี่ค่าหนึ่งแล้วเราปรับความกว้าง Pulseให้เพิ่มขึ้นหรือลดลงก็จะเป็นการปรับความเร็วมอเตอร์ให้เพิ่มหรือลดตามความกว้าง Pulse โดยการใช้เทคนิค PWM นี้ต้องกำหนดความถี่ที่เราป้อนให้กับมอเตอร์ให้เหมาะสมถ้าเรากำหนดมากไปการใช้เทคนิค PWM จะไม่ได้ผลแต่ถ้าเรากำหนดน้อยไปมอเตอร์อาจจะหมุนไม่สม่ำเสมอ

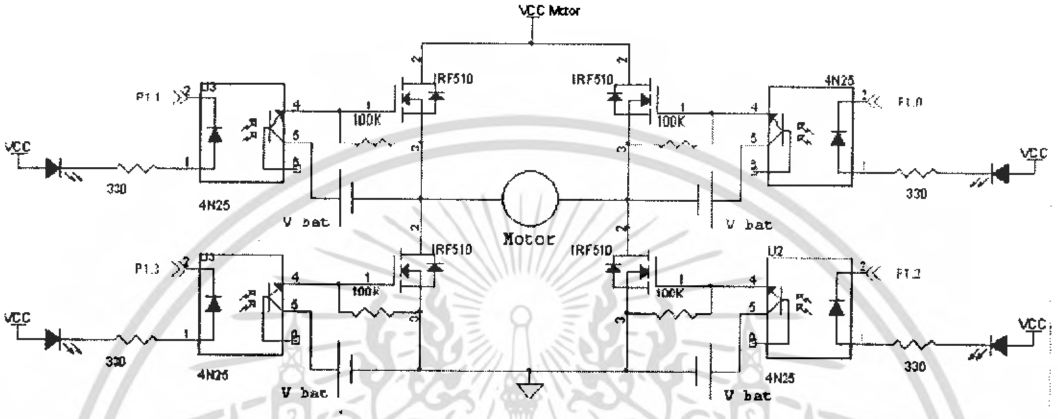


รูปที่ 2.13 แสดงการกระตุ้นด้วยเทคนิค PWM ที่ความเร็วรอบต่างๆ

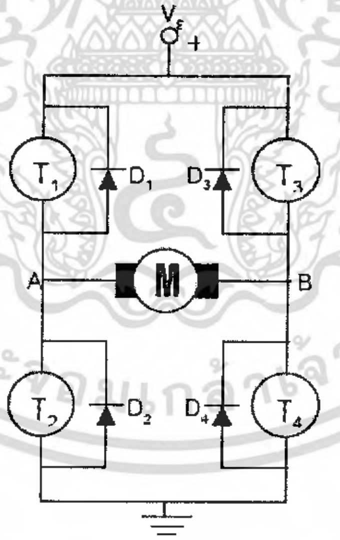
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.2 การขับมอเตอร์ไฟตรงแบบ H-Bridge

การขับมอเตอร์แบบ H-Bridge นี้เราสามารถใช้ไมโครคอนโทรลเลอร์ขับมอเตอร์ไฟตรงให้หมุนได้ทั้ง CW (Clock wise) และ CCW (Counterclock wise)



รูปที่ 2.14 แสดงวงจรขับมอเตอร์ไฟตรง H-Bridge

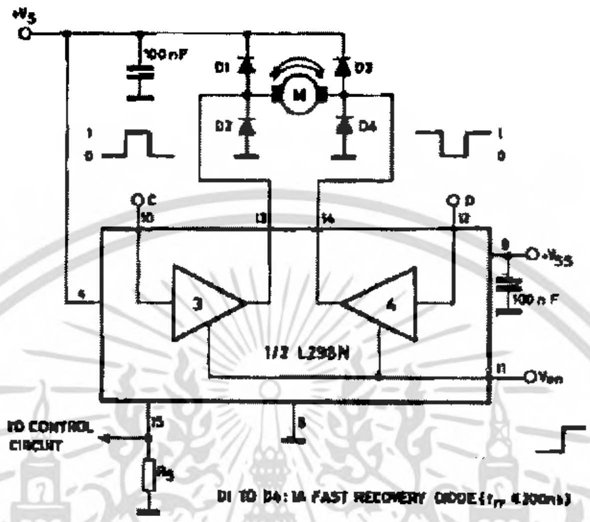


รูปที่ 2.15 แสดงวงจรสมมูลย์ H-Bridge

จากวงจรสมมูลจะพบว่าในการสั่งงานให้ วงจร H-Bridge ทำงานจะทำได้โดยสั่งให้ T1 และ T4 ON หรือ สั่งให้ T2 และ T3 ON แต่มีเงื่อนไขว่า T2 จะต้องไม่ ON พร้อมกับ T4 เพราะถ้า T2 ON พร้อมกับ T4 จะทำให้เสมือนว่าเป็นการ Short Circuit ระหว่าง Supply Vs

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการขับมอเตอร์ไฟตรงนั้นถ้าใช้วงจร H-Bridge ดังรูปที่ 3.5 นั้นจะต้องต่ออุปกรณ์ค่อนข้างมากดังนั้นจึงแนะนำให้ใช้ไอซีเบอร์ L298 Full Bridge Driver มาใช้ในการขับมอเตอร์ไฟตรง



รูปที่ 2.16 Full Bridge Driver Circuit

คุณสมบัติของ L298 Full Bridge Driver

- สามารถใช้ไฟเลี้ยงสูงสุด 48 V
- สามารถขับกระแสโหลดได้สูงสุด 4 A
- สามารถต่อกับ 8051 เพื่อใช้สัญญาณดิจิตอลควบคุมได้โดยตรง
- เพื่อป้องกันสัญญาณรบกวน ลอจิก 0 สามารถเพิ่มแรงดันได้ถึง 1.5 V

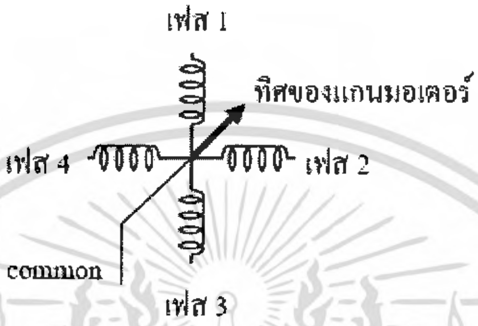
2.7.3 การควบคุม Stepping Motor แบบยูนิโพลาร์

สเต็ปปิ้งมอเตอร์แบบนี้มีการพันขดลวด 2 ขดบนแต่ละขั้วแม่เหล็กของสเตเตอร์ แต่ละขดแบ่งเป็น 2 เฟส รวมมอเตอร์ทั้งตัวจะมี 4 เฟสคือ เฟส 1,2,3 และ 4 มีการต่อสายออกมาจากขดลวดแต่ละขดเพื่อจ่ายไฟเลี้ยง ทำให้สเต็ปปิ้งมอเตอร์แบบนี้มีทั้งแบบ 5 สายและ 6 สาย ถ้าเป็นแบบ 5 สาย จะเป็นการนำสายไฟเลี้ยงของขดลวดทั้งสองมาต่อรวมกันเป็นสายเดียว การกระตุ้นและควบคุมการหมุนของมอเตอร์ให้เคลื่อนที่ไปแต่ละสเต็ปทำได้โดยจ่ายกำลังไฟฟ้าไปยังขดลวดที่ละขดบนสเตเตอร์ ซึ่งต้องป้อนเป็นแบบซีเควนเชียลในรูปแบบที่ถูกต้องด้วย สามารถแบ่งได้เป็น 3 รูปแบบคือ แบบหนึ่งเฟส,แบบ 2 เฟส (two phase) และแบบครึ่งสเต็ป (half step)

แบบหนึ่งเฟสหรือแบบฟูลสเต็ป (full step) เป็นการกระตุ้นที่มีรูปแบบง่ายที่สุด โดยทำการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

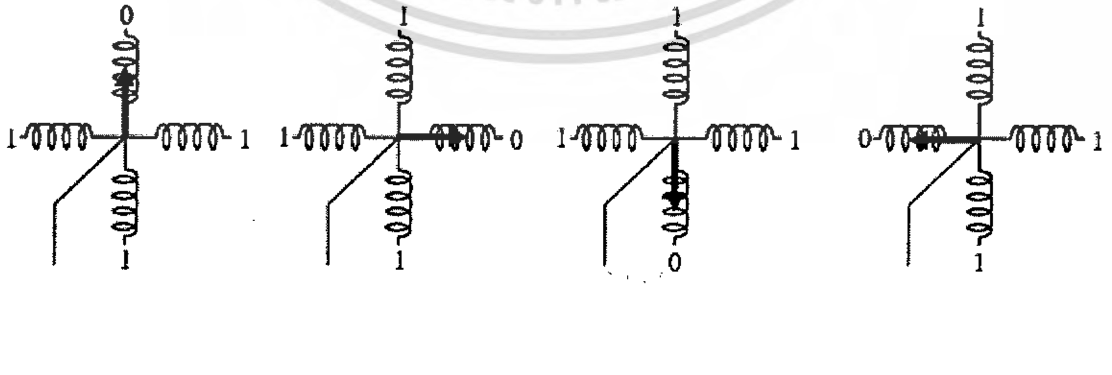
กระตุ้นขดลวดทีละขดในเวลาหนึ่งไล่เรียงถัดกันไป เช่นเริ่มต้นที่ขดที่ 1,2,3,4 แล้ววนกลับมาขดที่ 1 วนไปเรื่อยๆ หรือเริ่มที่ขดที่ 1 แล้วย้อนไปยังขดที่ 4,3,2 แล้วกลับมายังขดที่ 1 อีกครั้ง ซึ่งทำให้ทิศทางของการหมุนสวนกันในการกระตุ้นรูปแบบนี้จึงมีขดลวดเพียงขดเดียวในเวลาหนึ่งที่ถูกกระตุ้นเท่านั้นวงจรกระตุ้นแบบเวฟจึงมีราคาถูกและง่ายขั้นตอนการทำงานต่างๆ แสดงดังในตาราง 2.4



รูปที่ 2.17 แสดงการวางของขดลวดแต่ละเฟสของ stepping motor

ตารางที่ 2.4 แสดงลำดับการทำงานของขดลวดในแต่ละเฟสของมอเตอร์ เมื่อได้รับการกระตุ้นแบบหนึ่งเฟส

สเต็ปที่	เฟสที่ 1	เฟสที่ 2	เฟสที่ 3	เฟสที่ 4
1	ทำงาน	-	-	-
2	-	ทำงาน	-	-
3	-	-	ทำงาน	-
4	-	-	-	ทำงาน



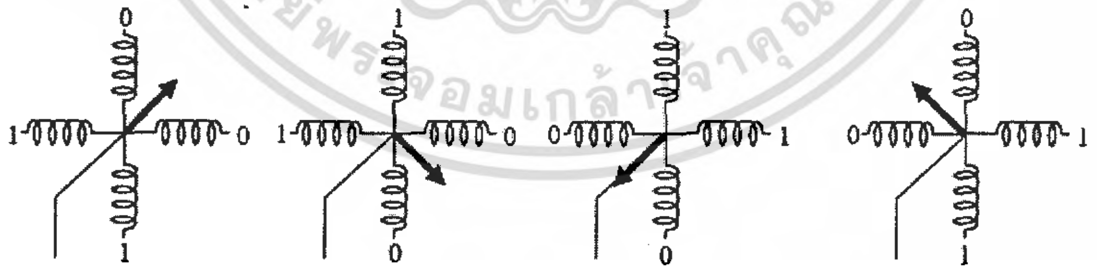
รูปที่ 2.18 แสดงการหมุนของแกนมอเตอร์เมื่อได้รับการกระตุ้นแบบหนึ่งเฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบ 2 เฟสเป็นการกระตุ้นซึ่งคล้ายกับแบบหนึ่งเฟส แต่การกระตุ้นแบบนี้จะทำการกระตุ้นโดยจ่ายกำลังไฟฟ้าไปที่ขดลวด 2 ขด ที่อยู่ใกล้กันในเวลาเดียวกัน และเรียงถัดกันไปเช่นเดียวกับแบบเวฟ ดังตัวอย่าง ขดลวดชุดแรกที่ถูกกระตุ้นจะเป็นขดที่ 1 และ 2 ตามด้วยการกระตุ้นขดที่ 2 และ 3 ต่อไปเป็นขดที่ 3 และ 4 ถัดไปเป็นขดที่ 4 และ 1 แล้วกลับมาที่ขดที่ 1 และ 2 วนไปตามลำดับเช่นนี้หรือเริ่มที่ขด 1 และ 4 ตามด้วยขดที่ 4 และ 3 ถัดไปเป็นขดที่ 3 และ 2 ต่อไปเป็นขดที่ 2 และ 1 แล้ววนกลับมาที่ขดที่ 1 และ 4 ทิศทางการหมุนจะสวนทางกัน การกระตุ้นสเต็ปปิ้งมอเตอร์แบบนี้สามารถเพิ่มแรงบิดได้มากกว่าแบบเวฟโรเตอร์จะเคลื่อนที่ด้วยแรงดึงอย่างเต็มแรงจาก 2 ขดลวดที่ถูกกระตุ้นพร้อมกัน และต่อไปด้วยแรงดึงจากอีก 2 ขดลวดถัดไป สำหรับข้อเสียคือการกระตุ้นแบบนี้ต้องใช้กำลังไฟฟ้ามากขึ้น ขั้นตอนการทำงานต่างๆ แสดงดังในตารางที่ 2.5

ตารางที่ 2.5 แสดงลำดับการทำงานของขดลวดในแต่ละเฟสของมอเตอร์ เมื่อได้รับการกระตุ้นแบบสองเฟส

สเต็ปที่	เฟสที่ 1	เฟสที่ 2	เฟสที่ 3	เฟสที่ 4
1	ทำงาน	ทำงาน	-	-
2	-	ทำงาน	ทำงาน	-
3	-	-	ทำงาน	ทำงาน
4	ทำงาน	-	-	ทำงาน



รูปที่ 2.19 แสดงการหมุนของแกนมอเตอร์เมื่อได้รับการกระตุ้นแบบสองเฟส

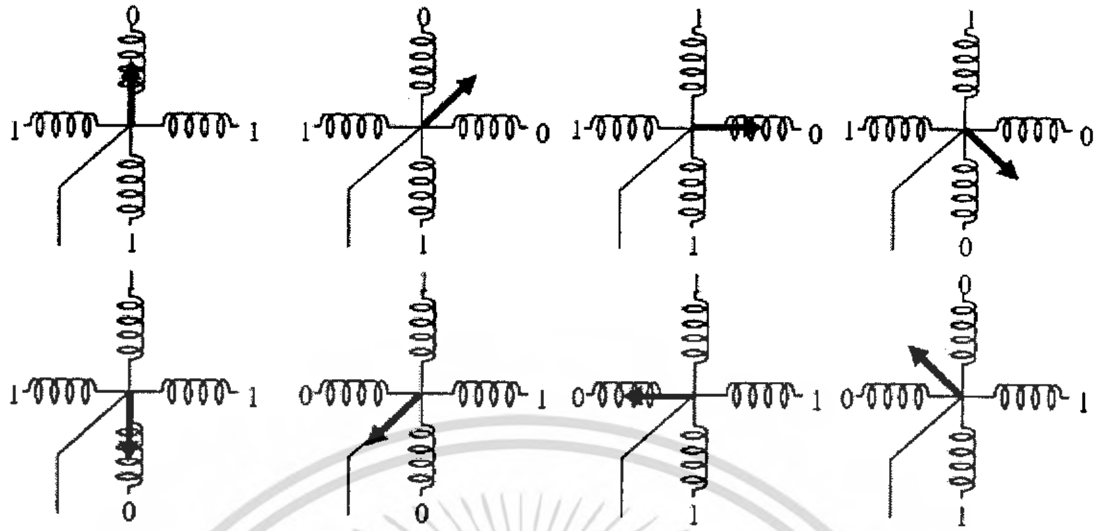
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบครึ่งสเต็ปเป็นรูปแบบที่ผสมผสานระหว่างการกระตุ้นแบบหนึ่งเฟสและแบบ 2 เฟส เพื่อเพิ่มจำนวนของสเต็ปต่อรอบอีกเท่าตัวหนึ่ง ในระบบนี้จะทำการกระตุ้นขดลวดเรียงกันไปเป็นลำดับ ดังนี้ เริ่มจากขดลวดที่ 1,1 และ 2,2,2 และ 3,3,3 และ 4,4,4 และ 1 แล้ววนกลับมายังขดลวดที่ 1 แรงบิดที่ได้จากการกระตุ้นแบบนี้จะเพิ่มมากขึ้นอีก เพราะช่วงสเต็ปมีระยะสั้นลง แต่ละสเต็ปเกิดแรงเกิดแรงดึงจากขดลวด 2 ขดที่ถูกกระตุ้นพร้อมกัน ความถูกต้องของตำแหน่งมีเพิ่มมากขึ้น แต่ต้องพึงระวังไว้อีกประการหนึ่งว่า เมื่อกระตุ้นให้ทำงานในรูปแบบนี้จะต้องทำการหมุนถึง 2 สเต็ป จึงจะได้เท่ากับระยะเท่ากับ 1 สเต็ปเต็มของการควบคุมใน 2 แบบแรก สำหรับแหล่งจ่ายกำลังไฟฟ้าต้องใช้ขนาดเท่ากับแบบ 2 เฟสเป็นอย่างน้อย จึงจะเพียงพอ ขั้นตอนการทำงานต่างๆ แสดงดังในตารางที่ 2.6

ตารางที่ 2.6 แสดงลำดับการทำงานของขดลวดในแต่ละเฟสของมอเตอร์ เมื่อได้รับการกระตุ้นแบบครึ่งสเต็ป

สเต็ปที่	เฟสที่ 1	เฟสที่ 2	เฟสที่ 3	เฟสที่ 4
1	ทำงาน	-	-	-
2	ทำงาน	ทำงาน	-	-
3	-	ทำงาน	-	-
4	-	ทำงาน	ทำงาน	-
5	-	-	ทำงาน	-
6	-	-	ทำงาน	ทำงาน
7	-	-	-	ทำงาน
8	ทำงาน	-	-	ทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.20 แสดงการหมุนของแกนมอเตอร์เมื่อได้รับการกระตุ้นแบบครึ่งสเต็ป

ในการใช้ไมโครคอนโทรลเลอร์ในการควบคุม Stepping Motor เราต้องใช้ภาคขับที่สามารถจ่ายกระแสได้เพียงพอที่ทำให้ Stepping Motor หมุนได้ ซึ่งเราอาจใช้ไอซีแบบ Open Collector เช่น ULN2003

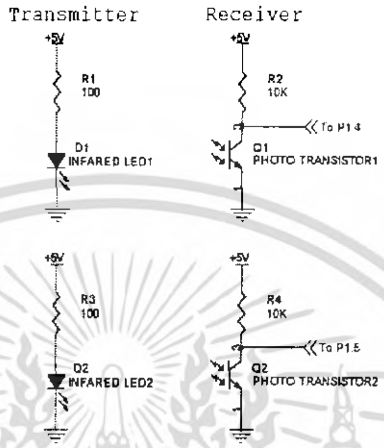
2.8 อินฟราเรดเซ็นเซอร์ Infrared sensor)

การตรวจวัดระยะทางโดยใช้คลื่นอินฟราเรดนั้นจัดว่าเป็นการใช้อุปกรณ์ที่อยู่ในกลุ่มประเภทของอุปกรณ์ตรวจจับทางแสง (Photo Sensors)โดยใช้หลักการหักเหของแสงที่ตกกระทบ กับแท่งแก้วผลึก ในที่นี้จะมีอินฟราเรดเป็นตัวกำเนิดคลื่นอินฟราเรด โดยมีโฟโต้ทรานส์ซิสเตอร์ (Photo Transistor) เป็นอุปกรณ์ตัวรับ (Receiver) คลื่นอินฟราเรดที่สะท้อนมาจากแก้วผลึก การทำงานของระบบขณะที่วัดระยะทางการเคลื่อนที่ของรถ คลื่นอินฟราเรดจะเดินทางไปกระทบแถบขาว-ดำตรงล้อของรถก็จะสะท้อนคลื่นกับมายังโฟโต้ทรานส์ซิสเตอร์ซึ่งทำหน้าที่เป็นตัวรับคลื่น โดยที่แถบขาวจะสะท้อนคลื่นกลับมาส่วนแถบดำก็จะดูดกลืนแสงทำให้ไม่มีการส่งค่ากลับมาทำให้เราสามารถวัดระยะการเคลื่อนที่ของล้อโดยวัดจากจำนวนที่เกิดการสะท้อนกลับของคลื่นอินฟราเรดในการวิ่งของรถแต่ละครั้ง

โฟโต้ทรานซิสเตอร์ (Photo Transistor) เป็นอุปกรณ์ทรานสดิวเซอร์ชนิดสารกึ่งตัวนำแบบเฉื่อยงานอีกชนิดที่นิยมใช้กันอย่างแพร่หลายโดยอาศัยหลักการทำงานเช่นเดียวกับโฟโต้ไดโอดแต่จะดีกว่าตรงกระแสที่ไหลผ่านในตัวเองสูงกว่าโฟโตไดโอดสำหรับการนำเอาโฟโต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทรานส์ซิสเตอร์ไปใช้งานนั้นจะต้องต่อวงจรในลักษณะไบอัสกลับที่รอยต่อระหว่างขา Base กับขา Collector เมื่อคลื่นอินฟราเรดมาตกกระทบบริเวณนี้จะเกิดกระแสเบสที่เกิดจากอิเล็กตรอนไหลเข้าสู่ทรานซิสเตอร์และกระแสเบสนี้จะถูกขยายด้วยอัตราการขยายกระแสของทรานซิสเตอร์



รูปที่ 2.21 วงจรอินฟราเรดเซ็นเซอร์

จากวงจรโฟโตทรานซิสเตอร์

ในขณะที่ไม่มีแสงมาตกระทบโฟโตทรานซิสเตอร์

$$I_C = 0, V_{RL} = I_C R_L = 0\ V$$

จาก

$$E = V_{RL} + V_{CE}$$

$$5 = 0\ V + V_{CE}$$

$$V_{CE} = 5\ V$$

ในขณะที่มีแสงมาตกระทบโฟโตทรานซิสเตอร์

$$V_{CE} = 0\ V, V_{RL} = E = 5\ V$$

$$I_C = \frac{V_{RL}}{R_L} = \frac{5V}{10K\Omega}$$

$$I_C = 0.5\ mA$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9 Ds 1820 1-Wire Thermometer

2.9.1 หลักการเบื้องต้นของไอซี DS1820

ไอซี DS1820 เป็นไอซีที่มีระบบการสื่อสารข้อมูลอนุกรมแบบหนึ่งสายซึ่งถือได้ว่าเป็นระบบที่มีความชาญฉลาด และใช้จำนวนสายสัญญาณเพียง 1 เส้นเท่านั้น โดยไม่ต้องมีสายสัญญานนาฬิกามาควบคุมจังหวะการถ่ายทอดข้อมูลเหมือนกับระบบสื่อสารข้อมูลอนุกรมในแบบอื่น สายข้อมูลจะทำหน้าที่เสมือนเป็นสายสัญญาณนาฬิกาในตัว ส่วนค่าของข้อมูลจะพิจารณาจากลักษณะของรูปสัญญาณที่ปรากฏบนสายสัญญาณในแต่ละช่องของเวลาซึ่งเรียกว่า ไทม์สล็อต (Time Slot) โดยคาบเวลาต่ำสุดและสูงสุดของสถานะต่าง ๆ ในการสื่อสารข้อมูลในแต่ละไทม์สล็อตมีการกำหนดขอบเขตไว้อย่างชัดเจนการถ่ายทอดข้อมูลจะเกิดขึ้นในแต่ละไทม์สล็อคนั้นรูปแบบการถ่ายทอด ข้อมูลจะเป็นแบบอะซิงโครนัสในระดับบิด ไม่มีการกำหนดความยาวของข้อมูลเป็นระดับไบต์ระบบสื่อสารแบบนี้เหมาะที่จะใช้ในการสื่อสารข้อมูลระหว่างไอซีแผงวงจรเดียวกัน

2.9.2 การอินเตอร์เฟสผ่านสายเส้นเดียว

การเชื่อมต่อหรือการอินเตอร์เฟส (Interface) ระหว่างไมโครคอนโทรลเลอร์กับอุปกรณ์ภายนอก โดยใช้จำนวนสายสัญญาณให้น้อยที่สุดได้มีการพัฒนามาอย่างต่อเนื่องจากหลายบริษัทผู้ผลิต เช่น การเชื่อมต่ออุปกรณ์ต่อพ่วงแบบอนุกรม (Serial Peripheral Interface, SPI) ในไมโคร-คอนโทรลเลอร์ 68HC1 ของโมโตโรลา การเชื่อมต่อแบบ SPI นี้ช่วยให้ ไมโครคอนโทรลเลอร์แลกเปลี่ยนข้อมูลกับอุปกรณ์ต่อพ่วงได้ด้วยความเร็วถึง 1 ล้านบิตต่อวินาที โดยใช้สายรับส่งสัญญาณเพียง 3 หรือ 4 เส้น รวมกับสายกราวด์อีกหนึ่งเส้น

2.9.3 คุณสมบัติ

3.1) DS1820 สามารถ Interface โดยใช้สายสัญญาณเพียงเส้นเดียว

3.2) DS1820 เพียงตัวเดียว สามารถวัดอุณหภูมิได้โดยไม่ต้องต่ออุปกรณ์ร่วม

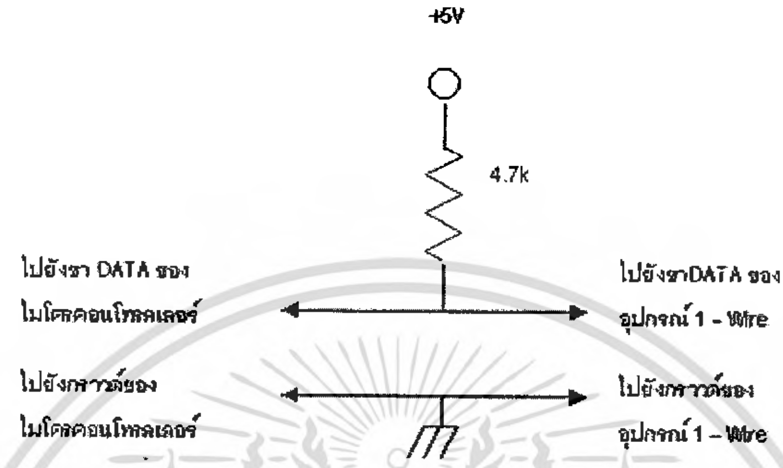
3.3) DS1820 มีย่านวัดอยู่ที่ +125 ถึง -55 C

3.4) DS1820 มีความละเอียดในการวัดได้ 0.5 C

อุปกรณ์ที่สนับสนุนระบบบัสเพียงเส้นเดียวจะมีสายสัญญาณเพียง 2 เส้นเท่านั้นคือสายกราวด์และสายสัญญาณ ซึ่งเรียกอีกอย่างว่า สาย DATA สายนี้จะจัดการเกี่ยวกับทั้งสัญญาณข้อมูลและสัญญาณนาฬิกาที่ใช้ในการแลกเปลี่ยนข้อมูล สาย DATA นี้จะเป็นชนิด Open Drain ดังนั้นใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบวงจรจะต้องออกแบบให้มีตัวต้านทานมาพูลอัพสาย DATA นี้ด้วยให้ดูรูปแผนผังแสดงการต่อระบบบัสของ 1 – Wire Bus ดังรูป

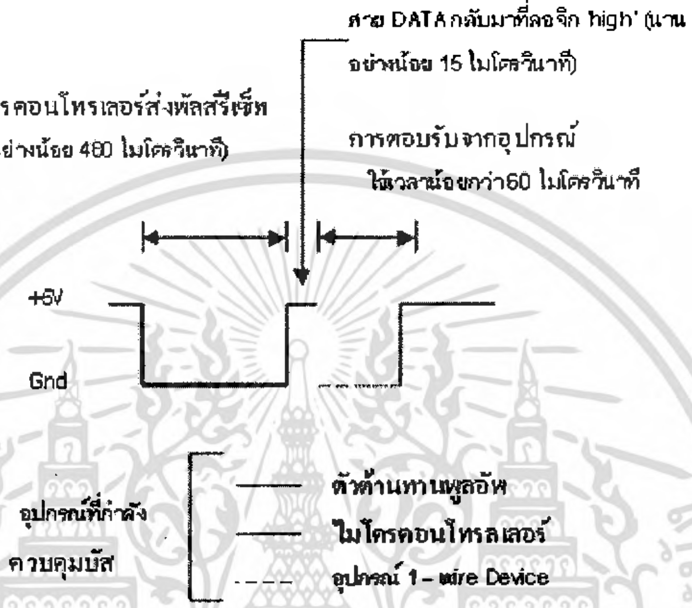


รูปที่ 2.22 แผนผังของระบบบัสแบบ 1- Wire Bus

จากการแลกเปลี่ยนข้อมูลระหว่างไมโครคอนโทรลเลอร์กับอุปกรณ์ที่ใช้ 1 – Wire Bus นี้ไม่ได้ง่าย ๆ เหมือนกับการส่งข้อมูลผ่านทางบัสแบบ SPI เพราะในระบบ 1 – Wire Bus นั้นสาย DATA จะต้องจัดการเกี่ยวกับจังหวะเวลา (Timing) ระดับสัญญาณ (Level) และทิศทาง (Direction)ของข้อมูลทั้งหมด ทำให้การเขียนซอฟท์แวร์ของไมโครคอนโทรลเลอร์ที่ติดต่อกับอุปกรณ์พวกนี้ต้องมีความซับซ้อนมากขึ้น ในสภาวะพัก (Quiescent State) อุปกรณ์ที่ใช้บัสแบบ 1 – Wire Bus จะทำให้สาย DATA อยู่ในสภาวะลอย (float) ทำให้ขานี้มีแรงดันเท่ากับแรงดันพูลอัพ (Vcc) ซึ่งปกติก็คือ 5 โวลต์นั่นเอง ส่วนไมโครคอนโทรลเลอร์ก็จะปล่อยให้ขาเอาต์พุตที่ใช้ติดต่อกับขา DATA นี้อยู่ในสภาวะ ความต้านทานสูง (High-Impedance State) เช่นกัน เมื่ออุปกรณ์ทั้งสองชนิดนี้ปล่อยให้ขา DATA อยู่ในสภาวะลอยแล้วนี้ความต้านทานพูลอัพก็จะช่วยรักษาระดับแรงดันไฟเลี้ยงที่จ่ายให้กับอุปกรณ์ที่ใช้บัสแบบ 1 – Wire Bus นี้ได้อย่างสม่ำเสมอเพราะอุปกรณ์ที่ใช้บัสแบบ 1 – Wire Bus นี้จะใช้พลังงานในการทำงานน้อยมาก ในการแลกเปลี่ยนข้อมูลกันไมโครคอนโทรลเลอร์และอุปกรณ์ที่ใช้บัสแบบ 1 – Wire Bus จะต้องดำเนินการอย่างระมัดระวังตามลำดับขั้นตอนในการทำให้สาย DATA มีลอจิกเป็น 'low' ปล่อยให้สาย DATA กลับมามีลอจิกเป็น 'high' และตรวจจับการตอบรับกับอุปกรณ์อีกด้านหนึ่ง ช่วงจังหวะเวลาที่ใช้ในกระบวนการนี้จะถูกกำหนดโดยข้อกำหนดเฉพาะของระบบ 1 – Wire Bus นี้ซึ่งต้องใช้ไมโครคอนโทรลเลอร์ที่มีการตอบสนองได้อย่างรวดเร็วด้วย ดังนั้นเราจะต้องตรวจสอบความสามารถของระบบให้ดีเสีย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก่อนที่จะใช้ระบบบัสแบบ 1 – Wire Bus นี้ จากสภาวะพักข้างต้นการแลกเปลี่ยนข้อมูลจะเริ่มขึ้นด้วยการที่ไมโครคอนโทรลเลอร์กระทำกระบวนการรีเซ็ต (Reset Sequence) ซึ่งทำได้โดยการทำให้สาย DATA มีลอจิก 'low' เป็นเวลาอย่างน้อย 480 ไมโครวินาที แล้วจึงปล่อยให้กลับมาอยู่ในสภาวะ 'high' อีกครั้งหนึ่ง ดังรูป

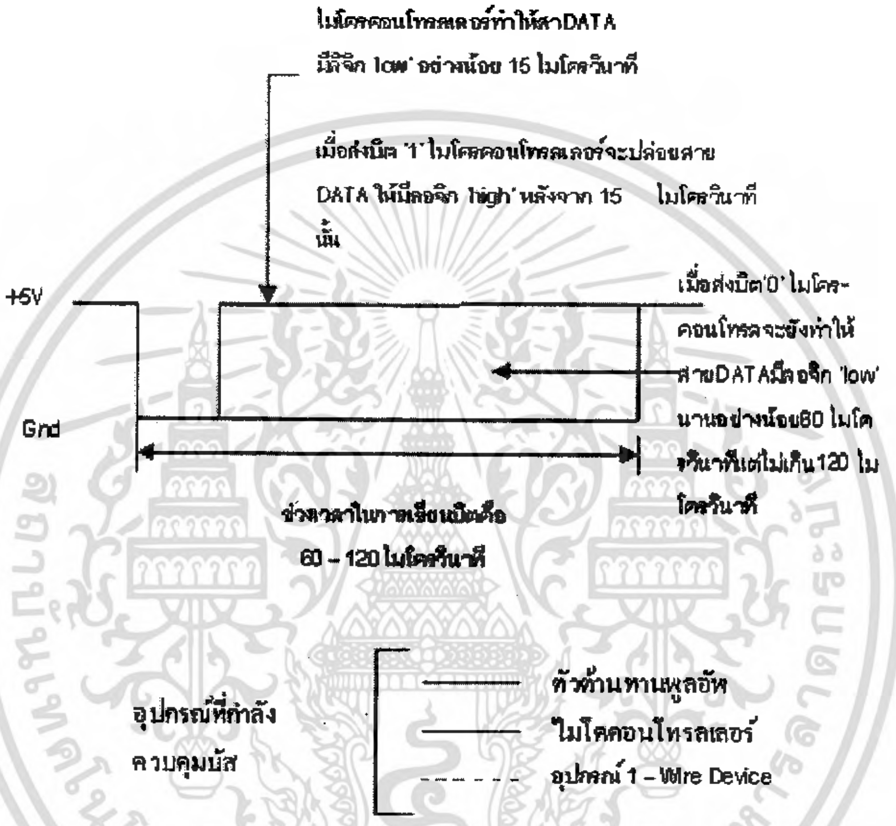


รูปที่ 2.23 จังหวะเวลาในการทำกระบวนการตรวจสอบว่ามีอุปกรณ์อยู่บนบัส

เมื่ออุปกรณ์ที่ใช้บัสแบบ 1 – Wire Bus นี้ตรวจพบสภาวะ RESET นี้ มันก็จะตอบสนองด้วยการส่งพัลส์กลับไปเพื่อบอกให้ไมโครคอนโทรลเลอร์รู้ว่าบนสายบัสนี้มีอุปกรณ์แบบ 1 – Wire Device กำลังทำงานอยู่ โดย อุปกรณ์แบบ 1 – Wire Devicc จะปล่อยให้สาย DATA อยู่ในลอจิก 'high' อย่างน้อย 15 ไมโครวินาที แต่ไม่เกิน 60 ไมโครวินาที จากนั้นมันก็จะให้สาย DATA ลงมามีลอจิกเป็น 'low' ในช่วงเวลาประมาณ 60-240 ไมโครวินาที แล้วจึงปล่อยให้กลับไปมีลอจิก 'high'เช่นเดิม ช่วงเวลานี้มีชื่อเรียกกันหลายชื่อเช่น ช่วงเวลาเริ่มติดต่อ (Initialization) หลังจาก อุปกรณ์ 1 – Wire Device ปล่อยให้สาย DATA กลับมาอยู่ในลอจิก 'high' แล้ว ไมโครคอรโทรลเลอร์จะต้องปล่อยให้สาย DATA อยู่ในลอจิกนี้นานอย่างน้อย 240 ไมโครวินาที ต่อไปจากนั้น ไมโครคอนโทรลเลอร์ก็จะส่งคำสั่งเริ่มต้น (Initial Command) ขนาด 1 ไบต์ไปยังอุปกรณ์ 1 – Wire Bus ซึ่งคำสั่งนี้อาจจะเป็นคำสั่งอะไรก็ได้ ไมโครคอนโทรลเลอร์จะส่งบิตของ คำสั่งนั้นออกไป โดยการเปลี่ยนสถานะของสาย DATA กลับไปกลับมาโดยตอนแรกจะให้เป็น ลอจิก 'low' แล้วจึง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

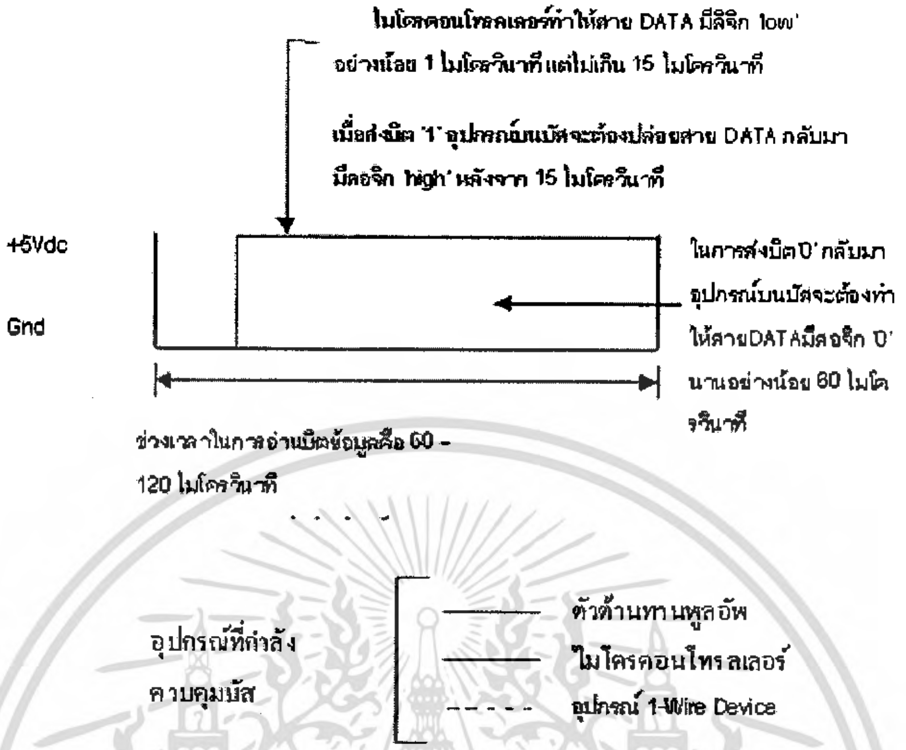
กลับมาเป็น 'high' ตามช่วงจังหวะเวลาที่เหมาะสม ช่วงเวลานานที่ ไมโครคอนโทรลเลอร์ทำให้สาย DATA มีลอจิก 'low' จะเป็นตัวแยกแยะว่าบิตไหนที่มีลอจิกเป็น '1' บิตไหนมีลอจิกเป็น '0' ให้ดูแผนภูมิเวลาในการเขียนบิต '1' หรือ '0' ดังรูป



รูปที่ 2.24 จังหวะเวลาที่ไมโครคอนโทรลเลอร์ใช้เขียนข้อมูล '0' หรือ '1' ไปยังอุปกรณ์บนบัส

ช่วงเวลาระหว่าง 15-120 ไมโครวินาที หลังไมโครคอนโทรลเลอร์ทำให้สาย DATA มีลอจิก 'low' ในตอนนี้จะเรียกว่าช่วงการอ่านสถานะบิตข้อมูล (Sampling Window)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



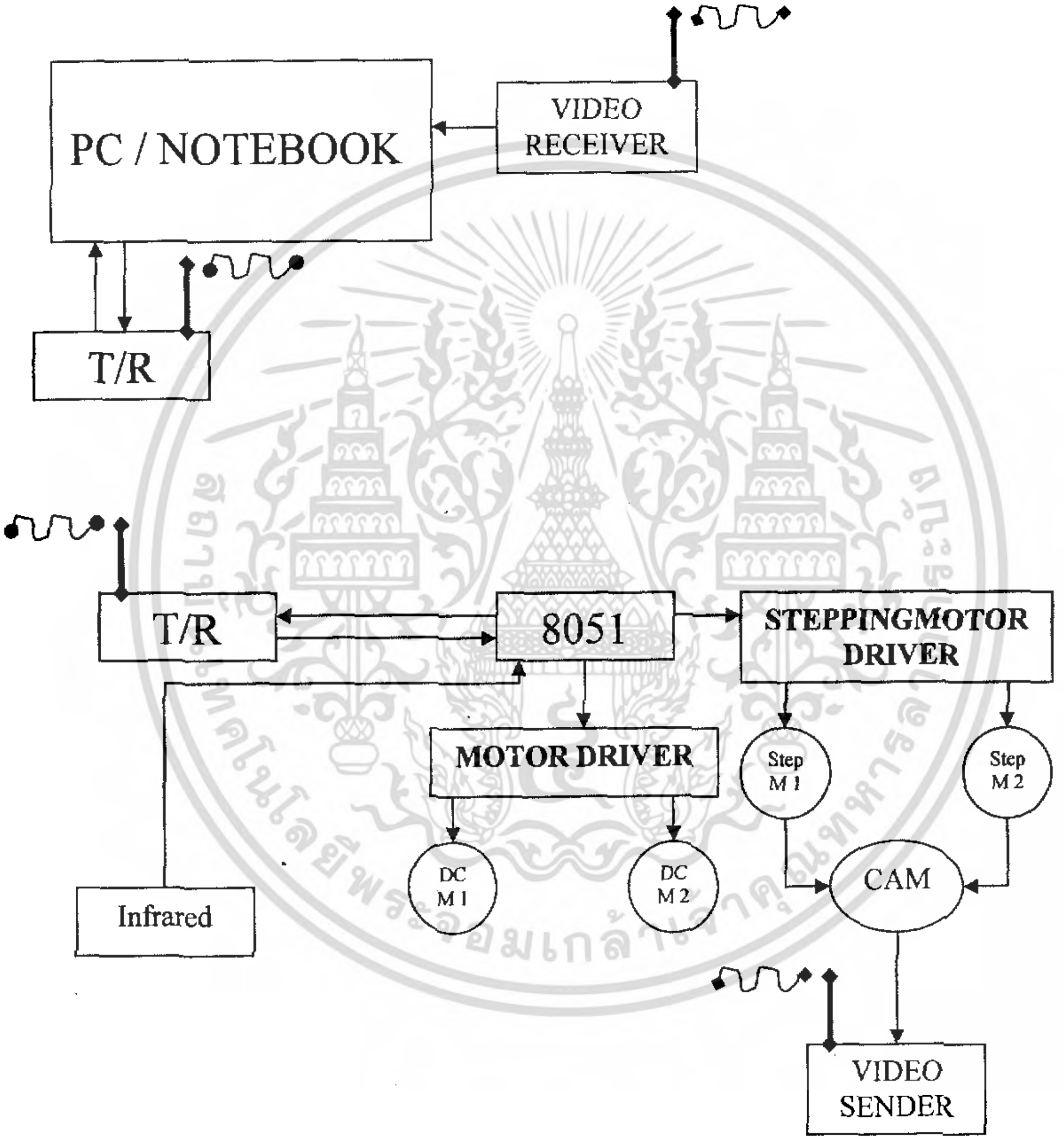
รูปที่ 2.25 จังหวะเวลาที่ไมโครคอนโทรลเลอร์ใช้อ่านบิตข้อมูล '0' หรือ '1' จากอุปกรณ์บนบัส

ในการอ่านบิตข้อมูลจากบัสแบบ 1-Wire Bus ตอนแรกไมโครคอนโทรลเลอร์จะต้องทำให้สาย DATA มีลอจิก 'low' เป็นเวลานานไม่เกิน 15 ไมโครวินาทีแล้วจึงปล่อยให้สาย DATA กลับมามีลอจิก 'high' เช่นเดิมจากนั้นอุปกรณ์ 1 – Wire Device ก็จะเข้าควบคุมสาย DATA แทนโดยจะส่งบิต '0' โดยการทำให้สาย DATA มีลอจิกเป็น 'low' และส่งบิต '1' โดยการปล่อยให้สาย DATA กลับมามีลอจิก 'high' ตามเดิม เมื่อส่งข้อมูลออกไปยังไมโครคอนโทรลเลอร์เรียบร้อยแล้ว จะมีการพักอยู่ชั่วขณะหนึ่ง จากนั้นอุปกรณ์ 1 – Wire Device จะปล่อยการควบคุมจากสาย DATA ให้เป็นอิสระ และรอรับคำสั่งการอ่านของข้อมูลครั้งต่อไป ถ้าอุปกรณ์ 1 – Wire Device ส่งบิต '0' ออกไป ไมโครคอนโทรลเลอร์ก็จะสามารถตรวจสอบจุดสิ้นสุดของลอจิก '0' นี้ง่าย เพราะสาย DATA จะกลับมาอยู่ที่ลอจิก 'high' ตามเดิม แต่ถ้าตรวจสอบจุดสิ้นสุดของการส่งบิต '1' ของอุปกรณ์ 1 – Wire Device จะต้องใช้เทคนิคมากกว่านี้ เพราะสาย DATA จะอยู่ที่ลอจิก 'high' อยู่แล้วนี่ก็เป็นเหตุผลว่าทำไมจังหวะเวลาในการอ่านเขียนข้อมูลจึงเป็นเรื่องที่สำคัญมาก เมื่อจะอ่านบิต '1' จากอุปกรณ์ 1 – Wire Device ไมโครคอนโทรลเลอร์จะต้องทำตามช่วงเวลาที่แสดงในแผนภูมิเวลาอย่างเคร่งครัด และต้องไม่ทำการอ่านลอจิกของบิตถัดมา จนกว่าจะผ่านเวลาไปแล้วอย่างน้อย 60 ไมโครวินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

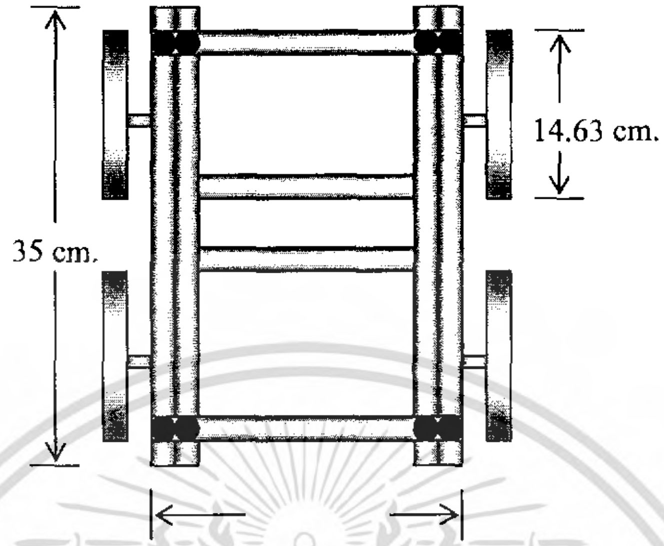
บทที่ 3
วิธีดำเนินการวิจัย

3.1 ส่วนการออกแบบตัวรถและระบบควบคุม



รูปที่ 3.1 บล็อกไดอะแกรมของรถสำรวจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



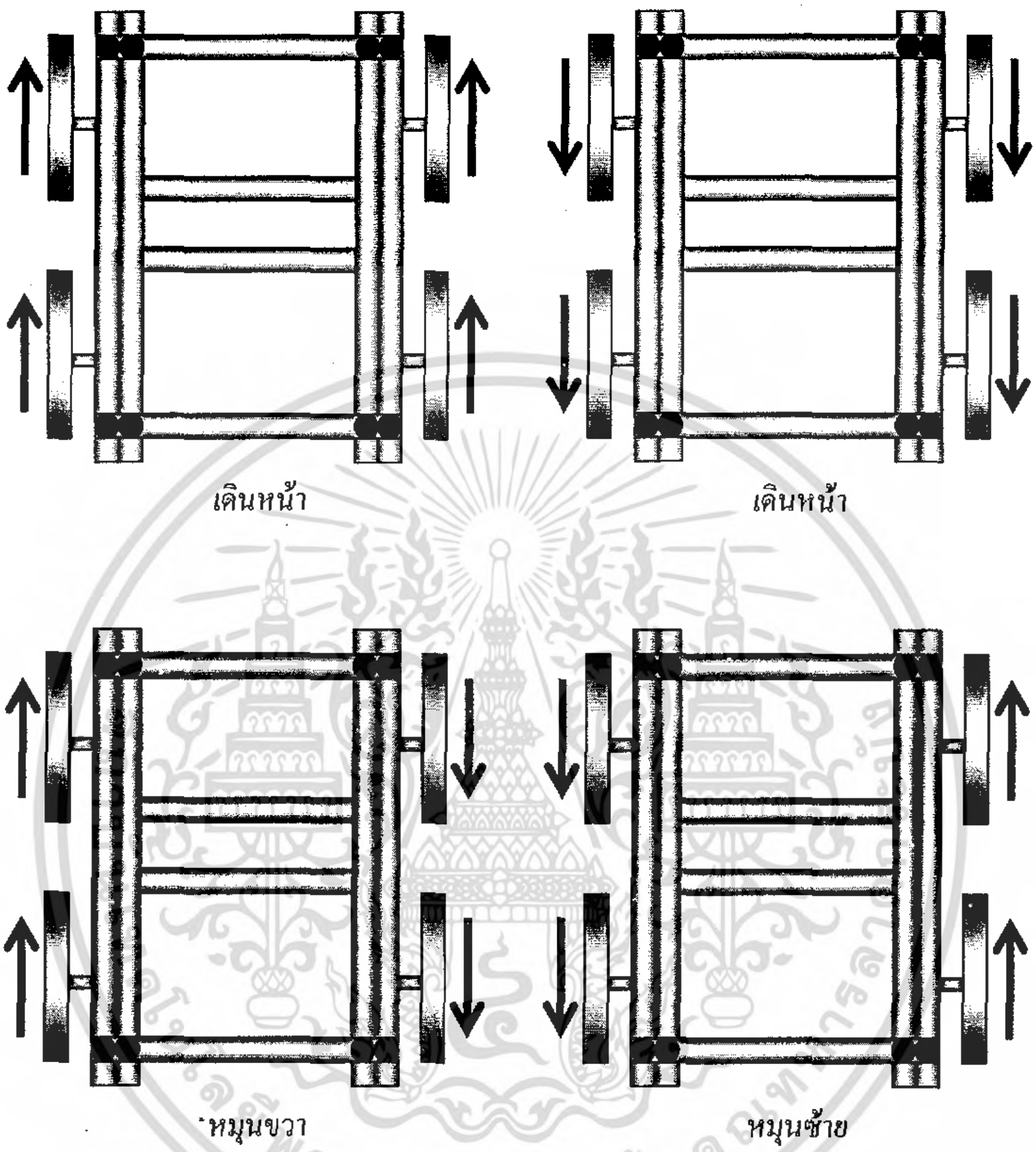
รูปที่ 3.2 ภาพแสดงตัวอย่างโครงสร้างของตัวรถ

โครงสร้างของตัวรถ เป็นตัวรถที่สามารถเคลื่อนที่สำรวจเส้นทางและจุดต่างๆรวมถึงข้ามสิ่งกีดขวางที่เป็นอุปสรรคสามารถควบคุมแบบไร้สายจากคอมพิวเตอร์ที่อยู่ไกลออกไป โครงรถ ใช้อลูมิเนียมเป็นโครงสร้างหลัก ล้อแบบพลาสติกสังเคราะห์ และมอเตอร์ไฟตรง 12 V 100 RPM ในการขับเคลื่อน

<u>ข้อมูลของโครงสร้างรถ</u>

ขนาดของล้อ	1.30 cm
เส้นผ่าศูนย์กลางล้อ	14.63 cm
ระยะห่างระหว่างล้อหน้าและล้อหลัง	20.00 cm
ขนาดของตัวรถ	
- กว้าง	22.00 cm
- ยาว	35.00 cm
ขนาดของตัวรถ (รวมล้อ)	
- กว้าง	32.00 cm
- ยาว	35.00 cm
ความสูงจากพื้นถึงรถ	11.00 cm
จากพื้นถึงเสาอากาศกล้อง (ส่วนที่สูงที่สุดของตัวรถ)	36.00 cm
น้ำหนักของรถ	6.10 Kg

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 ภาพแสดงตัวอย่างการเคลื่อนที่ของรถสำรวจ

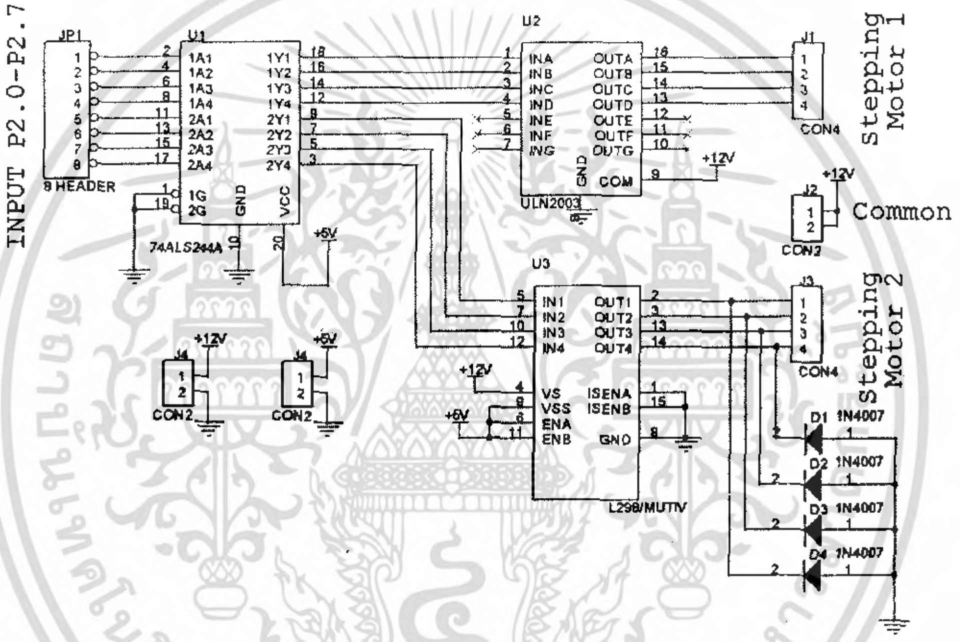
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1 ส่วนการรับ-ส่งสัญญาณควบคุมจากคอมพิวเตอร์

ในส่วนการควบคุมไร้สายนี้จะใช้สัญญาณคลื่นวิทยุเป็นสัญญาณควบคุมโดยอาจจะใช้โมดูลในการติดต่อสื่อสารในส่วนนี้ ซึ่งโมดูลที่ใช้นั้นใช้โมดูลแบบ GFSK ความถี่ 2.4 GHz

3.1.2 ส่วนควบคุมการหมุนของกล้อง

ส่วนนี้จะใช้ Stepping Motor 2 ตัวในการหมุนของกล้อง โดยตัวแรกจะทำการหมุนกล้องในแนวระดับ ส่วนอีกตัวหนึ่งจะช่วยหมุนกล้องในแนวดิ่ง



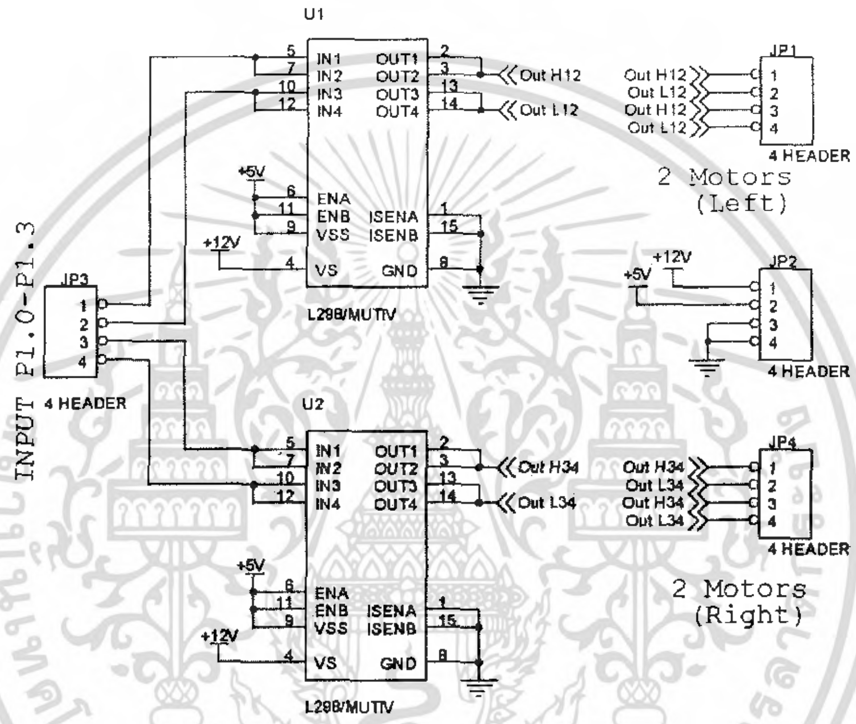
รูปที่ 3.4 แสดงวงจรขับ Stepping Motor โดยใช้ไอซี Driver เบอร์ ULN2003และ L298MUTIV

โดยที่สเตปปิ้งมอเตอร์ตัวเล็กใช้หมุนในแนวดิ่งเนื่องจากสเตปปิ้งมอเตอร์ชนิดนี้ได้มีการทดเกียร์เอาไว้ทำให้ได้ทอร์กมากขึ้นความเร็วรอบน้อยเมื่อเทียบกับสเตปปิ้งมอเตอร์แบบไม่ทดเกียร์จึงทำให้สามารถหมุนในแนวดิ่งได้ดี ส่วนสเตปปิ้งมอเตอร์ตัวที่ใช้ในการหมุนในแนวระดับนั้นใช้เป็นฐานในการหมุนของกล้องต้องรับภาระของอุปกรณ์ต่างที่ติดกับตัวกล้องจึงต้องใช้สเตปปิ้งมอเตอร์ที่มีขนาดในการขับเคลื่อนมากสมควร และตัวกล้องด้านบนก็จะมีLED Superbright ติดไว้เพื่อส่องแสงใช้ในการจับภาพบริเวณที่มืด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3 ส่วนการขับมอเตอร์ไฟตรง

ในส่วนมอเตอร์ไฟตรงนี้จะใช้ในการขับเคลื่อนของตัวรถซึ่งในการขับมอเตอร์ไฟตรงที่สามารถหมุนได้ทั้ง CW และ CCW เราต้องใช้วงจร H-Bridge ดังที่กล่าวไว้ในบทที่ 2 แต่เพื่อความสะดวกจึงควรใช้ไอซี L298 ซึ่งเป็น Full Bridge Driver ในการขับ DC-Motor ส่วนการปรับความเร็วรอบเราจะใช้เทคนิค PWM ซึ่งกล่าวไว้ในบทที่ 2 แล้วเช่นกัน



รูปที่ 3.5วงจรขับมอเตอร์ไฟตรง

3.1.4 ส่วนไมโครคอนโทรลเลอร์ 8051

ส่วนไมโครคอนโทรลเลอร์มีหน้าที่ดังนี้

- รับคำสั่งจาก 8051 (1) ผ่านส่วนรับสัญญาณวิทยุแล้วควบคุมอุปกรณ์ต่างๆตามที่ได้รับคำสั่ง
- รับข้อมูลจาก Sensor ต่างๆ เพื่อประมวลผลและตัดสินใจรวมทั้งส่งค่าที่ได้ไปยัง 8051 (1) ผ่านทางส่วนส่งสัญญาณวิทยุเพื่อส่งไปแสดงค่าทาง Computer

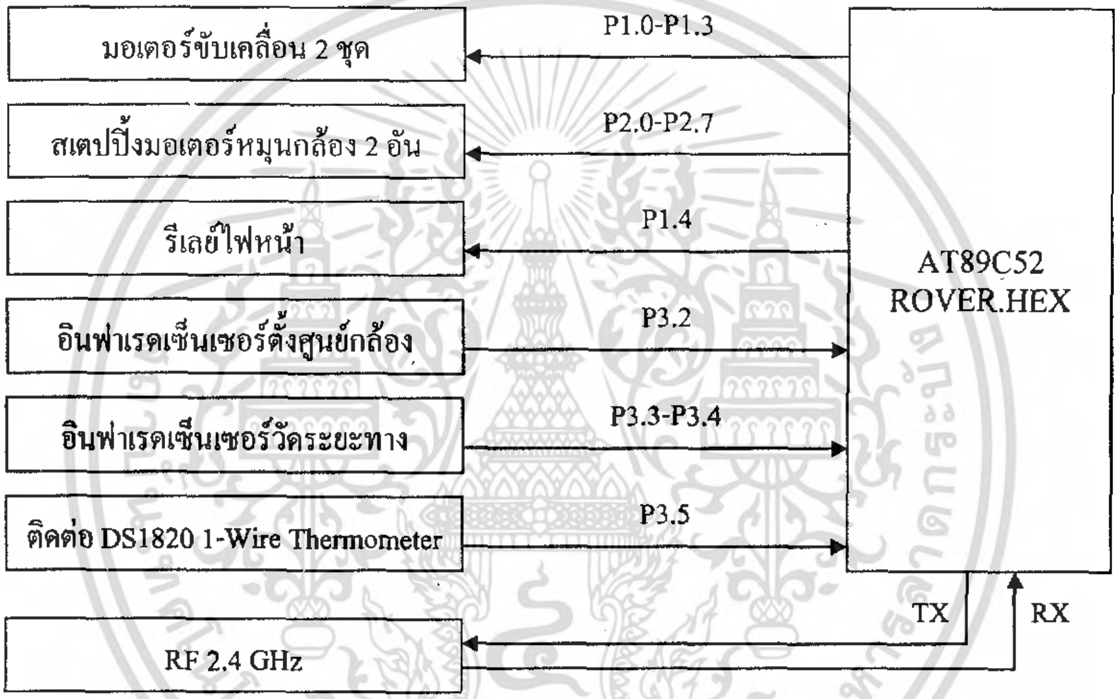
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.5 ส่วนกล้องและส่วนรับส่งสัญญาณภาพไร้สาย

ในส่วนนี้จะรับภาพจากการสำรวจของรถจากสภาพแวดล้อมภายนอกเพื่อส่งสัญญาณไร้สายไปแสดงผลยัง Computer

3.2 ตัวตรวจรู้

แสดงพอร์ทที่ควบคุมอุปกรณ์และรับค่าจากเซ็นเซอร์ต่างๆของไมโครคอนโทรลเลอร์

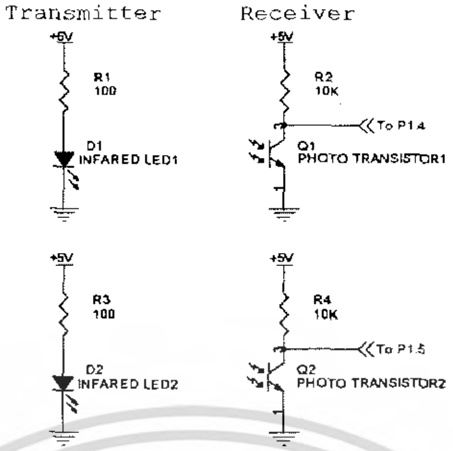


3.2.1 อินฟาเรดเซ็นเซอร์

- อินฟาเรดเซ็นเซอร์วัดระยะทาง

โดยคุณสมบัติของคลื่นอินฟราเรดนั้นสามารถดูดกลืนได้ดีในวัตถุสีดำและสะท้อนได้ดีในวัตถุที่เป็นสีขาว ดังนั้นเราจึงนำแถบดำมาติดที่ล้อที่เป็นสีขาว 8 แถบ เมื่อใช้วงจรตามรูปดังกล่าว เมื่อเซ็นเซอร์อยู่ที่แถบดำนั้นจะได้ output ของเซ็นเซอร์เป็น high และเมื่ออยู่ที่ส่วนล้อที่เป็นสีขาว เซ็นเซอร์จะให้ output เป็น low เมื่อเป็นเช่นนี้จึงสามารถวัดระยะทางที่รถเคลื่อนที่ไปได้โดย 1 รอบล้อ รถจะเคลื่อนที่ได้ 46 เซนติเมตร ดังนั้นความละเอียดในการวัดระยะทางคือ 46 cm. / 8 tickหรือ 5.75 cm./tick

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6วงจร อินฟราเรดเซ็นเซอร์

- อินฟาเรดเซ็นเซอร์ตั้งศูนย์กล้อง

ในการหมุนกล้องของรถสำรวจนั้นเมื่อกล้องหมุนไปซ้าย-ขวาเพื่อความสะดวกก่อนที่เราจะออกรถนั้นเราจึงมีระบบตั้งศูนย์กล้องเพื่อให้กล้องกลับมาอยู่ตรงกลาง โดยอัตโนมัติ โดยจะใช้อินฟราเรดเซ็นเซอร์ในการตั้งศูนย์ ซึ่งเราใช้ความสามารถในการหมุนที่แม่นยำของขับสเตปปิ้งมอเตอร์ในการตั้งศูนย์กลางของกล้อง โดยที่เราจะติดอินฟราเรดเซนเซอร์ไว้ที่ตำแหน่งที่ 90 องศา และติดแถบขาวไว้กลับฐานกล้องเมื่อกล้องหมุนมาโดนอินฟราเรดเซ็นเซอร์ ซึ่งเราใช้ความสัมพันธ์ของอินฟราเรดเซ็นเซอร์ที่มีคุณสมบัติในการสะท้อนแถบขาวได้ดี เมื่อกล้องหมุนมาตำแหน่งที่อินฟราเรดเซ็นเซอร์ติดอยู่ เราก็สั่งให้กล้องหมุนกลับมา90 องศาที่ตำแหน่งศูนย์กลางกล้องพอดี

3.2.3 DS1820 1-Wire Thermometer

ในส่วนของ DS1820 จะทำการวัดค่าอุณหภูมิที่ได้ โดยมีย่านวัดอยู่ที่ +125 ถึง -55 C มีความละเอียดในการวัดได้ 0.5 C จากนั้นก็จะส่งค่าที่ได้ไปแสดงผลยังคอมพิวเตอร์

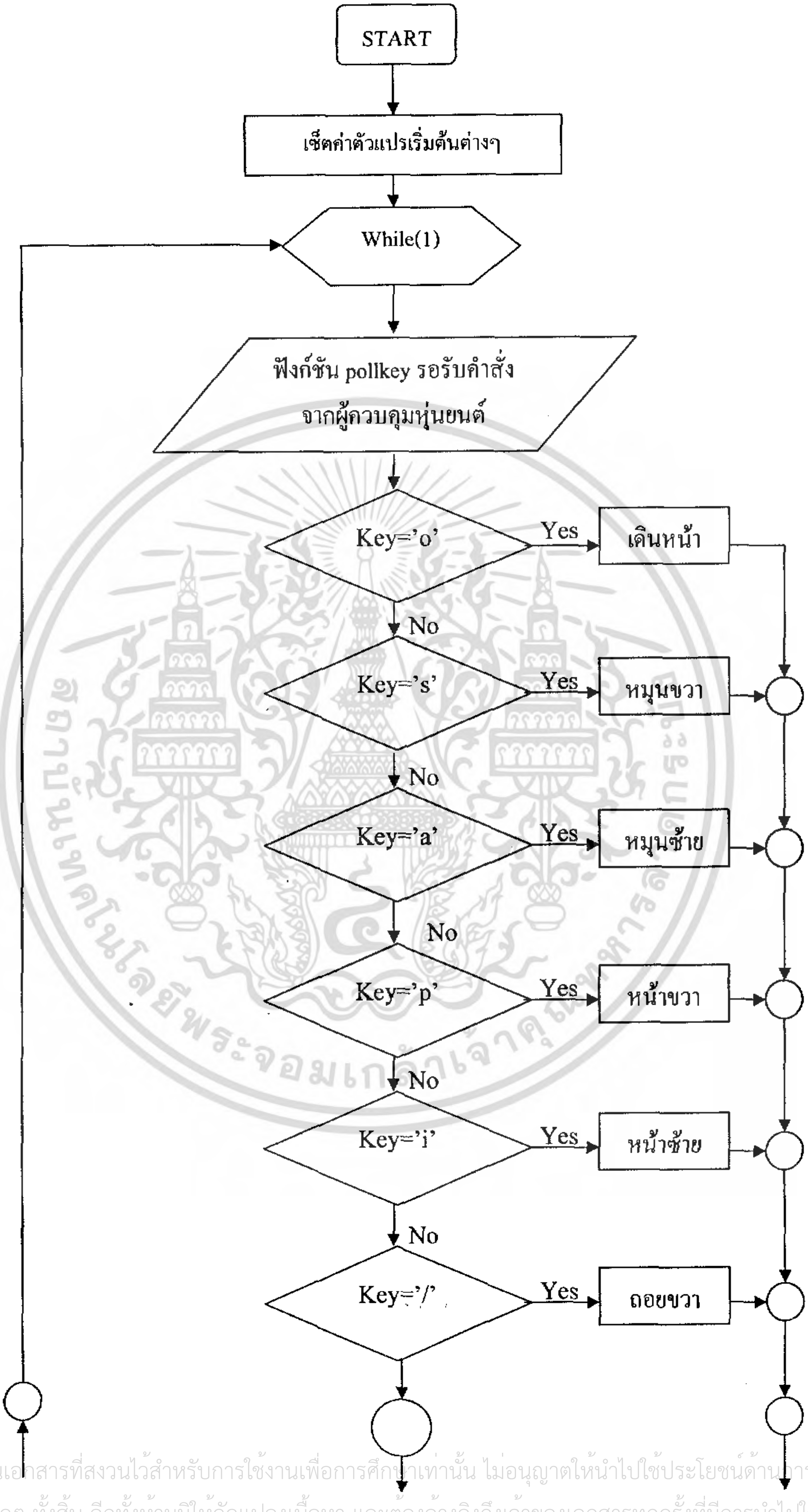
3.3 ส่วนซอฟต์แวร์ควบคุมการทำงานของรถ

3.3.1 ส่วนโปรแกรมควบคุมบน 8051 จะใช้โปรแกรม Micro C และภาษา Assembly

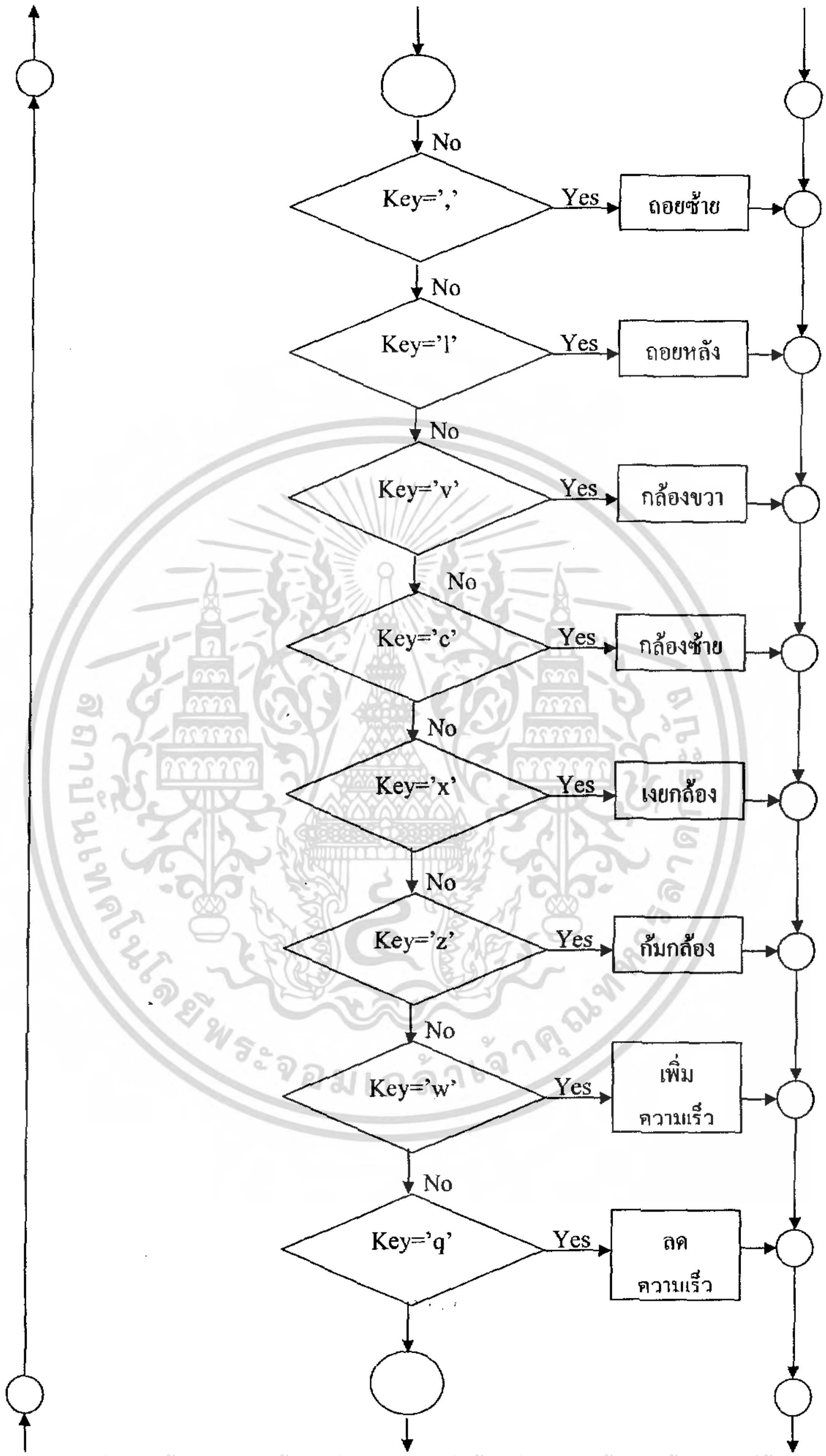
3.3.2 ส่วนโปรแกรมติดต่อกับผู้ใช้จะใช้ Visual Basic ในการสร้างโปรแกรม

3.3.1 ส่วนโปรแกรมควบคุมบน 8051 จะใช้โปรแกรม Micro C และภาษา Assembly

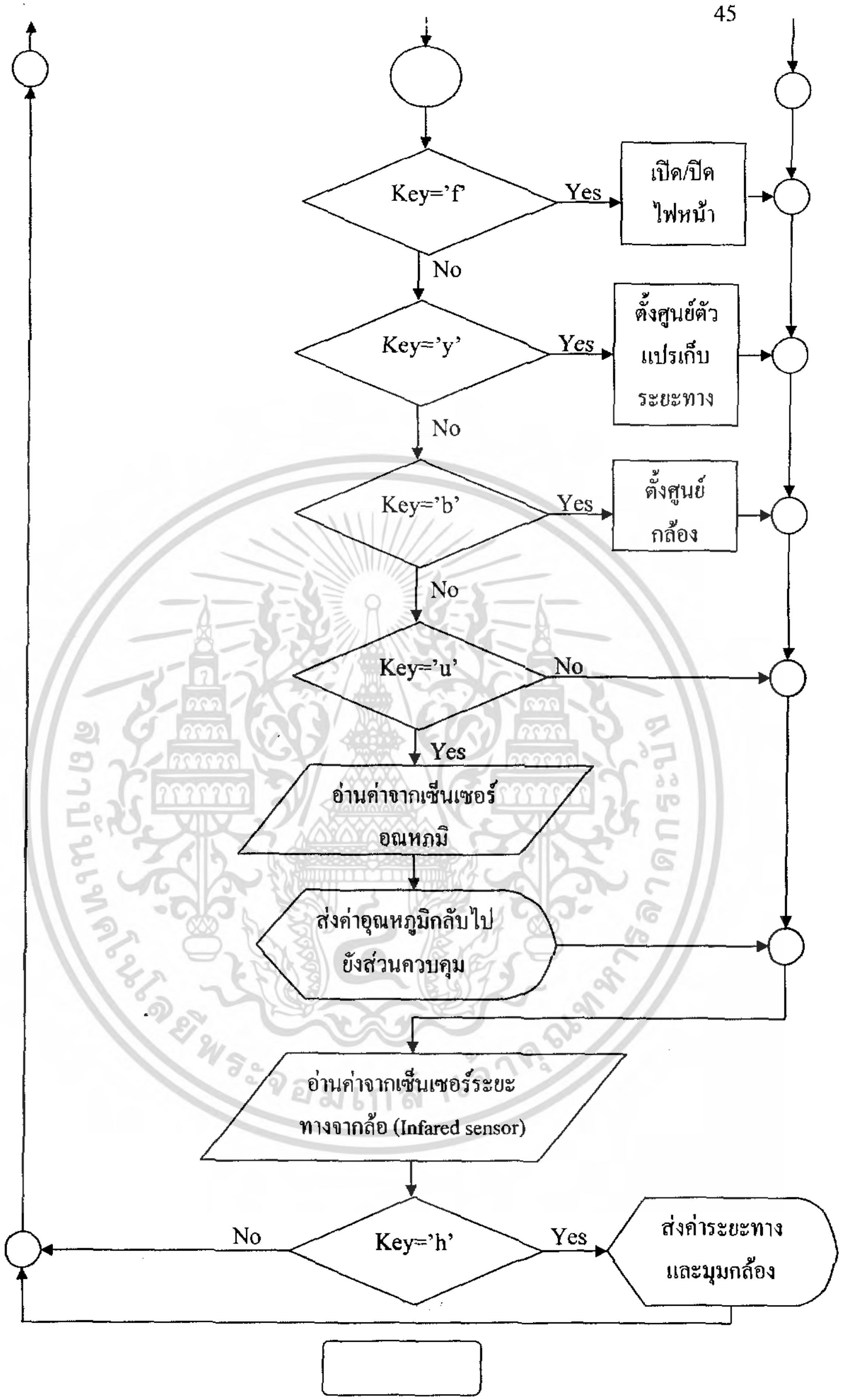
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 รูปแสดงแผนผังการส่วนควบคุมด้วยโปรแกรม Micro C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

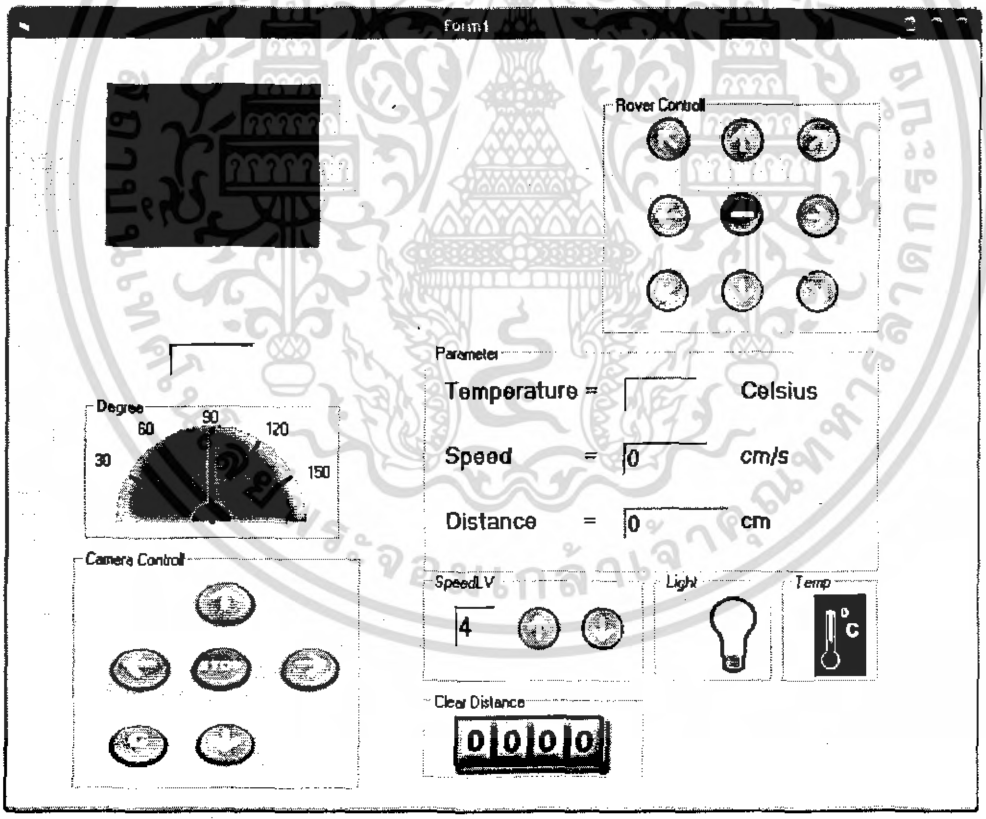
3.2.2 ส่วนโปรแกรมติดต่อกับผู้ใช้จะใช้ Visual Basic ในการสร้างโปรแกรม

โปรแกรมควบคุมการทำงานของรถสำรวจ

โปรแกรมควบคุมหุ่นยนต์ส่วนนี้เป็นส่วนที่ถูกใช้โดยผู้ควบคุมที่อยู่หน้าคอมพิวเตอร์โดยโปรแกรมนี้จะทำหน้าที่ติดต่อกับไมโครคอนโทรลเลอร์ผ่านทางพอร์ทอนุกรมและสัญญาณไร้สาย
เพื่อควบคุมการเคลื่อนที่ของหุ่นยนต์และการนำค่าข้อมูลต่างๆที่ไมโครคอนโทรลเลอร์ส่งมา นำมาแปลผลข้อมูลและควบคุมให้รถทำงานในฟังก์ชันต่างๆ

หลักการทำงานของโปรแกรม

ในส่วนของโปรแกรมจะมีส่วนต่างๆ ที่ ประกอบด้วย ภาพจากกล้องที่ติดอยู่กับรถสำรวจ ส่วนที่เป็นคอนโทรลในการบังคับทิศทางการเคลื่อนที่ของรถ และส่วนที่เป็นคอนโทรลในการบังคับทิศทางของกล้อง(หมุนกล้อง)



รูปที่ 3.8 โปรแกรมที่ใช้ในการควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปโปรแกรมที่ใช้ในการควบคุมจะตอบสนองได้ทั้งการ Click Mouse ที่ภาพหรือกดคีย์ Shot cut โดยมีค่าต่างๆดังนี้

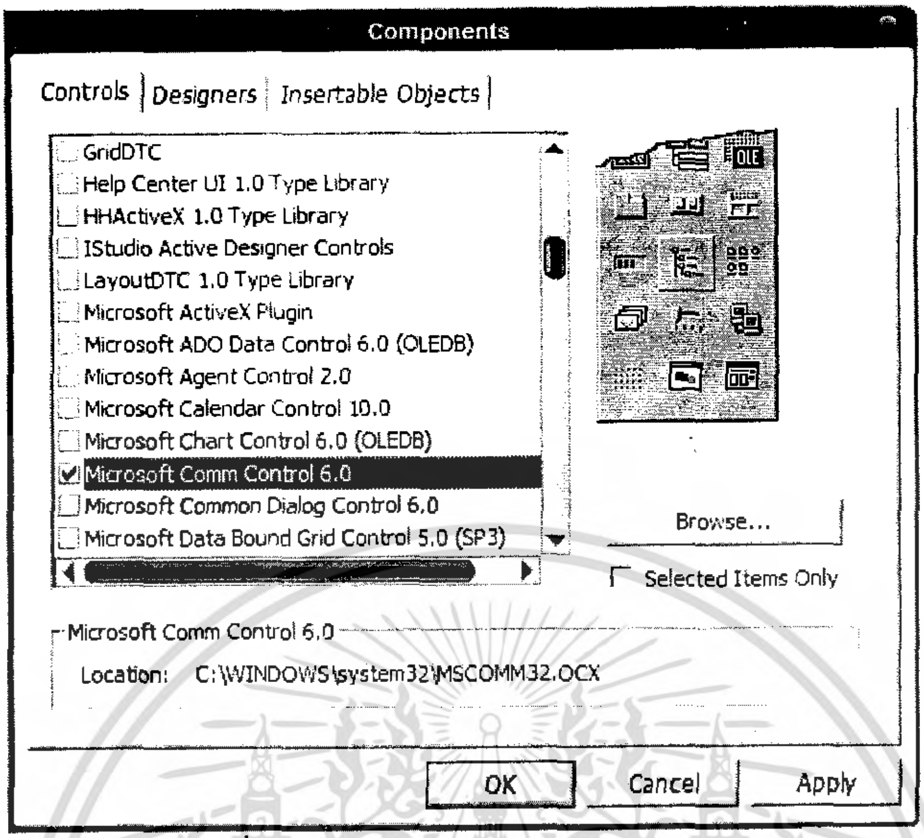
ตารางที่ 3.1 แสดงการตอบสนองการทำงานเมื่อคลิก Mouse และกดคีย์บอร์ด

Picture	ShotKey	Status	Picture	ShotKey	Status
	L	หยุด		D	หยุดกล้อง
	O	เดินหน้า		Z	หมุนกล้องขึ้น
	P	เลี้ยวขวาหน้า		V	หมุนกล้องซ้าย
	S	กลับตัวขวา		X	หมุนกล้องลง
	/	เลี้ยวขวาหลัง		C	หมุนกล้องขวา
	.	ถอยหลัง		B	หมุนกล้องมากลาง
	,	เลี้ยวซ้ายหลัง		W	เพิ่มความเร็ว
	A	กลับตัวซ้าย		Q	ลดความเร็ว
	I	เลี้ยวซ้ายหน้า		U	ขอรับค่าอุณหภูมิ
	Y	ClearDistance			

การเขียนโปรแกรมติดต่อและควบคุม Serial Port กับ Visual Basic

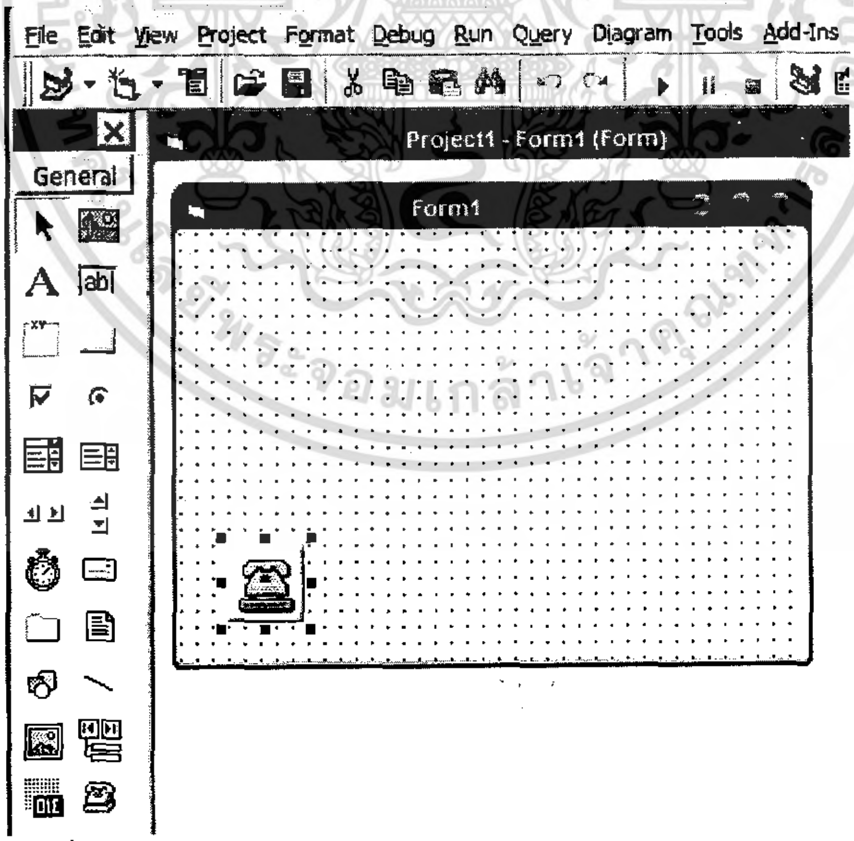
ในการติดต่อกับพอร์ทอนุกรมโดยใช้ VB นั้น จะต้องใช้คอนโทรลที่มีชื่อว่า MSCommซึ่งจะไม่ใช่คอนโทรลมาตรฐานดังนั้นเราจะต้องทำการ Add Component โดยเข้าไปที่ Project---> Component แล้วเลือกคลิกที่ MSComm แล้วจะมีรูปโทรศัพท์สีเหลืองซึ่งเป็นคอนโทรล MSComm ปรากฏอยู่ในคอนโทรลมาตรฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 แสดงการ Add Component MSComm

จากนั้นก็ไป ลาก Mouse นำ MSComm มาใช้



รูปที่ 3.10 แสดงการนำ Control MSComm มาใช้ใน Form ของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Property ที่สำคัญในการใช้งาน MSComm

CommPort คือการกำหนดหมายเลขพอร์ตที่จะใช้ติดต่อ RS 232

Settings ใช้ในการกำหนดอัตรา Buad,Parity,Data(จำนวนบิต),Stop ตัวอย่างเช่น 9600,n,8,1

PortOpen ใช้สำหรับการเปิดปิดการใช้งานพอร์ตอนุกรม โดยมีรูปแบบการใช้งานดังนี้

MSComm.PortOpen = true, MSComm.PortOpen = False

InBuffersize เป็นการกำหนดขนาดของบัฟเฟอร์ ในการรับข้อมูลเข้า

OutBuffersize เป็นการกำหนดขนาดของบัฟเฟอร์ในการส่งข้อมูลออก

InPutlen คือจำนวนของข้อมูลที่ไปอ่านในบัฟเฟอร์รับข้อมูล

EOFEnable เป็นการบอกสิ้นสุดของไฟล์(EOF) End of File

InBufferCount เมื่อกำหนดให้ = 0 จะเป็นการล้าง Buffer input

การกำหนดคุณสมบัติให้ MSComm เพื่อสามาถให้ติดต่อกับตัวพอร์ตได้

1 ต้องกำหนด Property ที่ชื่อ CommPort คือเลือกว่าจะใช้พอร์ตไหน ตัวอย่างเช่น

MSComm1.ComPort = 1 คือเลือกการทำงานกับพอร์ตคอม 1

2 Property Settings คือตั้งค่าอัตราการรับส่งข้อมูลซึ่งจะต้องรู้ว่าอัตราของอุปกรณ์ที่จะติดด้วย เป็นเท่าไหร่ โดยมีการรายละเอียดดังนี้

MSComm1.Settings = “Baud(อัตราการรับส่งข้อมูล),Parity(ถ้าไม่ใส่จะใส่ค่าN).

จำนวนบิต

ข้อมูล,BitStop ตัวอย่างเช่น MSComm1.Settings = “9600,N,8,1”

*3 Property InputLen คือ กำหนดขนาดขณะที่มีข้อมูลเข้ามาให้ไปอ่านข้อมูลทั้งหมดที่*อยู่ในบัฟเฟอร์

4 Property PortOpen คือการที่เราทำการเปิดใช้งานพอร์ตหรือไม่ ถ้าเปิด setค่า = true ถ้าปิด set ค่า = false เช่น MSComm1.PortOpen = true

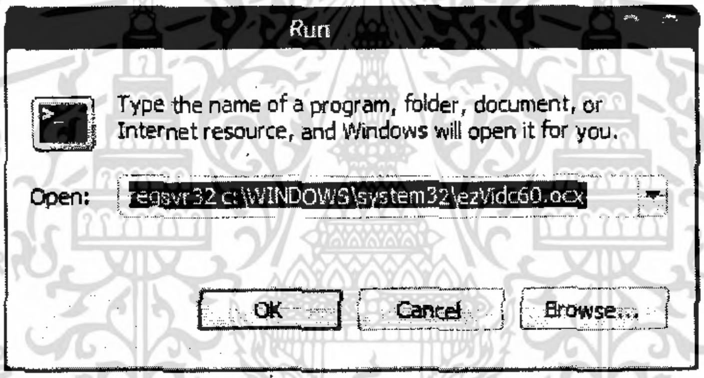
ในกรณีนี้ได้ใช้การset ค่าต่างๆไว้ในตอนเริ่มต้นเปิดโปรแกรมโดยเขียนไว้ใน โฟรซีเดอร์ชื่อ Sub Form_Load() **โดยใช้ดังนี้**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub Form_Load()
        MSComm1.Settings = "9600,N,8,1"
        MSComm1.PortOpen = True
        MSComm1.CommPort = 1
        MSComm1.InputLen = 1
End Sub
```

ส่วนการแสดงภาพจากกล้องวีดีโอในโปรแกรม VB

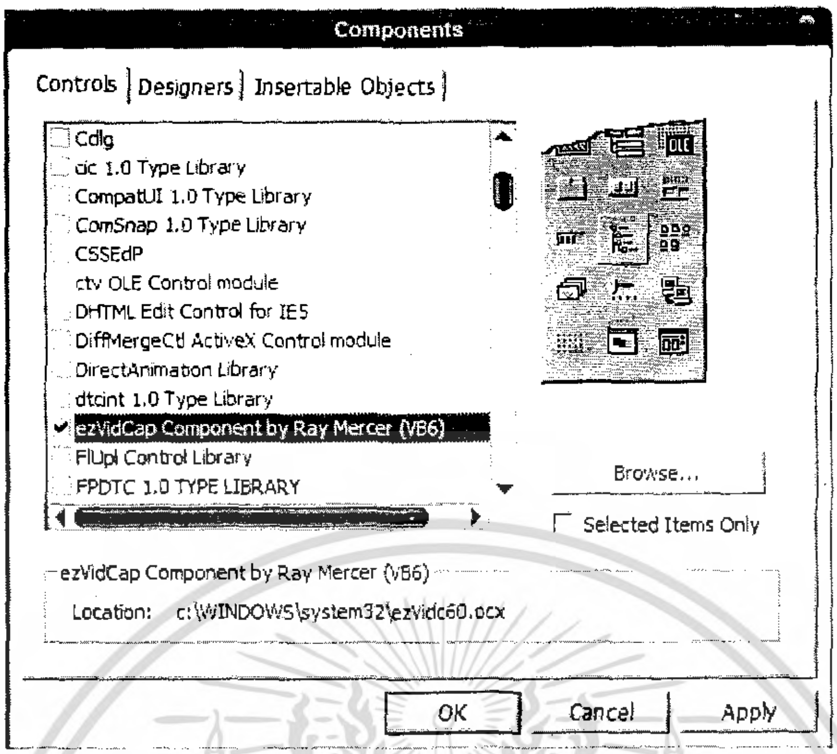
ในส่วนของการแสดงภาพในโปรแกรม VB จะต้องใช้ คอนโทรลที่มีชื่อว่า ezvid60 ซึ่งสามารถ Download ได้จาก www.shrinkwrapvb.com โดยเมื่อDownload มาแล้วจะต้องทำการลงทะเบียนโดยที่เราต้องทำการ copy file ezVidc60ไปไว้ที่ e:\WINDOW\system32 จากนั้นให้ไปที่ Start เมนู แล้ว Run โปรแกรม regsvr32 ตามด้วย path และ file ดังรูป



รูปที่ 3.11 การลงทะเบียนFile*.ocx

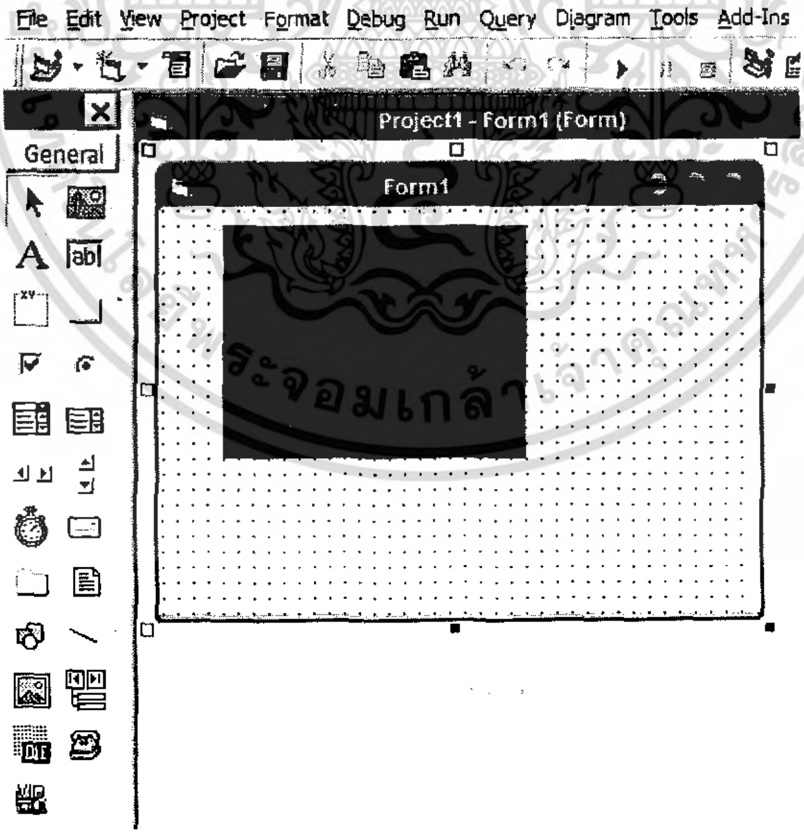
หลังจากนั้น Control นี้ ก็จะไปปรากฏใน Component ใน โปรแกรม VB ให้เรา add มาใช้ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 แสดง การ Add Control Component ezVidCap

ezVidc Control จะปรากฏใน หน้า Control หลักให้เราลากเม้าส์มาวางใน Form ได้เลย



รูปที่ 3.13 แสดงการนำ Control ezVidCap มาวางบน Form ของโปรแกรม

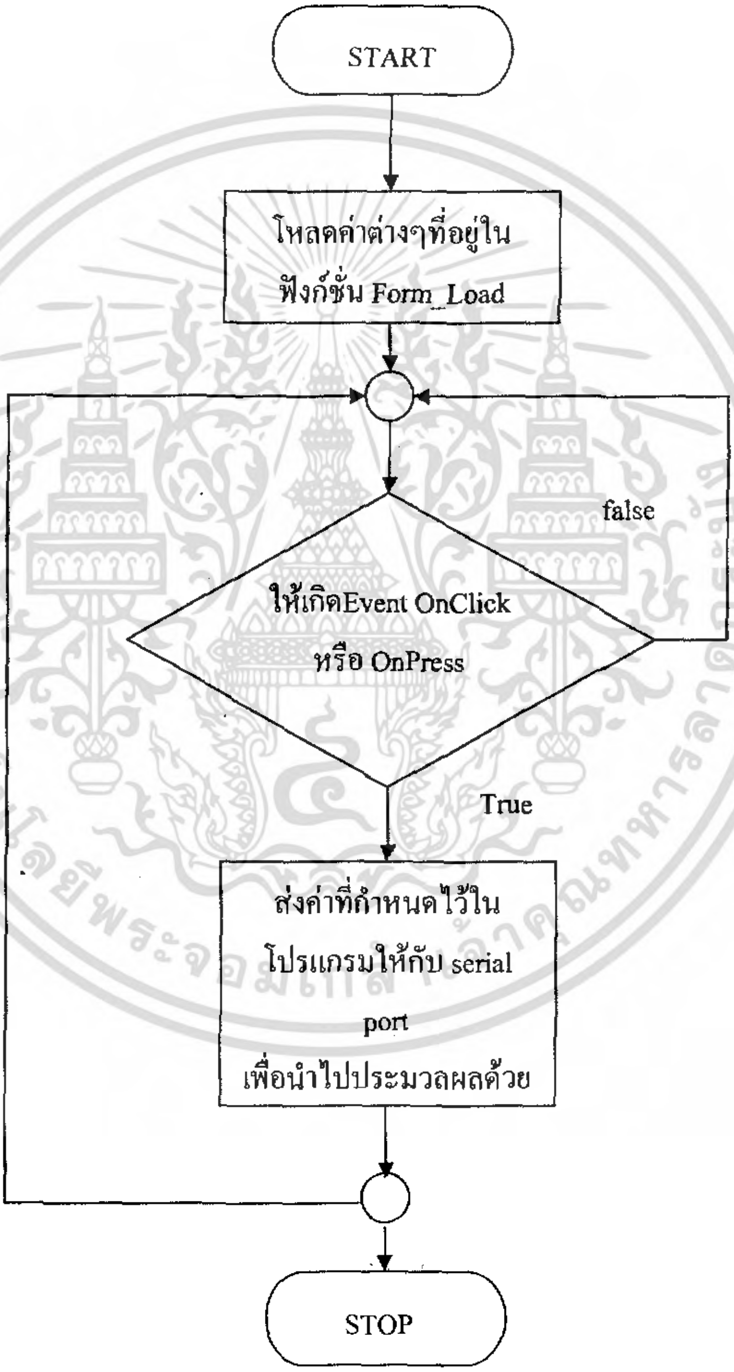
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอนโทรลนี้เมื่อนำมาวาง ภาพจากกล้องวีดีโอจะมาแสดงใน Form เลย ถ้าต้องการจะทำการเปลี่ยนขนาดต้องไปที่ Property แล้วเปลี่ยน ค่า Autosize ให้เป็น false ก็จะทำการขยายขนาดได้อย่างอิสระ

รูปแบบการทำงานของโปรแกรม Visual Basic

จะขอแบ่งการทำงานตามโปรแกรมเป็น 3 กรณีหลักๆ คือกรณีทั่วไปในการควบคุม กรณีเกิด Timer Interrupt และกรณี OnPress พิเศษ(กดตัว u)

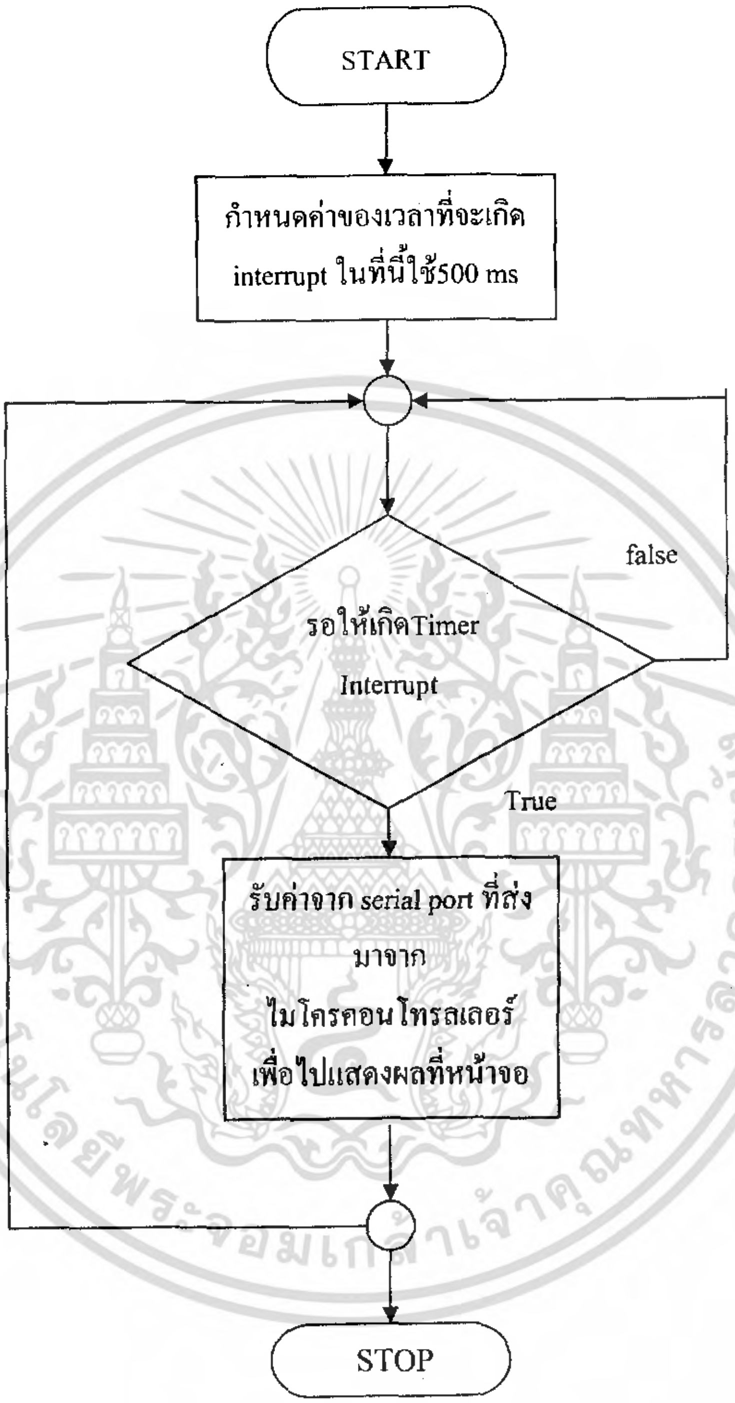
กรณีที่ 1 การควบคุมรถทั่วไป



รูปที่ 3.14 แสดงแผนผังการทำงานของโปรแกรม VB ในส่วนการควบคุมหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

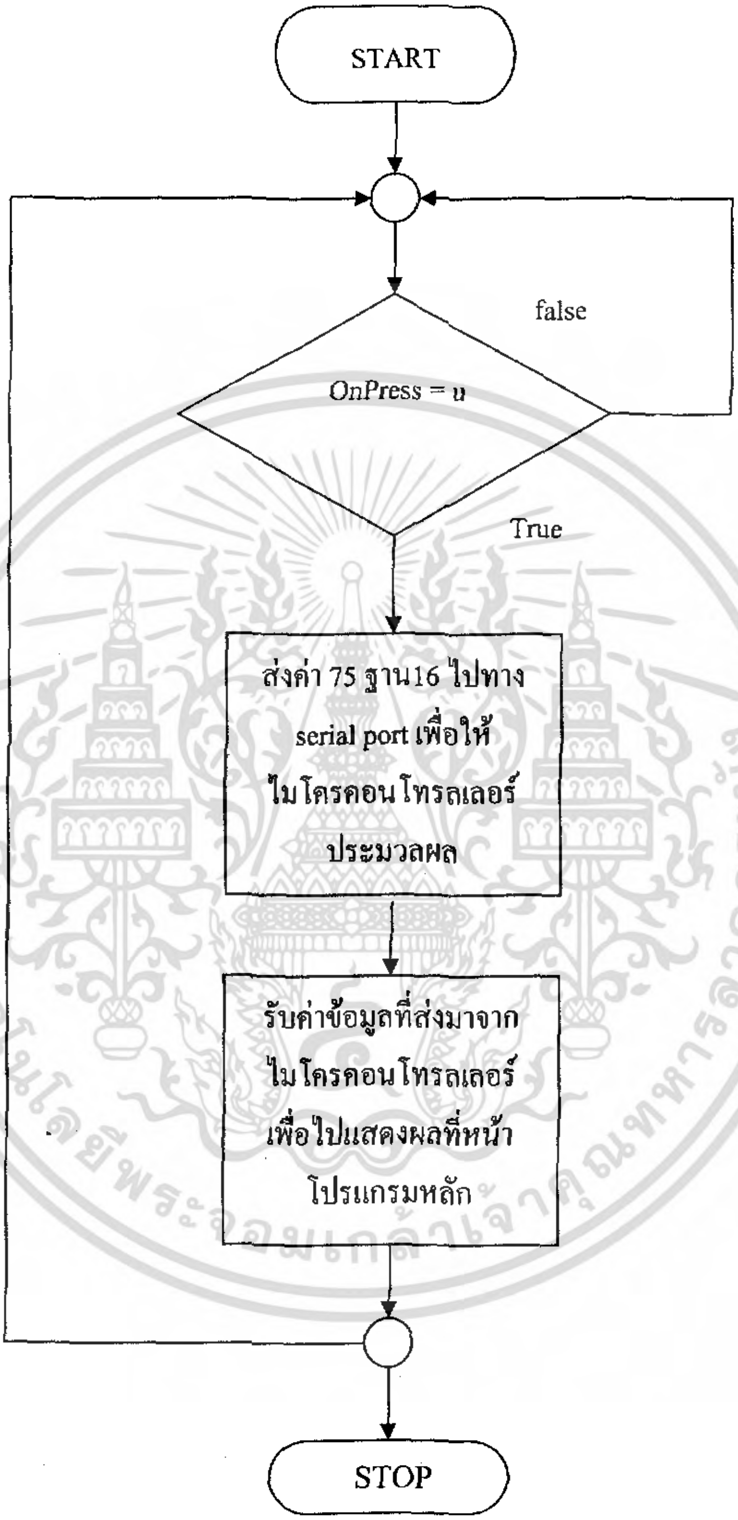
กรณีที่2 เกิด Timer Interrupt



รูปที่ 3.15 แสดงแผนผังการทำ งานของโปรแกรม VB เมื่อในตัวโปรแกรมเป็นเกิดTimer Interrupt

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีที่3 เมื่อ OnPress ที่กดเป็นตัว u



รูปที่ 3.16 แสดงแผนผังการทำงานของโปรแกรม VB เมื่อมีการกดปุ่ม U บนคีย์บอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

ในการทดลองวิจัยโครงงานนี้ได้ประกอบออกเป็น 3 ส่วนใหญ่ๆดังนี้

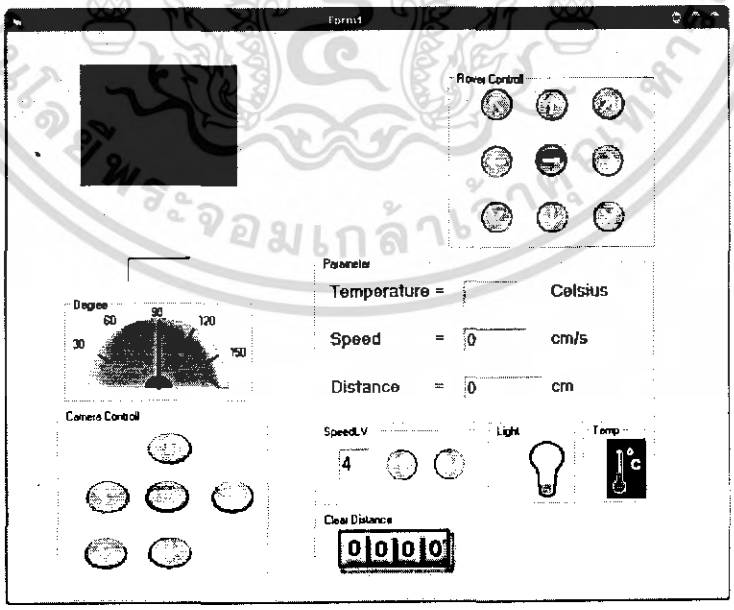
1. ผลการทดลองในส่วนรับส่งข้อมูลระหว่างคอมพิวเตอร์กับตัวรถสำรวจ
2. ผลการทดลองในส่วนรับส่งข้อมูลระหว่างรีโมทคอนโทลกับตัวรถสำรวจ
3. ผลการทดลองในส่วนของการขับเคลื่อน
4. ผลการทดลองในส่วนของระบบตรวจรู้

สามารถอธิบายผลการทดลองในส่วนย่อยๆลงไปได้อีกซึ่งประกอบด้วย

4.1 ผลการทดลองในส่วนรับส่งข้อมูลระหว่างคอมพิวเตอร์กับตัวรถสำรวจ

ทดลองรับส่งข้อมูลระหว่างคอมพิวเตอร์กับไมโครคอนโทรเลอร์โดยผ่านชุดรับส่งข้อมูล RS232 แบบไร้สาย

เราสามารถติดต่อและรับส่งข้อมูลกับรถสำรวจ ได้จากคอมพิวเตอร์โดยอาศัยโปรแกรมที่ติดต่อกับผู้ใช้งานคือ Visual Basic 6 ซึ่งจะติดต่อกับไมโครคอนโทรเลอร์ชุดรับส่งข้อมูล RS232 แบบไร้สาย 2 ตัวโดยตัวหนึ่งต่อกับคอมพิวเตอร์เพื่อรับค่าจากคีย์บอร์ดหรือเมาส์ และส่งข้อมูลไปยังชุดรับส่งข้อมูล RS232 แบบไร้สายอีกตัวหนึ่งซึ่งต่ออยู่กับตัวรถสำรวจ โดยมีหน้าตาของโปรแกรมดังรูปที่ 4.1



รูปที่ 4.1 แสดงหน้าตาโปรแกรม Visual Basic ที่ติดต่อระหว่างผู้ควบคุมรถสำรวจกับรถสำรวจ

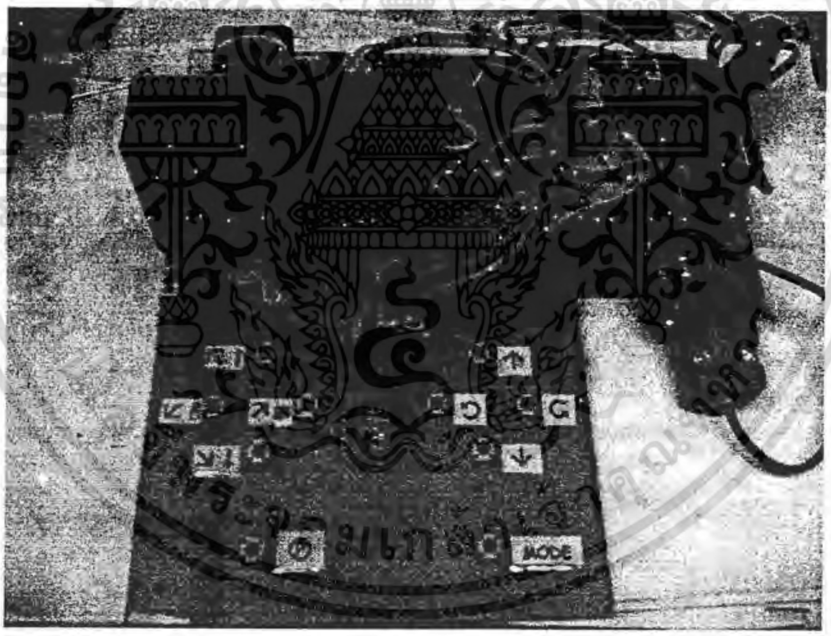
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป 4.1 จะเห็นว่าเราสามารถบังคับรถสำรวจได้โดยผ่านโปรแกรมด้านบน ซึ่งสามารถมองเห็นภาพการเคลื่อนที่ของรถสำรวจได้จากกล้องวิดีโอไร้สายที่หมุนไปรอบๆบริเวณนั้น โดยที่สามารถบังคับให้รถเลี้ยวซ้ายขวา ลดหรือเพิ่มความเร็ว หรือบังคับการหมุนของกล้อง เปิดปิดไปในส่วนที่ติดกับกล้อง นอกจากนี้ยังแสดงค่าของระยะทางการเคลื่อนที่ของรถ ความเร็วต่อรอบ ค่าของอุณหภูมิที่วัดได้

4.2 ผลการทดลองในส่วนรับส่งข้อมูลระหว่างรีโมทคอนโทลกับตัวรถสำรวจ

ทดลองรับส่งข้อมูลระหว่างรีโมทคอนโทลกับไมโครคอนโทรเลอร์โดยผ่านชุดรับส่งข้อมูล RS232 แบบไร้สาย

เราสามารถติดต่อกับรถสำรวจ ได้จากรีโมทคอนโทลกับไมโครคอนโทรเลอร์ชุดรับส่งข้อมูล RS232 แบบไร้สาย 2 ตัวโดยตัวหนึ่งต่อกับรีโมทคอนโทลเพื่อรับค่าต่างๆจากรีโมทคอนโทลและส่งข้อมูลไปยังชุดรับส่งข้อมูล RS232 แบบไร้สายอีกตัวหนึ่งซึ่งต่ออยู่กับตัวรถสำรวจ



รูปที่ 4.2 แสดงหน้าตารีโมทคอนโทล

จะเห็นว่าเราสามารถบังคับรถสำรวจได้โดยผ่านรีโมทคอนโทลโดยที่เราสามารถบังคับให้รถเลี้ยวซ้ายขวา ลดหรือเพิ่มความเร็ว หรือบังคับการหมุนของกล้อง เปิด - ปิดไฟในส่วนที่ติดกับ

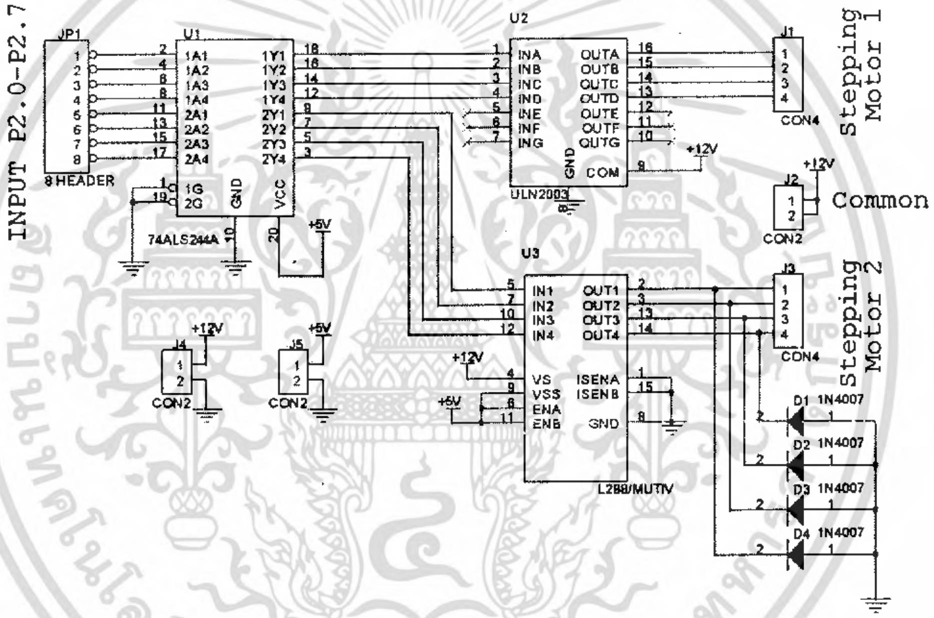
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กล้องได้เช่นเดียวกับการบังคับผ่านคอมพิวเตอร์ เพียงแต่ไม่มีส่วนของการแสดงผลที่ได้จากการสำรวจเช่นภาพ ความเร็วของรถ และอุณหภูมิ เป็นต้น

4.3 ผลการทดลองในส่วนของการขับเคลื่อน

4.3.1 ผลการทดลองขับสเต็ปปิ้งมอเตอร์

การทดลองการขับสเตปปิ้งมอเตอร์ที่ใช้ในการหมุนกล้อง โดยที่เราต่อวงจรการขับสเตปเปอร์มอเตอร์ ดังรูปที่ 4.3 การทดลองนี้จะมีสเตปปิ้งมอเตอร์ 2 ตัว ใช้ในการหมุนกล้องของหุ่นยนต์แบบ 2 ทิศทาง คือ การหมุนในแนวดิ่งและการหมุนในแนวราบ



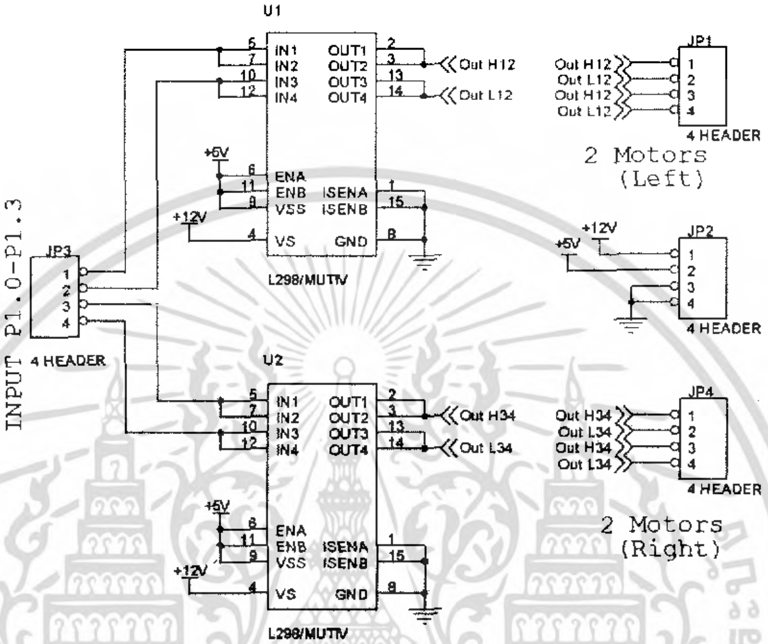
รูปที่ 4.3 รูปวงจรขับสเต็ปเปอร์มอเตอร์

โดยสเตปปิ้งมอเตอร์ที่ใช้หมุนในแนวดิ่งซึ่งเป็นสเตปปิ้งมอเตอร์ตัวเล็กนั้นกินกระแสไม่มาก (<200 mA)จึงสามารถใช้ IC ULN2003 ในการขับสเตปปิ้งมอเตอร์ตัวเล็กนี้ได้ แต่สเตปปิ้งมอเตอร์อีกตัวหนึ่งใช้ในการหมุนฐานของกล้องและอุปกรณ์ซึ่งมีภาระโหลดค่อนข้างมากจึงเลือกใช้สเตปปิ้งมอเตอร์ที่มีขนาดใหญ่ขึ้น ซึ่งสเตปปิ้งมอเตอร์ตัวใหญ่นี้ต้องการกระแสมากประมาณ (> 1A) จะทำให้สามารถทำงานได้อย่างเต็มกำลังด้วยเหตุนี้เราจึงไม่สามารถใช้ ULN2003 ได้เนื่องจากไอซีตัวนี้ขับกระแสได้เพียงไม่เกิน 500 mA เราจึงเลือกใช้ L298N ซึ่งเป็นไอซีขับกระแสสูงในการขับสเตปปิ้งมอเตอร์ตัวนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2 ผลการทดลองขับมอเตอร์กระแสตรง

การทดลองการขับมอเตอร์กระแสตรงซึ่งใช้เป็นส่วนในการขับเคลื่อนของรถสำรวจ สามารถทดลองได้โดยการต่อวงจรตามรูปที่ 4.4



รูปที่ 4.4 รูปวงจร DC Motor

ผลการทดลองการขับมอเตอร์กระแสตรงซึ่งใช้เป็นส่วนของการขับเคลื่อนของรถ ซึ่งมอเตอร์ที่ใช้นั้นมีความเร็วรอบ 100 RPM 12 V จากการทดลองจะเห็นว่าการใช้งาน DC Motor ค่อนข้างใช้งานได้ง่ายกว่ากว่าสเตปปิ้งมอเตอร์

4.4 ผลการทดลองในส่วนของระบบตรวจรู้

4.4.1 ผลการทดลองการวัดระยะทางการเคลื่อนที่ของรถสำรวจ

ทำการวัดระยะทางการเคลื่อนที่ของรถเปรียบเทียบกับระยะทางที่แท้จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 การวัดระยะทางการเคลื่อนที่

ระยะทาง (cm)	ระยะทางการการเคลื่อนที่รถ (cm)			เฉลี่ย (cm)
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	
10	7.6	7.8	8.7	8.03
20	18.7	17.4	18.6	18.23
30	27.8	28.2	27.3	27.77
40	38.6	38.5	38.6	38.57
50	48.7	48.1	48.2	48.33
60	58.1	59.0	58.4	58.50
70	68.5	68.6	67.9	68.33
80	77.8	78.6	78.7	78.37
90	88.1	88.5	87.5	88.03
100	98.6	98.7	98.2	98.50
120	118.5	117.4	118.2	118.03
130	128.8	128.5	127.9	128.40
140	138.1	138.7	139.0	138.60
150	148.0	148.7	148.5	148.40

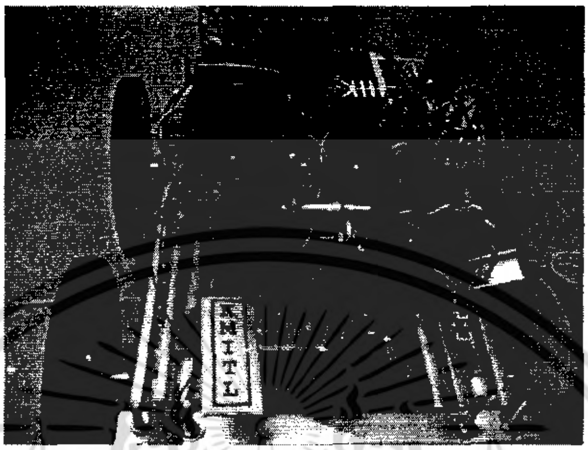
จากตารางการทดลองเราจะเห็นได้ว่าผลการวัดระยะทางที่ได้เมื่อนำมาเปรียบเทียบกับระยะทางที่แท้จริงนั้น ระยะทางการเคลื่อนที่ที่วัดได้จากเซ็นเซอร์นั้นค่อนข้างมีค่าใกล้เคียงกับระยะที่แท้จริง

4.4.2 ผลการทดลองการวัดมุมของการตั้งศูนย์กล้อง

ผลการทดลองการตั้งศูนย์กลางของกล้อง โดยการติดอินฟราเรดเซ็นเซอร์ไว้ที่มุมด้านซ้ายใกล้กับตัวกล้อง เมื่อเราสั่งให้กล้องทำการตั้งศูนย์กลางกล้อง กล้องก็จะรับคำสั่งให้หมุนมาทางซ้ายตลอดโดยที่เราติดแถบขาวไว้กับฐานของกล้อง เมื่อแถบขาวเจอกับอินฟราเรดเซ็นเซอร์เราก็ให้กล้องหมุนกลับมาที่ตำแหน่งศูนย์กลางกล้อง เนื่องจากเราใช้คุณสมบัติของอินฟราเรดเซ็นเซอร์ซึ่งอินฟราเรดนั้นสามารถสะท้อนแถบขาวได้ดี จะเห็นว่าเมื่อเราสั่งให้กล้องหมุนเพื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลับมาที่ตำแหน่งศูนย์กลางกล้อง เมื่อทำการทดลองซ้ำหลายๆครั้งจะเห็นว่า สเต็ปปิ้งมอเตอร์สามารถหมุนกล้องกลับมา ณ ตำแหน่งศูนย์กลางกล้องได้จุดศูนย์กลางกล้องพอดี



รูปที่ 4.5 แสดงตำแหน่งของการตั้งศูนย์กล้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุป ปัญหาและข้อเสนอแนะ

5.1 สรุปผลการทดลอง

โครงงานพิเศษนี้จะเป็นการออกแบบระบบเพื่อควบคุมรถสำรวจ โดยสามารถควบคุมการเคลื่อนที่ของรถสำรวจให้สามารถ เดินหน้า ถอยหลัง เลี้ยวซ้าย เลี้ยวขวา และสามารถบังคับกล้องวีดีโอไร้สาย ที่ทำน้าที่ส่งภาพในขณะที่รถเคลื่อนที่อยู่ได้ โดยตัวกล้องจะสามารถบังคับให้หมุนซ้าย หมุนขวา ก้ม เงย ได้ตามที่ผู้ควบคุมต้องการ โดยในการควบคุมจะใช้คอมพิวเตอร์ส่งข้อมูลไปที่เครื่องส่งสัญญาณแล้วเครื่องส่งสัญญาณส่งข้อมูลไปที่ไมโครคอนโทรเลอร์ที่ติดอยู่กับตัวรถเพื่อที่คอนโทรลเลอร์จะได้แปลงข้อมูลเพื่อใช้ในการควบคุมรถและจะมีส่วนที่คอนโทรลเลอร์ส่งข้อมูลกลับให้คอมพิวเตอร์ด้วย โดยส่วนควบคุมที่คอมพิวเตอร์จะใช้ Visual Basic ในการเขียนโปรแกรม และส่วนโปรแกรมในไมโครคอนโทรลเลอร์จะใช้ Micro C ในการเขียนโปรแกรม

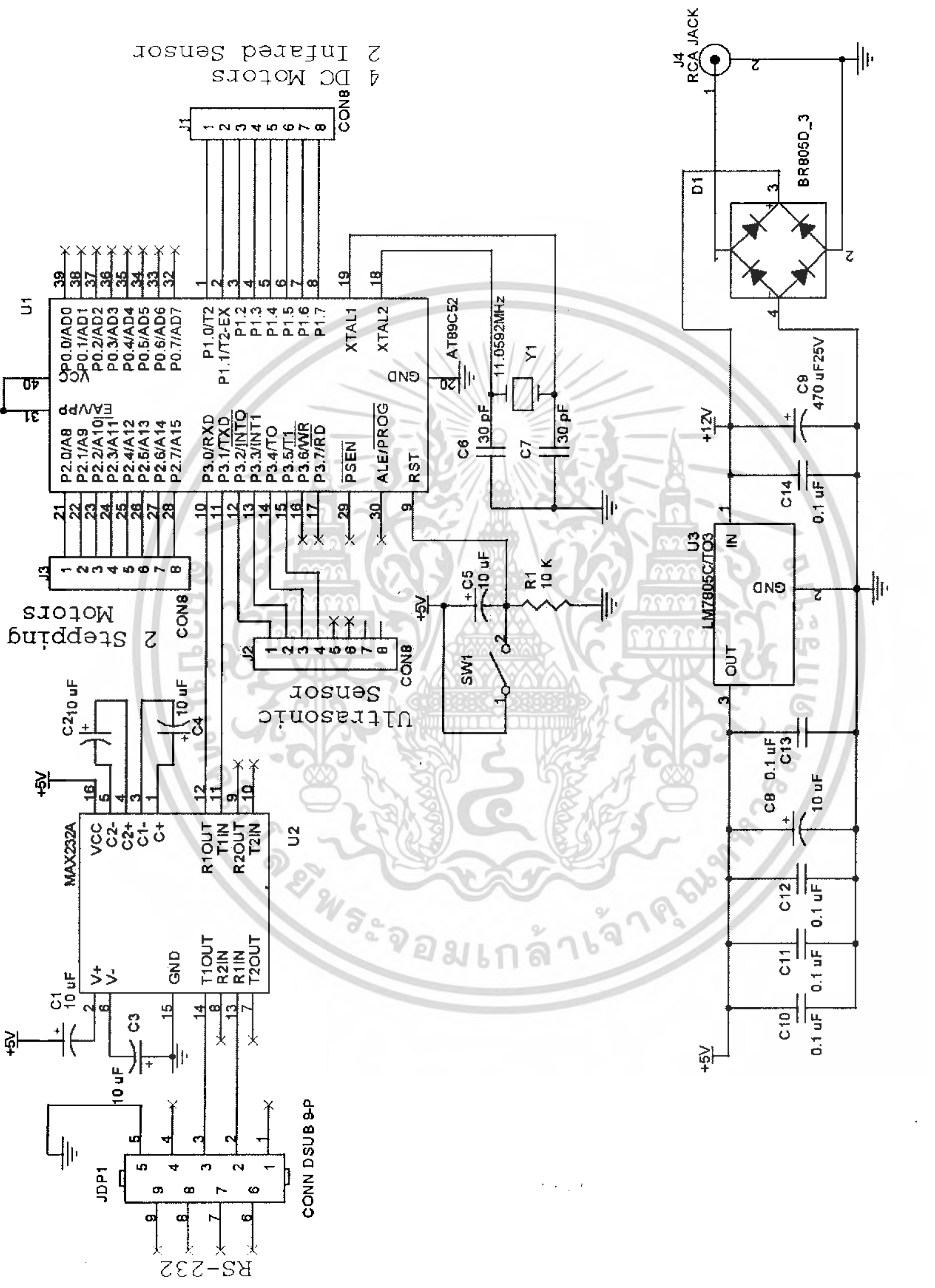
5.2ปัญหาที่พบและวิธีแก้ไข

ปัญหาที่พบในตอนแรกเกิดจากการใช้สายคีบต่อวงจรและการใช้แผ่นปริ้นไข่ปลาในการต่อวงจรจึงอาจเป็นผลทำให้ ตัวคอนเน็กเตอร์ไม่แน่น ต่อมาได้มีการแก้ไขโดยใช้การกัดแผ่นปริ้นก็เป็นผลทำให้ระบบดีขึ้น และตอนการเขียน VB มีปัญหาเกิดขึ้นคือเมื่อนำ Control ezVidcCapซึ่งเป็นส่วนที่ทำการ Capture ภาพจากกล้องวีดีโอไร้สาย มาวางใน Form ของโปรแกรมแล้วจะทำให้การทำงานของโปรแกรม VB ทุกฟังก์ชันช้าลงอย่างเห็นได้ชัดทุกฟังก์ชันเป็นผลเสียในการที่เราจะส่งค่าออกและรับค่าเข้ามาจากคอนโทรลเลอร์ก็จะทำได้ช้าเช่นกัน วิธีแก้ไขคือเราต้องศึกษาโปรแกรม VB ขั้น win-32 api แล้วเขียน Control ในการรับภาพขึ้นมาเองโดยจะต้องลดความสามารถบางอย่างของ Control นี้ ที่ทำให้ทำงานช้าลงไปบ้างเพื่อจะได้ทำงานได้เร็วขึ้น การรับส่งค่าก็จะมีความเป็น RealTime มากขึ้น

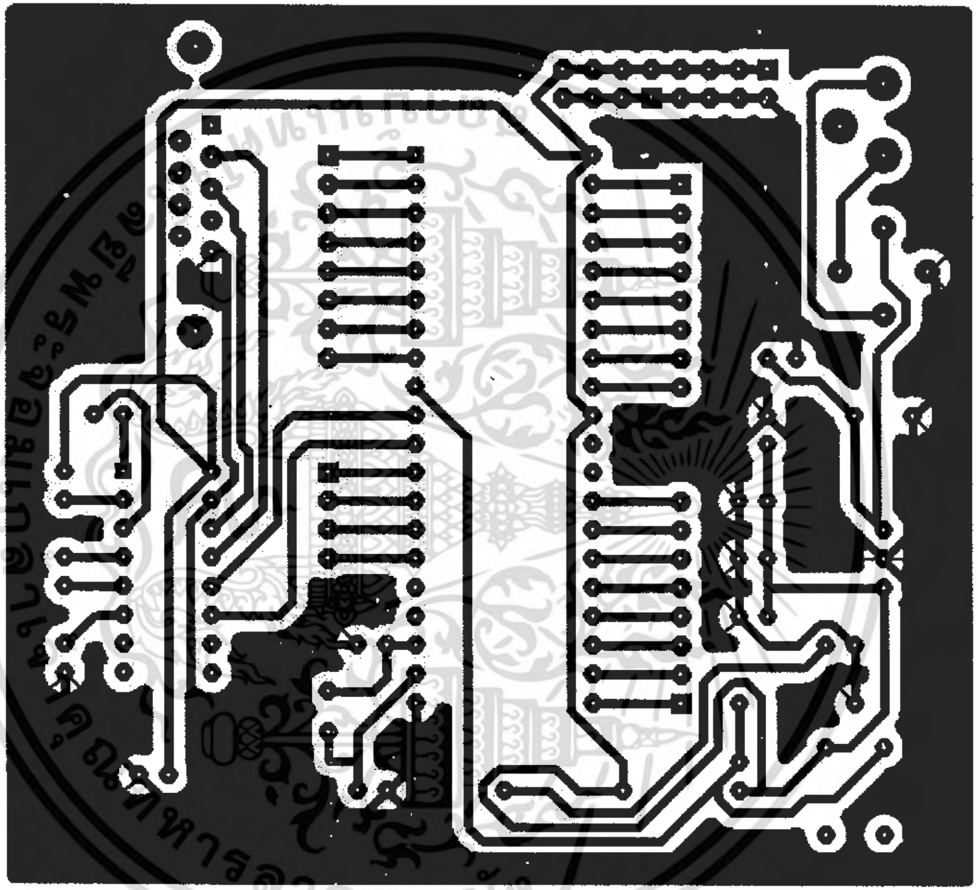
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

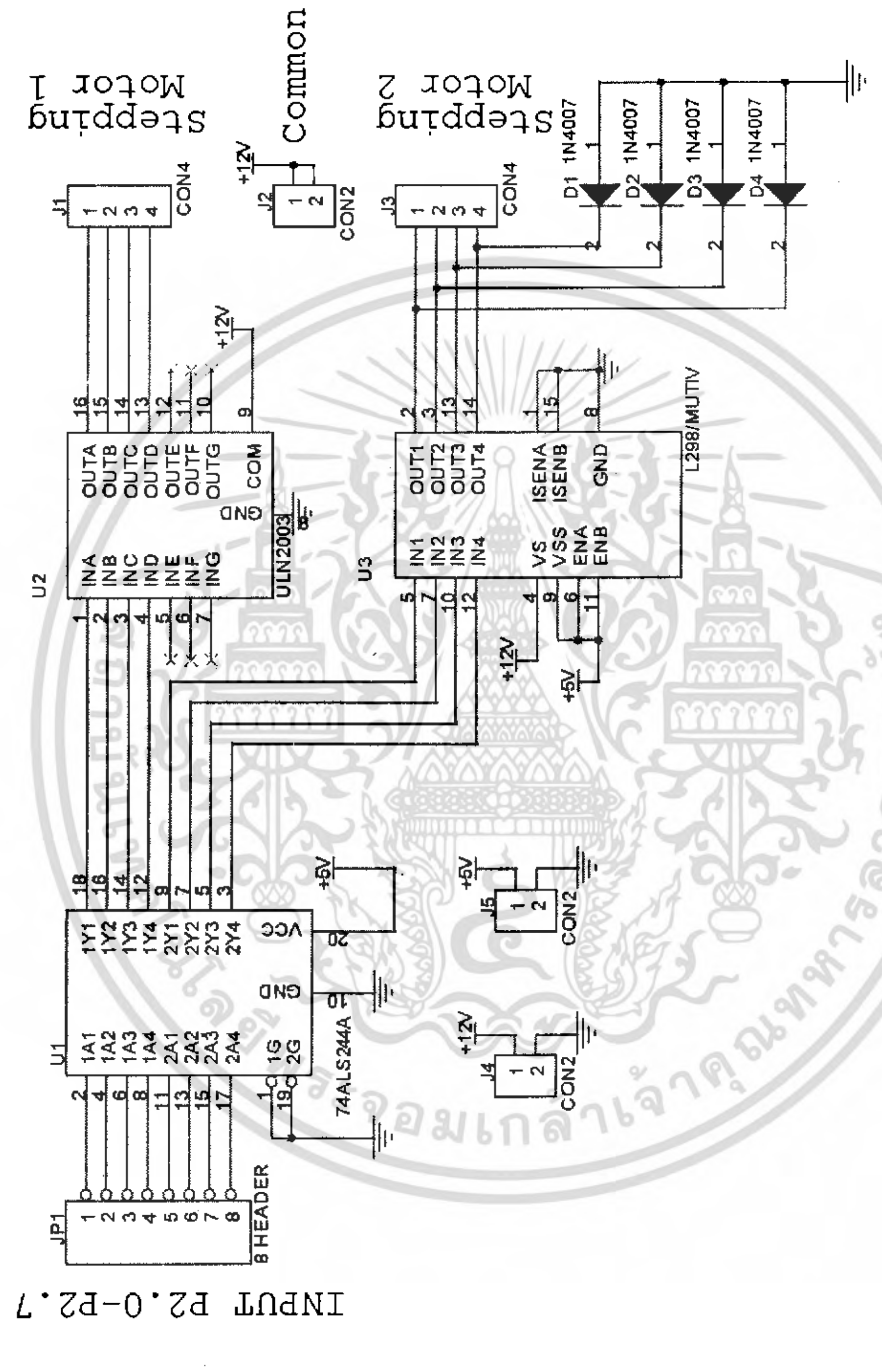


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

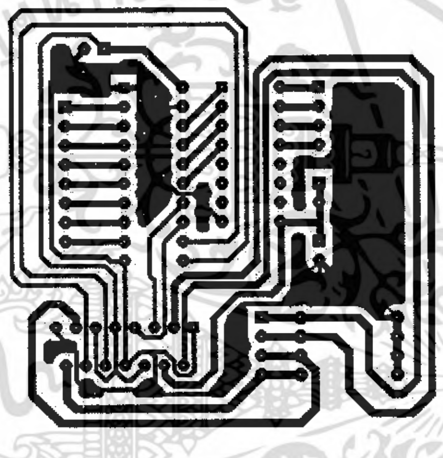


แสดงรูปลายปริ้นท์ด้านบนของส่วนควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

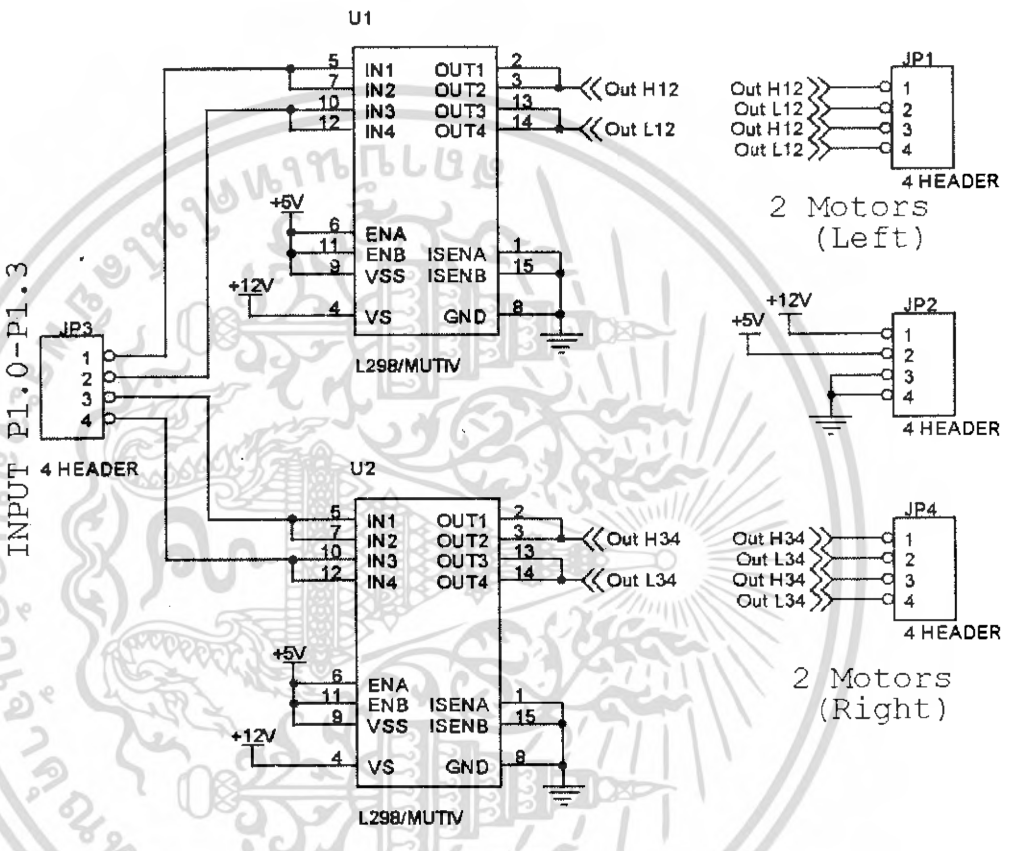


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



แสดงรูปลายปรินท์ด้านบนของส่วนขับสเตปปิงมอเตอร์ 2 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

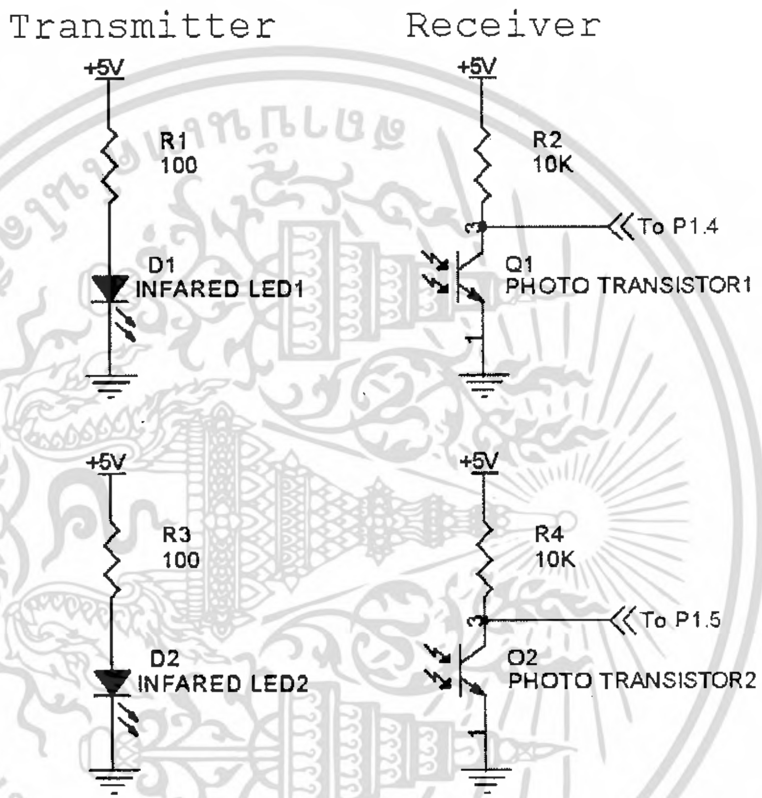


รูปวงจรขับมอเตอร์ 2 ชุด 4 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงรูปลายปรินท์ด้านบนของส่วนขับมอเตอร์ 4 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปวงจรอินฟาเรดนับระยะทางที่รถเคลื่อนที่ไป 2 ชุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงรูปลายปริ้นท์ด้านบนของวงจรอินฟาเรด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
' *****************************************************
'                 ส่วนของการประกาศตัวแปลและโหลด Graphic
'*****************************************************
Option Explicit


Dim neg As Single
Dim pos As Single
Dim c As Single ' declar compus
Dim v As Single ' declar compus
Dim d As Single
Dim r As Single
'+++++++++++++++++ Option Set ++++++++++++++++

Dim SpeedLV As Integer
Dim light As Boolean

'+++++++++++++++++ ConTroll set ++++++++++++++

Dim cam As Integer
Dim cambuf As Integer
Dim con As Integer
Dim conbuf As Integer

'++++++++++++++++ Polling set +++++++++++++

Dim colTemp As Collection
Dim colDis As Collection
Dim strTemp As String
Dim strbufT As String
Dim strL As String
Dim strbufD As String
Dim strR As String
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Dim strC As String

'+++++++++++++++        colTemp set     +++++++++++++

Dim Data As String
Dim i As Integer
Dim n As Integer

'++++++++++++++        colDis set      +++++++++++++

Dim distance As String
Dim valDistance As String
Dim valspeed As String
Dim m As Integer
Dim k As Integer

'+++++++++++++++++++ Light Toggle    +++++++++++++++++++
Private Sub LightDis()
light = light + 1
If light = False Then
imgLight.Picture = LoadPicture("c:\a\pic\bulbD.gif")
Else
imgLight.Picture = LoadPicture("c:\a\pic\bulbB.gif")
End If
End Sub
'+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

'+++++++++++    All Confunction  ++++++++++++
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Public Function con_all(con)
If con = 0 Then
ST.Picture = LoadPicture("C:\a\brightnew\STB.gif")
MS.Output = Chr(&H6C)
ElseIf con = 1 Then
arU.Picture = LoadPicture("c:\a\brightnew\arUB.gif")
MS.Output = Chr(&H6F)
ElseIf con = 2 Then
arRU.Picture = LoadPicture("c:\a\brightnew\arRUB.gif")
MS.Output = Chr(&H70)
ElseIf con = 3 Then
arR.Picture = LoadPicture("c:\a\brightnew\arRB.gif")
MS.Output = Chr(&H73)
ElseIf con = 4 Then
arRD.Picture = LoadPicture("c:\a\brightnew\arRDB.gif")
MS.Output = Chr(&H2F)
ElseIf con = 5 Then
arD.Picture = LoadPicture("c:\a\brightnew\arDB.gif")
MS.Output = Chr(&H2E)
ElseIf con = 6 Then
arLD.Picture = LoadPicture("c:\a\brightnew\arLDB.gif")
MS.Output = Chr(&H2C)
ElseIf con = 7 Then
arL.Picture = LoadPicture("c:\a\brightnew\arLB.gif")
MS.Output = Chr(&H61)
ElseIf con = 8 Then
arLU.Picture = LoadPicture("c:\a\brightnew\arLUB.gif")
MS.Output = Chr(&H69)
Else
End If
End Function
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Public Function conbefore(con)
If con = 0 Then
ST.Picture = LoadPicture("C:\a\darknew\STD.gif")
ElseIf con = 1 Then
arU.Picture = LoadPicture("c:\a\darknew\arUD.gif")
ElseIf con = 2 Then
arRU.Picture = LoadPicture("c:\a\darknew\arRUD.gif")
ElseIf con = 3 Then
arR.Picture = LoadPicture("c:\a\darknew\arRD.gif")
ElseIf con = 4 Then
arRD.Picture = LoadPicture("c:\a\darknew\arRDD.gif")
ElseIf con = 5 Then
arD.Picture = LoadPicture("c:\a\darknew\arDD.gif")
ElseIf con = 6 Then
arLD.Picture = LoadPicture("c:\a\darknew\arLDD.gif")
ElseIf con = 7 Then
arL.Picture = LoadPicture("c:\a\darknew\arLD.gif")
ElseIf con = 8 Then
arLU.Picture = LoadPicture("c:\a\darknew\arLUD.gif")
Else
End If
End Function

Private Sub conrenew()
Do Until con = conbuf
conbefore (con)
con = conbuf
con_all (con)
Loop
End Sub
```

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
'++++++++++++++++  All CamFunction  ++++++++++++++
Public Function cam_all(cam)
If cam = 0 Then
CStop.Picture = LoadPicture("C:\a\percam\CStopB.gif")
MS.Output = Chr(&H64)
ElseIf cam = 1 Then
CUp.Picture = LoadPicture("C:\a\percam\UpB.gif")
MS.Output = Chr(&H7A)
ElseIf cam = 2 Then
CRight.Picture = LoadPicture("C:\a\percam\RightB.gif")
MS.Output = Chr(&H76)
ElseIf cam = 3 Then
CDown.Picture = LoadPicture("C:\a\percam\DownB.gif")
MS.Output = Chr(&H78)
ElseIf cam = 4 Then
Cleft.Picture = LoadPicture("C:\a\percam\LeftB.gif")
MS.Output = Chr(&H63)
Else
End If

End Function

Public Function cambefore(cam)
If cam = 0 Then
CStop.Picture = LoadPicture("C:\a\percam\CStopD.gif")
ElseIf cam = 1 Then
CUp.Picture = LoadPicture("C:\a\percam\UpD.gif")
ElseIf cam = 2 Then
CRight.Picture = LoadPicture("C:\a\percam\RightD.gif")
ElseIf cam = 3 Then
CDown.Picture = LoadPicture("C:\a\percam\DownD.gif")
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ElseIf cam = 4 Then
Cleft.Picture = LoadPicture("C:\a\percam\LeftD.gif")
Else
End If

End Function

Private Sub camrenew()
Do Until cam = cambuf
cambefore (cam)
cam = cambuf
cam_all (cam)
Loop
End Sub

'+++++++++++ First_StatusSpeed +++++++++++++++

Private Sub Sfirst_status()
SUp.Picture = LoadPicture("C:\a\Speed\SUpD.gif")
SDown.Picture = LoadPicture("C:\a\Speed\SDownD.gif")
End Sub

'++++++++++++++  Image Speed  +++++++++++++
Private Sub img_SUp()
MS.Output = Chr(&H77)
If SpeedLV >= 4 Then
SpeedLV = 4
Else
SpeedLV = SpeedLV + 1
End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
txtSpeed = SpeedLV
Sfirst_status
SUp.Picture = LoadPicture("c:\a\Speed\SUpB.gif")
End Sub

Private Sub img_SDown()
MS.Output = Chr(&H71)
If SpeedLV <= 1 Then
SpeedLV = 1
Else
SpeedLV = SpeedLV - 1
End If
txtSpeed = SpeedLV
Sfirst_status
SDown.Picture = LoadPicture("c:\a\speed\SDownB.gif")
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
'************************************************
'               ส่วนการทำงานในช่วงเริ่มเปิดโปรแกรม
'++++++++++++++++ Form Load +++++++++++++
'************************************************
Private Sub Form_Load()
MS.CommPort = 1
MS.Settings = "9600,n,8,1"
MS.PortOpen = True
Timer1.Interval = 100
Timer2.Interval = 1
Timer1.Enabled = False
Timer2.Enabled = False
Timer3.Enabled = True
Timer4.Enabled = False
Timer5.Enabled = False
Set colTemp = New Collection
Set colDis = New Collection
cam = 0
con = 0
Sfirst_status
SpeedLV = 4
imgLight.Picture = LoadPicture("c:\a\pic\bulbD.gif")
CStop.Picture = LoadPicture("C:\a\percam\CStopB.gif")
ST.Picture = LoadPicture("c:\a\brightnew\STB.gif")
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
'*********************************************
'    ส่วนของโปรแกรมเมื่อเกิดการกดปุ่มที่ KeyBoard
'++++++++++++  command KeyPress   ++++++++++
'*********************************************
Private Sub command1_KeyPress(KeyAscii As Integer)
MS.Output = Chr(KeyAscii) '**************
Select Case KeyAscii
Case &H75 ' Key press U
Timer1.Enabled = True

Case &H6C  'stop
conbuf = 0
conrenew
'Debug.Print (strR)
'Debug.Print (strL)
Case &H6F ' U
conbuf = 1
conrenew
Case &H70 ' RU
conbuf = 2
conrenew

Case &H73 'R
conbuf = 3
conrenew

Case &H2F ' RD
conbuf = 4
conrenew

Case &H2E  'D
conbuf = 5
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
conrenew

Case &H2C ' LD
conbuf = 6
conrenew

Case &H61 'L
conbuf = 7
conrenew

Case &H69 ' LU
conbuf = 8
conrenew

'+++++++++++    CamSet    +++++++++++
Case &H64   ' Key d StopCam
cambuf = 0
camrenew
Case &H7A 'key z CUp
cambuf = 1
camrenew
Case &H76  ' Key v CRight
cambuf = 2
camrenew
Case &H78 'key X CDown
cambuf = 3
camrenew
Case &H63 ' key c CLeft
cambuf = 4
camrenew

'++++++++++++++++++++++++++++++
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Case &H66 ' Key f light on
LightDis

End Select
End Sub

'+++++++++   Command KeyDown  ++++++++++++++++++
Private Sub Command1_KeyDown(Keycode As Integer, Shift As Integer)
Select Case Keycode
Case vbKeyW
img_SUp
Case vbKeyQ
img_SDown
Case vbKeyB
CC.Picture = LoadPicture("c:\a\percam\CCB.gif")
cambuf = 0
camrenew
MS.Output = Chr(&H62)
End Select
End Sub

'++++++++   Command KeyUp   ++++++++++++++++++++
Private Sub Command1_KeyUp(Keycode As Integer, Shift As Integer)
Select Case Keycode
Case vbKeyB
CC.Picture = LoadPicture("c:\a\percam\CCD.gif")
Case vbKeyW
Sfirst_status
Case vbKeyQ
Sfirst_status
End Select
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
'*******************************************************
'                    โปรแกรมส่วนที่เป็น On click ต่างๆ
'*******************************************************
Private Sub imgLight_Click()
LightDis
MS.Output = Chr(&H66)
End Sub

'++++++++++++++++ Onclick Rover controll +++++++++++++++++
Private Sub ST_Click()
conbuf = 0
conrenew
End Sub
Private Sub arU_Click()
conbuf = 1
conrenew
End Sub

Private Sub arRU_Click()
conbuf = 2
conrenew
End Sub
Private Sub arR_Click()
conbuf = 3
conrenew
End Sub

Private Sub arRD_Click()
conbuf = 4
conrenew
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub arD_Click()
conbuf = 5
conrenew
End Sub

Private Sub arLD_Click()
conbuf = 6
conrenew
End Sub
Private Sub arL_Click()
conbuf = 7
conrenew
End Sub

Private Sub arLU_Click()
conbuf = 8
conrenew
End Sub

'++++++++++++++++ Onclick Camera controll +++++++++++++++++

Private Sub CUp_Click()
cambuf = 1
camrenew
End Sub

Private Sub CDown_Click()
cambuf = 3
camrenew
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub CRight_Click()
cambuf = 2
camrenew
End Sub

Private Sub CLeft_Click()
cambuf = 4
camrenew
End Sub

Private Sub CStop_Click()
cambuf = 0
camrenew
End Sub

'++++++++++++++++++++++    Onclick  Clear Temp ++++++++++++++

Private Sub CDistance_Click()
MS.Output = Chr(&H79)
End Sub


'++++++++++++++++++++++++    Onclick Call Temp ++++++++++++++++
Private Sub CTemp_Click()
MS.Output = Chr(&H75)
End Sub


'++++++++++++++++++++++++    Mouse    ++++++++++++++++++++++++++++

Private Sub CC_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
```
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
cambuf = 0
camrenew
CC.Picture = LoadPicture("c:\a\percam\CCB.gif")
MS.Output = Chr(&H62)
End Sub

Private Sub CC_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
CC.Picture = LoadPicture("c:\a\percam\CCD.gif")
End Sub

Private Sub SUp_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
img_SUp
End Sub

Private Sub SUp_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
Sfirst_status
End Sub

Private Sub SDown_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
img_SDown
End Sub

Private Sub SDown_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
Sfirst_status
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
'*************************************************
' ส่วนของโปรแกรมในการทำงานร่วมกัน ระหว่าง
' Timer Interrupt ,Polling (.ใช้ในการรับข้อมูลไปแสดงผล)
' และการวาดเส้นเพื่อบอกทิศทางของกล้อง
'*************************************************

Private Sub Timer1_Timer()
Timer2.Enabled = True

Timer1.Enabled = False

End Sub
Private Sub Timer3_Timer()
MS.Output = Chr(&H68)
'Debug.Print ("CCC")
Timer4.Enabled = True
End Sub
Private Sub Timer4_Timer()
Timer5.Enabled = True
Timer4.Enabled = False
End Sub

Private Sub Timer2_Timer()
Data = MS.Input
If Data = "" Then
Else

'+++++++++++++ add to collect ++++++++++++++
colTemp.Add (Data)
If Hex(Asc(Data)) = Hex(116) Then
MS.InBufferCount = 0
Timer2.Enabled = False
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
For i = 2 To colTemp.Count
n = i - 1

strbufT = strbufT + colTemp.Item(n)
If Hex(Asc(colTemp.Item(i))) = Hex(116) Then
strTemp = strbufT
i = i + 1
strbufT = ""
'ElseIf Hex(Asc(coltest.Item(i))) = Hex(114) Then
'StrR = Strbuf
'i = i + 1
'Strbuf = ""
'ElseIf Hex(Asc(coltest.Item(i))) = Hex(100) Then

'StrD = Strbuf
'Strbuf = ""
End If
Next
strTemp = strTemp / 2
txtTemp = strTemp
'++++++++++++ clear collect 100 = d ,116 = t 114 = r +++++++++++++
Do Until colTemp.Count = 0
colTemp.Remove (colTemp.Count)

Loop

End If
End If
End Sub

Private Sub Timer5_Timer()
distance = MS.Input
```
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
If distance = "" Then
Else

'+++++++++++++  add to collect  ++++++++++++++
colDis.Add (distance)
If Hex(Asc(distance)) = Hex(97) Then
MS.InBufferCount = 0
Timer5.Enabled = False
For m = 2 To colDis.Count
k = m - 1

strbufD = strbufD + colDis.Item(k)
If Hex(Asc(colDis.Item(m))) = Hex(112) Then
strL = strbufD
m = m + 1
strbufD = ""
ElseIf Hex(Asc(colDis.Item(m))) = Hex(106) Then
strR = strbufD
m = m + 1
strbufD = ""
ElseIf Hex(Asc(colDis.Item(m))) = Hex(97) Then
strC = strbufD
m = m + 1
strbufD = ""
Debug.Print strC

If Val(strC) <= 0 Then
neg = Val(strC) + 55
c = (neg * 90) / 55
v = (c) * 3.141592654 / 180
txtDegree = c
Line1.X2 = 1440 - (Cos(v) * 900)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Line1.Y2 = 1365 - (Sin(v) * 900)

Else
pos = Val(strC) + 55
Debug.Print (pos)
txtDegree = (pos * 180 / 100)
d = 180 - ((pos * 180) / 100)
r = d * 3.141592654 / 180

Line1.X2 = 1440 + (Cos(r) * 900)
Line1.Y2 = 1365 - (Sin(r) * 900)
End If

End If
Next
valDistance = ((Val(strL) + Val(strR)) / 2) * 5.75
valspeed = (Val(valDistance) - Val(txtDistance))
txtDistance = valDistance
txtSP = Val(valspeed) / 0.5

'txtSP = strDis(5.75)
'++++++++++++ clear collect 100 = d ,116 = t 114 = r +++++++++++++
Do Until colDis.Count = 0
colDis.Remove (colDis.Count)

Loop

End If
End If
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนโปรแกรมที่ใช้ Micro C เขียนเพื่อสั่งการทำงานของรถสำรวจ

```
#include "c:\mc\8051io.h"
#include "c:\mc\8051reg.h"
#include "c:\mc\8051bit.h"
#include "c:\mc\8051intt.h"

register unsigned char i,j,k,flag,save,dis,ccam,tickkey;
register char key,key2,neem,x,tick,tick2,navl,navr;
register unsigned int val,n,tick3,timer1,timer2,timerc;
register int count,count2,veccam;

INTERRUPT(_TF2_) timer2_interrupt()
{
    clrbit(T2CON.7)
    timer1++;
    timer2++;
}
move()
{
switch(key)
{
        case('z'):
        {
    clrbit(P0.3)
    setbit(P0.0)
        delay(neem);
    clrbit(P0.0)
    setbit(P0.1)
        delay(neem);
    clrbit(P0.1)
    setbit(P0.2)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
delay(neem);
clrbit(P0.2)
setbit(P0.3)
break;
}
case('x'):
{
clrbit(P0.0)
setbit(P0.3)
delay(neem);
clrbit(P0.3)
setbit(P0.2)
delay(neem);
clrbit(P0.2)
setbit(P0.1)
delay(neem);
clrbit(P0.1)
setbit(P0.0)
break;
}
case('c'): //Cam Left
{
delay(14);
veccam--;
switch(i)
{
case(0):
{
clrbit(P0.7)
setbit(P0.4)
break;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
case(1):
{
clrbit(P0.4)
setbit(P0.5)
break;
}
case(2):
{
clrbit(P0.5)
setbit(P0.6)
break;
}
case(3):
{
clrbit(P0.6)
setbit(P0.7)
break;
}
}
if(++i>3)
i=0;
break;
}
case('v'): //Cam Right
{
delay(14);
veccam++;
switch(x)
{
case(0):
{
clrbit(P0.4)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
setbit(P0.7)
break;
}
case(1):
{
clrbit(P0.7)
setbit(P0.6)
break;
}
case(2):
{
clrbit(P0.6)
setbit(P0.5)
break;
}
case(3):
{
clrbit(P0.5)
setbit(P0.4)
break;
}
}
if(++x>3)
x=0;
break;
}
//Drive 4-Motors*************************************************************//
case('o'):
{
clrbit(P1.0)
setbit(P1.1)
clrbit(P1.2)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
setbit(P1.3)
navl=navr=1;
break;
}
case('.'):
{
//Forward//
setbit(P1.0)
clrbit(P1.1)
setbit(P1.2)
clrbit(P1.3)
navl=navr=2;
break;
}
case('l'):
{
//Backward//
navl=navr=0;
P1&=0xF0;
break;
}
case('/'):
{
//Backward Right //
clrbit(P1.0)
clrbit(P1.1)
setbit(P1.2)
clrbit(P1.3)
navr=navl=2;
break;
}
case(','):
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
//Backward Left//
setbit(P1.0)
clrbit(P1.1)
clrbit(P1.2)
clrbit(P1.3)
navr=navl=2;
break;
}
case('p'):
{
//Forward Right//
clrbit(P1.0)
clrbit(P1.1)
clrbit(P1.2)
setbit(P1.3)
navr=navl=1;
break;
}
case('i'):
{
//Forward Left//
clrbit(P1.0)
setbit(P1.1)
clrbit(P1.2)
clrbit(P1.3)
navr=navl=1;
break;
}
case('a'):
{
//Rotage Left//
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
clrbit(P1.0)
setbit(P1.1)
setbit(P1.2)
clrbit(P1.3)
navr=1;
navl=2;
break;
}
case('s'):
{
//Rotage Right//
setbit(P1.0)
clrbit(P1.1)
clrbit(P1.2)
setbit(P1.3)
navr=2;
navl=1;
break;
}
}
}
chk_delay()
{
        if(key=='q')
        {
        if(--val<3)
        val=3;
        printf("\n val is %d level %d",val,val-3);
        }
        if(key=='w')
        {
          if(++val>7)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    val=7;
                printf("\n val is %d level %d",val,val-3);
                }
if(val!=7)
        {
        if((tick3==0)&&(timer1>=val))   //delay(val);
        {
        timer1=0;
        save=(P1&0x0F);
        P1&=0xF0;
        tick3=1;
        }
        if((tick3==1)&&(timer1>=(7-val)))       //delay(50-val);
        {
        P1=P1|save;
        timer1=tick3=0;
        }
        }
}

setinfared()
{
        setbit(P3.4)
        setbit(P3.5)
}
chk_sta()//left//
{
        if(P3&0x10) //P3.4
        {
                if(!tick)
                {
                tick=1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
		if(navl==1)
		count++;
		if(navl==2)
		count--;
		}
	}
	else
		{
		tick=0;
		}
}
chk_sta2()//right//
{
	if(P3&0x20) //P3.5
	{
		if(!tick2)
		{
		tick2=1;
		if(navr==1)
		count2++;
		if(navr==2)
		count2--;
		}
	}
	else
		{
		tick2=0;
		}
}
set_cam()
{
	if((P3&0x08)&&(ccam!=2))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            {
            ccam=1;
            key=0x76; //key v
            timer2=0;
            }
        if(!(P3&0x08))
            ccam=2;
        if(ccam==2)
            {
            if(timer2<92)
            key=0x63;  //key c
            else
            {
            ccam=timerc=veccam=0;
            }
            }
}
pollkey()
{
        key=chkch();
        if(key=='z')
            tickkey=1;
        if(key=='x')
            tickkey=2;
        if(key=='c')
            tickkey=3;
        if(key=='v')
            tickkey=4;
        if(key=='d'||key=='l')
            tickkey=0;
        if(tickkey!=0)
        {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                if(tickkey==1)
                        key=0x7A;  //z
                if(tickkey==2)
                        key=0x78;  //x
                if(tickkey==3)
                        key=0x63;  //c
                if(tickkey==4)
                        key=0x76;          //v
        }
}
ds1820()
{
   StartDS1820()
   writeDS1820(0xCC);   //Skip ROM
   writeDS1820(0x44);   //Convert Temperature
   delay(5);        //Change
   while (InitDS1820());
   writeDS1820(0xCC);  //Skip ROM
   writeDS1820(0xBE);  //Read Scratchpad
   readtemperature();
   showtemperature();
}
main()
{
serinit(9600);
RCAP2L=0x00;
RCAP2H=0xDC;
setbit(T2CON.2) //AUTO RELOADED//
setbit(IE.5)
P1=0;
P0=0;
setbit(P1.4)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
setbit(P1.5)
val=7;
n=35;
neem=9; //Small Step
count=count2=i=x=timer1=timer2=tickkey=veccam=0;
tick=tick2=tick3=navr=navl=flag=save=ccam=timerc=0;
whilc(1)
{
        pollkey();
        if((key=='b')||(ccam!=0))
        set_cam();        //Command Cam to Center
        if(key!=-1)
        move();           //Control 4 DC Motors & 2 Stepping Motors
//      if(key=='n')
//      chk_distance(); // Check Distance from Robot to Object in Front
        chk_delay();      //Check Pulse Width Modulation
        if(key=='f')
        cplbit(P1.4)      //Complement Bit SuperBright LED
        if(key=='g')
        cplbit(P1.5)      //Complement Bit Bird Beep
        if(key=='y')
        count=count2=0;           // Clear Distance Variable
        setinfared();     // Set Infared Sensor --> Ready
        chk_sta();        //Check Left Wheel Infared Sensor
        chk_sta2();       //Check Right Wheel Infared Sensor
        if(key=='u')
        ds1820();         // Convert&Read Temperature--> Print Temperature to terminal
        if(key=='h')
        printf("%dp%dj%da",count2,count,veccam);
}
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้