สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบควบคุมหุ่นยนต์เคลื่อนที่ผ่านคอมพิวเตอร์

COMPUTER CONTROL FOR MOBILE ROBOT

นายเจียร์ จตุรานน
นายนรินทร์ รูปพรหม
นายสัมฤทธิ์ สันติธารา

เลขหมู่.....
เลขทะเบียน 62700
วัน,เดือน,ปี 21 ส.ค. 2549

b 11b28k01
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
ภาควิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COMPUTER CONTROL FOR MOBILE ROBOT

JIAR JATURANON

NARIN ROOPPROM

SAMRIT SANTITHARA

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2005

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาควิชาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

- -

หัวข้อปริญญานิพนธ์ ระบบควบคุมหุ่นยนต์เคลื่อนที่ผ่านคอมพิวเตอร์

COMPUTER CONTROL FOR MOBILE ROBOT

นักศึกษาผู้จัดทำ	นายเจียร์	จตุรานน	รหัสประจำตัว 45010146
	นายนรินทร์	รูปพรหม	รหัสประจำตัว 45010378
	นายสัมฤทธิ์	สันติธารา	รหัสประจำตัว 45010818

ปริญญา วิศวกรรมศาสตรบัณฑิต

สาขาวิชา วิศวกรรมการวัดคุม

ปีการศึกษา 2548

อาจารย์ผู้ควบคุมปริญญานิพนธ์	**ลายมือชื่อ**
รศ.สุพรรณ กุลพาณิชย์	

ภาควิชารับรองแล้ว

(รศ.ประสิทธิ์ จุลเสรีวงศ์)

หัวหน้าภาควิชาวิศวกรรมการวัดคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	ระบบควบคุมหุ่นยนต์เคลื่อนที่ผ่านคอมพิวเตอร์		
	COMPUTER CONTROL FOR MOBILE ROBOT		
นักศึกษาผู้จัดทำ	นายเจียร์	จตุรานน	รหัสประจำตัว 45010146
	นายนรินทร์	รูปพรหม	รหัสประจำตัว 45010378
	นายสัมฤทธิ์	สันติธารา	รหัสประจำตัว 45010818
อาจารย์ที่ปรึกษา	รศ.สุพรรณ	กุลพาณิชย์	
ปีการศึกษา	2548		

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้นำเสนอแนวความคิดในการพัฒนาโปรแกรมสั่งการเคลื่อนที่หุ่นยนต์ จากเดิมผู้ใช้งานจะต้องสั่งการเคลื่อนที่ให้หุ่นยนต์ตลอดเวลาซึ่งมีความแม่นยำในการสั่งการเคลื่อนที่น้อยและสิ้นเปลืองเวลาในการทำงาน เพื่อลดปัญหาที่กล่าวมาข้างต้นจึงพัฒนาโปรแกรมที่ผู้ใช้งานสามารถกำหนดเส้นทางเดินล่วงหน้าโดยผ่านเครือข่ายไร้สายและสามารถตรวจสอบผลการเคลื่อนที่ผ่านกล้อง webcam คณะผู้จัดทำหวังว่าจะสามารถนำโปรแกรมไปใช้ในการทำงานที่ต้องการให้หุ่นยนต์เคลื่อนที่ตามเส้นทางที่กำหนดได้อย่างอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Computer Control For Mobile Robot	
Authors	Mr. Jiar	Jaturanon
	Mr. Narin	Roopprom
	Mr. Samrit	Santithara
Thesis Advisor	Assoc.Prof. Suparn	Kullapanich
Year	2005	

ABSTRACT

This thesis performs a developement of aprogram which is design for controlling mobile robot .The program was developed to solve the problem that user must give commands to robot continuously in order to make a robot operate certainly which waste the time and lack of precision . The developed program helps the mobile robot to move in the route that user has set before automatically and can be used via wireless lan network . Moreover , user can check if the mobile robot can reach the destination or not by using the webcam which is installed on mobile robot .

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สามารถสำเร็จลุล่วงไปได้ด้วยดี เนื่องด้วยได้รับความอนุเคราะห์และการช่วยเหลือในทุก ๆ ด้านจากอาจารย์ที่ปรึกษา จึงขอขอบพระคุณ รศ.สุพรรณ กุลพาณิชย์ และ ผศ.ทวีพล สัตย์ซื่อ รวมทั้งอาจารย์ภาควิชาวิศวกรรมการวัดคุมทุกท่านที่ประสิทธ์ประสาทวิชาให้ความรู้และคำแนะนำอันเป็นประโยชน์ต่อการทำปริญญานิพนธ์ฉบับนี้

ข้าพเจ้าขอกราบขอบพระคุณ คุณพ่อ, คุณแม่ที่สนับสนุนและเป็นกำลังใจให้ข้าพเจ้าเสมอมา สิ่งศักดิ์สิทธิ์ที่ข้าพเจ้าเคารพนับถือที่คอยปกป้องดูแลรักษาข้าพเจ้า

ขอขอบคุณคณะอาจารย์ทุก ๆ ท่าน, รุ่นพี่, เพื่อน ๆ และรุ่นน้องทุกคนที่ให้การช่วยเหลือและข้อเสนอแนะในการทำงาน ข้อเสนอแนะเหล่านี้ทำให้งานวิจัยมีความสมบูรณ์มากขึ้น

คุณค่าและประโยชน์อันพึงมีจากปริญญานิพนธ์ฉบับนี้ ผู้วิจัยขอมอบให้ผู้มีพระคุณทุกท่าน

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญตาราง.....VII

สารบัญรูป.....VIII

บทที่ 1 บทนำ.....1

1.1 ความเป็นมาและเหตุจูงใจของการวิจัย.....1

1.2 วัตถุประสงค์ของปริญญานิพนธ์.....1

1.3 ขอบเขตของปริญญานิพนธ์.....2

1.4 ขั้นตอนการศึกษา.....2

บทที่ 2 ทฤษฎีและหลักการ.....3

2.1 ความรู้ทั่วไปเกี่ยวกับหุ่นยนต์.....3

2.1.1 แบ่งตามลักษณะการควบคุม(Controller).....3

2.1.2 แบ่งตามระดับความสามารถ (Capability Level).....4

2.2 ไมโครคอนโทรลเลอร์ MCS-51.....5

2.2.1 คุณสมบัติทางเทคนิคที่สำคัญของ P89C51RD2.....6

2.2.2 พอร์ทอินพุต/เอาต์พุต.....11

2.2.3 รีจิสเตอร์สำหรับใช้งานทั่วไปใน MCS-51.....12

2.2.4 ไทเมอร์/เคาน์เตอร์.....12

2.2.5 การอินเตอร์รัปต์.....16

2.2.6 สื่อสารข้อมูลผ่านพอร์ทอนุกรมของไมโครคอนโทรลเลอร์ MCS-51.....20

2.2.7 หลักการทำงานและคุณสมบัติของระบบบัส I^2C.....22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.3 เซอร์โวมอเตอร์ (Servo motor) 30

2.3.1 หลักการทำงานของเซอร์โวมอเตอร์ 31

2.4 โมดูลเข็มทิศดิจิตอล(Digital Compass Module).....34

2.4.1 คุณสมบัติ.....34

2.4.2 ตำแหน่งขาและการต่อใช้งาน.....34

2.4.3 การปรับแต่งค่าของโมดูล CMPS03 ผ่านทางระบบบัส I²C 35

2.5 เครือข่ายไวร์เลสแลน (Wireless Lan).....36

2.5.1 เครือข่ายไร้สายตามมาตรฐาน IEEE 802.11.....37

2.5.2 อุปกรณ์ที่ใช้ในเครือข่ายไร้สาย.....37

2.5.3 รูปแบบการเชื่อมต่อต่างๆของเครือข่ายไร้สาย.....39

2.5.4 ลักษณะเครือข่ายไร้สายที่ใช้ในการทดลองหุ่นยนต์เคลื่อนที่.....40

2.5.5 การโปรแกรมคอมพิวเตอร์.....41

2.5.6 พื้นฐานต่างๆของการเชื่อมโยง TCP.....41

2.5.7 Winsock Control.....42

บทที่ 3 การออกแบบและการสร้าง.....44

3.1 ส่วนประกอบระบบของหุ่นยนต์เคลื่อนที่อัตโนมัติ.....44

3.2 ไมโครคอนโทรลเลอร์ 44

3.3 ส่วนวัดระยะทาง.....45

3.4 ส่วนวัดทิศทาง.....48

3.4.1 การอ่านค่าทิศทางเป็นข้อมูลดิจิตอลผ่านระบบบัส I²C.....48

3.4.2 รูปแบบการการสื่อสารข้อมูลบัส I²C 48

3.4.3 ลำดับขั้นการติดต่อ.....48

3.5 การทำงานของ VB.....49

3.5.1 คอนโทรล Winsock.....49

3.5.2 การทำงานของเครื่องแม่ข่าย.....49

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

3.5.3 การติดต่อกับเครื่องลูกข่าย....49
3.5.4 การติดต่อกับไมโครคอนโทรลเลอร์....55
3.5.5 การทำงานของเครื่องลูกข่าย....57
3.5.6 โปรแกรมการทำงาน....57

บทที่ 4 ผลการทดลอง....64
4.1 การติดต่อสื่อสารระหว่างคอมพิวเตอร์กับหุ่นยนต์สำรวจ....64
4.2 การทำงานของโครงงาน....64
4.2.1 การทำงานโหมด Control....65
4.2.2 การทำงานโหมด Observe....67
4.3 การทดลอง....68
4.3.1 การทดสอบความสามารถในการเคลื่อนที่....68
4.4 ข้อมูลจำเพาะของหุ่นยนต์เคลื่อนที่....69

บทที่ 5 สรุปผล ปัญหาและข้อเสนอแนะ....71
5.1 สรุปผล....71
5.2 ปัญหา....71
5.3 ข้อเสนอแนะ....72

บรรณานุกรม....73
ภาคผนวก โปรแกรมควบคุมการทำงานในส่วนของไมโครคอนโทรลเลอร์....74

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ **หน้า**

2.1 แสดงรายละเอียดขั้นต้นของขาต่อใช้งานของ P89C51RD2BN.....7

2.2 ความสัมพันธ์ของรีจิสเตอร์ IP และ IPH ในการกำหนดระดับของ
ลำดับความสำคัญในการตอบสนองการอินเตอร์รัปต์.....I9

2.3 แสดงตำแหน่งรีจิสเตอร์ของโมดูลเข็มทิศ.....36

4.1 แสดงการเคลื่อนที่ในโหมดAutomatic.....68

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ **หน้า**

2.1 แสดงขาต่างๆของ MCS-51 เบอร์P89C51RD2....11
2.2 รีจิสเตอร์ใช้งานเฉพาะ TCON....14
2.3 รีจิสเตอร์ใช้งานเฉพาะ TMOD....15
2.4 รายละเอียดของรีจิสเตอร์ IE ซึ่งใช้ในการเอ็นเอเบิลการอินเตอร์รัปต์....17
2.5 รายละเอียดของรีจิสเตอร์ IP ซึ่งใช้ในการกำหนดลำดับความสำคัญของ
แหล่งกำเนิดสัญญาณอินเตอร์รัปต์....18
2.6 รายละเอียดของรีจิสเตอร์ IPH ซึ่งใช้กำหนดลำดับความสำคัญสูงสุดให้แก่
แหล่งกำเนิดสัญญาณอินเตอร์รัปต์....19
2.7 แสดงรีจิสเตอร์ที่ใช้งานเฉพาะ SCON....20
2.8 สภาวะที่เกิดบนบัส I^2C....23
2.9 Acknowledge บนสาย I^2C....24
2.10 การส่งข้อมูลแบบ 7 บิตของอุปกรณ์มาสเตอร์....25
2.11 การอ่านสัญญาณ Acknowledge ของอุปกรณ์มาสเตอร์แบบ 7 บิต....25
2.12 การส่งข้อมูลแบบ 10 บิตของอุปกรณ์มาสเตอร์....26
2.13 การอ่านสัญญาณ Acknowledge ของอุปกรณ์มาสเตอร์แบบ 10 บิต....26
2.14 แสดงเซอร์โวมอเตอร์....30
2.15 แสดงส่วนประกอบต่างๆของเซอร์โวมอเตอร์....31
2.16 แสดงชิ้นส่วนต่างๆของเซอร์โวมอเตอร์....31
2.17 แสดงความกว้างของสัญญาณพัลส์ที่เวลาต่างๆ....32
2.18 แสดงชิ้นส่วนต่างๆของเซอร์โวมอเตอร์อย่างละเอียด....33
2.19 แสดงบอร์ดโมดูลเข็มทิศ....35
2.20 Access Point Mode....39
2.21 Repeater Mode....39
2.22 Wirelessbridge Mode....39
2.23 Multiportbridge Mode....40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

2.24 Wireless Client Mode....40

2.25 แสดงอุปกรณ์เข้าใช้งานเครือข่าย (Access point) ยี่ห้อ Linksys รุ่น WRT54G....40

2.26 แสดงการติดตั้งคอมพิวเตอร์แม่ข่ายกับตัวหุ่นยนต์....41

2.27 การสร้าง Client/Server โดยใช้ Winsock....43

3.1 จานเข้ารหัส (Encoder)....46

3.2 การติดจานเข้ารหัสเข้ากับเพลารถ....46

3.3 วงจรโฟโตทรานซิสเตอร์....47

3.4 โฟโตทรานซิสเตอร์ (Photo Transistor)....47

3.5 แสดงไทมิ่งไดอะแกรมของการติดต่อสื่อสารกับโมดูล CMPS03 ผ่านระบบบัส I²C....48

3.6 โฟลว์ชาร์ตแสดงการทำงานของ เมื่อเครื่องแม่ข่ายได้รับการร้องขอการติดต่อ จากเครื่องลูกข่าย....50

3.7 โฟลว์ชาร์ตแสดงการทำงานของ เครื่องแม่ข่ายเมื่อได้รับข้อมูลจากเครื่องลูกข่าย....53

3.8 โฟลว์ชาร์ตแสดงการทำงานของ Automatic Mode....56

3.9 แสดงหน้าจอ Disconnected....58

3.10 แสดงหน้าจอ Connected....59

3.11 แสดงหน้าจอ Observed Mode....60

3.12 แสดงหน้าจอ Control Mode....61

3.13 แสดงหน้าจอ Automatic Mode....62

3.14 แสดงหน้าจอ Scale....63

4.1 แสดงหน้าจอการติดต่อและเลือกโหมดการทำงานของเครื่องลูกข่าย (Client)....64

4.2 แสดงหน้าจอในโหมด Control....65

4.3 แสดงหน้าจอการทำงานในระบบ Automatic....66

4.4 แสดงหน้าจอในโหมด Automatic เมื่อสั่งให้ทำงาน....66

4.5 แสดงหน้าจอในโหมดสังเกตการณ์ (Observe)....67

4.6 แสดงรายละเอียดโครงสร้างของหุ่นยนต์....70

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและเหตุจูงใจในการวิจัย

ในปัจจุบันจะเห็นได้ว่าเทคโนโลยีรอบ ๆ ตัวเราได้มีการพัฒนาไปหลาย ๆ ด้าน ไม่ว่าจะเป็นด้าน คอมพิวเตอร์ ยานยนตร์ อากาศยาน เทคโนโลยีสารสนเทศ เทคโนโลยีหุ่นยนต์ ฯลฯ เทคโนโลยีหุ่นยนต์ถือได้ว่าเป็นการรวมศาสตร์และเทคโนโลยีแขนงต่าง ๆ เอาไว้รวมกัน ไม่ว่าจะเป็นด้านเครื่องกล คอมพิวเตอร์ อุปกรณ์และวงจรอิเล็กโทรนิค การติดต่อสื่อสาร

เทคโนโลยีหุ่นยนต์ในปัจจุบันนี้ได้มีการนำมาปฏิบัติงานต่าง ๆ แทนมนุษย์ในหลาย ๆ ด้าน เช่น ใช้หุ่นยนต์ทำหน้าที่เฝ้าบ้านแทนสุนัขซึ่งสามารถทำงานได้ดีกว่าสุนัขปรกติเพราะไม่ต้องให้อาหารอีกทั้งยังสามารถติดต่อสื่อสารกับเจ้าของได้จากระยะทางไกล นอกจากนี้ยังมีการใช้หุ่นยนต์ในโรงงานอุตสาหกรรม โดยมีการนำหุ่นยนต์มาใช้ ตรวจสอบสินค้า คัดแยกสินค้า หรือนับจำนวนสินค้าผ่านสายการผลิต และอื่นๆ อีกมาก ซึ่งล้วนแล้วแต่เป็นงานที่ง่ายแต่มีโอกาสเกิดความผิดพลาดได้ง่ายถ้าใช้มนุษย์ จึงมีการนำหุ่นยนต์เข้ามาแทนที่ ซึ่งสามารถช่วยลดต้นทุนและเพิ่มผลผลิต อีกทั้งยังสามารถช่วยลดจำนวนผู้ประสบอุบัติเหตุจากการทำงานได้อีกด้วย และเทคโนโลยีหุ่นยนต์ยังสามารถนำไปใช้สำรวจในพื้นที่เสี่ยงอันตรายต่าง ๆ เช่น ในอวกาศ ใต้ทะเลลึก ในบริเวณที่มีความร้อนสูง บริเวณที่มีสารกัมมันตภาพรังสีหรือสารเคมีมีพิษต่าง ๆ ซึ่งไม่เหมาะต่อการใช้มนุษย์เข้าไปสำรวจ

โครงงานนี้เป็นโครงงานที่พัฒนามาจากโครงงาน หุ่นยนต์เคลื่อนที่ (MOBILE ROBOT) ของบัณฑิตวิศวกรรมศาสตร์สาขาวิชาวิศวกรรมการวัดคุม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2547 โดยมีการพัฒนาให้หุ่นมีความสามารถมากยิ่งขึ้น ซึ่งยังคงไว้ซึ่งวัตถุประสงค์เดิมของโครงงานก่อนซึ่งก็คือการใช้หุ่นยนต์แทนมนุษย์และสามารถนำไปประยุกต์ใช้ในอุตสาหกรรมได้

1.2 วัตถุประสงค์ของปริญญานิพนธ์

โครงงานนี้ได้มีการพัฒนาให้หุ่นยนต์มีประสิทธิภาพสูงยิ่งขึ้น จากเดิมที่ควบคุมหุ่นผ่านคลื่นวิทยุซึ่งมีระยะการทำงานที่สั้นมาเป็นเครือข่ายไร้สายซึ่งสามารถครอบคลุมพื้นที่การทำงานได้มากกว่าและยังมีอัตรารับส่งข้อมูลที่สูงกว่าด้วย อีกทั้งยังเพิ่มในการกำหนดเส้นทางล่วงหน้าและเคลื่อนที่ตามเส้นทางนั้นจนถึงจุดหมายปลายทาง และยังส่งเส้นทางการเคลื่อนที่ที่รถได้เคลื่อนที่ผ่านไปแล้วกลับมายังผู้ควบคุมได้อีกด้วย ซึ่งนอกจากจะสามารถนำไปประยุกต์ใช้กับการสำรวจแล้วยังสามารถนำไปประยุกต์ใช้ในอุตสาหกรรม เช่น การขนส่งลำเลียง เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของปริญญานิพนธ์

ปริญญานิพนธ์นี้จะกล่าวถึง การควบคุมและสั่งงานหุ่นยนต์สำรวจประเภท Mobile Robot จากเครื่องคอมพิวเตอร์ผ่านระบบเครือข่ายไร้สาย โดยการควบคุมผ่านกล้อง Webcam ที่ติดตั้งอยู่ที่ตัวหุ่น หรือจะเป็นการกำหนดเส้นทางเดินให้ตัวหุ่นล่วงหน้า จากนั้นจะเป็นขั้นตอนการรับ-ส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์ที่อยู่ในระบบเครือข่าย และส่งข้อมูลให้ตัวหุ่นผ่านพอร์ตอนุกรม RS-232 ซึ่งภายในตัวหุ่นยนต์สำรวจจะมีไมโครคอนโทรลเลอร์ (MCS-51) คอยทำหน้าที่รับข้อมูลที่ส่งมาแล้วทำการประมวลผลเป็นคำสั่งควบคุมการเคลื่อนที่ หรือคำสั่งควบคุมกล้อง และส่งข้อมูลที่ตรวจจับได้กลับไปยังเครื่องคอมพิวเตอร์

1.4 ขั้นตอนการศึกษา

สำหรับการทำปริญญานิพนธ์นี้จะเริ่มจากการศึกษา การรับ-ส่งข้อมูลผ่านเครือข่ายไร้สาย จากนั้นทำการศึกษาโครงสร้างและการทำงานของไมโครคอนโทรเลอร์ MCS-51 และโปรแกรม Visual Basic Version 6.0 เพื่อนำไปใช้ในการเขียนโปรแกรมควบคุมการทำงานของหุ่นยนต์ผ่านพอร์ตอนุกรม RS-232 การรับส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์ที่อยู่ในระบบ กำหนดเส้นทางที่ตัวหุ่นจะต้องเคลื่อนที่ไป หลังจากนั้นจะทำการศึกษาส่วนต่าง ๆ ที่จะใช้เป็นองค์ประกอบเสริมในการทำงานของหุ่นยนต์ เช่น การทำงานของการถ่ายภาพรอบ ๆ ตัวหุ่นยนต์จากกล้อง Webcam และการทำงานของเซอร์โวมอเตอร์ที่ใช้หมุนกล้องและเพิ่มมุมมองให้กว้างขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

2.1 ความรู้ทั่วไปเกี่ยวกับหุ่นยนต์

หุ่นยนต์ถือว่าเป็นสิ่งที่ได้รับความสนใจเป็นอย่างมากในปัจจุบัน ดังจะเห็นได้จากการพัฒนาหุ่นยนต์เลียนแบบสัตว์เลี้ยง หรือกระทั่งหุ่นยนต์เลียนแบบมนุษย์ "ASIMO" หุ่นยนต์ดังที่ได้กล่าวมาแล้วนั้นเป็นรูปแบบหนึ่งของหุ่นยนต์ ซึ่งสามารถแบ่งออกได้หลายชนิดซึ่งในปริญญานิพนธ์นี้ขอกล่าวถึง 2 ลักษณะ คือ แบ่งตามลักษณะการควบคุมและแบ่งตามระดับความสามารถ ดังนี้

2.1.1 แบ่งตามลักษณะการควบคุม (Controller)

2.1.1.1 หุ่นยนต์แบบเดินจุดต่อจุด (Point to Point robots)

หุ่นยนต์ประเภทนี้สามารถเคลื่อนที่จากจุดที่ระบุไว้ได้แน่นอน โดยเคลื่อนที่จากจุดหนึ่งไปจุดหนึ่ง แต่ไม่สามารถหยุดที่จุดอื่นที่ไม่ได้กำหนดไว้ก่อน เป็นหุ่นยนต์แบบง่ายที่สุดและมีราคาถูกที่สุด ลักษณะการนำไปใช้งานของหุ่นยนต์แบบนี้ก็คือ หุ่นยนต์ที่ใช้เดินจากจุดไปอีกจุด

2.1.1.2 หุ่นยนต์แบบเดินทางต่อเนื่อง (Continuous-Path robots)

หุ่นยนต์ประเภทนี้สามารถหยุดที่ตำแหน่งใด ๆ ตามทางได้ และการเคลื่อนที่ของหุ่นยนต์ประเภทนี้จะเป็นเส้นตรง หรือเส้นโค้งที่แน่นอน การเคลื่อนที่ของแต่ละแกนจะเกิดขึ้นพร้อมกันแต่จะมีความเร็วต่างกัน ลักษณะการนำไปใช้งานของหุ่นยนต์แบบนี้คือ หุ่นยนต์ที่ใช้ในการเชื่อมอาร์ค

2.1.1.3 หุ่นยนต์ที่สามารถสร้างแนวการเคลื่อนที่ได้เอง (Computed Trajectory robots)

อุปกรณ์ควบคุมหุ่นยนต์แบบนี้สามารถสร้างแนวการเคลื่อนที่ ซึ่งอาจเป็นเส้นตรงหรือวงกลม เส้นโค้งหรืออื่น ๆ ด้วยความแม่นยำสูง ในหุ่นยนต์เส้นทางการเดินสามารถกำหนดได้ในเทอมเรขาคณิต หรือสูตรทางพีชคณิต โดยส่วนที่ควบคุมต้องการเพียงตำแหน่งเริ่มต้นตำแหน่งสุดท้าย และคำจำกัดความของการเคลื่อนที่เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1.4 หุ่นยนต์แบบเซอร์โว และนอนเซอร์โว (Servo and Non-Servo robots)

หุ่นยนต์ที่ควบคุมแบบเซอร์โวหมายถึง หุ่นยนต์ที่มีการตรวจจับตำแหน่งของตัวมันเอง และมีการป้อนกลับตำแน่งที่ตรวจจับเพื่อนำไปใช้ในการควบคุม ส่วนหุ่นยนต์แบบนอนเซอร์โวจะไม่มีทางรู้ได้ว่าอยู่ที่ตำแหน่งที่กำหนดหรือยัง หุ่นยนต์ที่สามารถเคลื่อนที่ได้จะต้องมีความสามารถในการป้อนกลับซึ่งทำให้มันเดินทางได้อย่างถูกต้อง เพื่อที่จะมีการเคลื่อนที่ตามที่กำหนดไว้

2.1.2 แบ่งตามระดับความสามารถ (Capability Level)

ระดับความสามารถและการยืดหยุ่นในการควบคุม เป็นเกณฑ์อีกอันหนึ่งสำหรับการอธิบายเกี่ยวกับหุ่นยนต์ ซึ่งจะแบ่งออกเป็น 5 ระดับดังนี้

2.1.2.1 เครื่องจักรที่ทำงานเป็นระดับขั้นตอน (Sequence- Controlled machine)

เครื่องจักรจะทำงานเป็นลำดับขั้นตอนตามขั้นตอนตามคำสั่งที่ป้อนไว้ ตัวอย่างเช่น เครื่องซักผ้าอัตโนมัติ ซึ่งจะทำงานเป็นลำดับขั้นคือ ใส่น้ำ ใส่ผงซักฟอก ซัก ใส่น้ำยาปรับผ้านุ่ม และอบแห้งโดยเครื่องตั้งเวลากำหนดไว้ ส่วนหุ่นยนต์ก็จะสามารถปรับเปลี่ยนเวลา และเปลี่ยนแปลงลำดับขั้นการทำงานได้ แต่ที่เรียกเครื่องจักรแบบนี้ว่าหุ่นยนต์ก็เพราะว่ามันมีแขนกลซึ่งถูกควบคุมด้วยเครื่องควบคุม

2.1.2.2 เครื่องจักรที่ทำตามได้ (Playback Machines)

เครื่องจักรที่ทำตามได้ หมายถึง เครื่องจักรที่สามารถ "สอน" ให้ทำการเคลื่อนที่เป็นตอน (series of movements) โดยมีอุปกรณ์บันทึก เช่น เทปแม่เหล็ก (magnetic tape) จานแม่เหล็ก (magnetic disk) แรม (RAM) เพื่อบันทึกจุดหรือขั้นตอนที่เคลื่อนที่ไป โดยไปบันทึกโคออร์ดิเนตจากอุปกรณ์ตรวจจับตามตำแหน่งหลังจากเดิน

2.1.2.3 หุ่นยนต์ที่สามารถแสดงแนวเคลื่อนที่ได้เอง (Computer Trajectory Robots)

หุ่นยนต์ที่สามารถแสดงแนวเคลื่อนที่ได้เอง สามารถถูกโปรแกรมให้เป็นไปตามเส้นทางที่แน่นอน โดยไม่จำเป็นต้องสอน ผู้ใช้สามารถกำหนดจุดและแนวทางซึ่งมันก็จะหาแนวทางการเคลื่อนที่ได้เอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2.4 หุ่นยนต์ที่สามารถเปลี่ยนแปลงให้เหมาะสมได้ (Adaptive Robots)

หุ่นยนต์จะมีการควบคุมด้วยคอมพิวเตอร์และมีการป้อนกลับ หุ่นยนต์เหล่านี้ส่วนมากจะมีโปรแกรมการเดินทางอยู่แล้ว แต่ยังสามารถปรับปรุงทางเดินและการทำงานที่มันทำได้ เช่น หุ่นยนต์เชื่อม สามารถไปตามแนวเชื่อมที่แตกต่างไปจากทางเดินที่กำหนด ในลักษณะนี้อุปกรณ์ตรวจจับจะรวบรวมข้อมูลให้คอมพิวเตอร์ปรับปรุงแนวทางการเชื่อมได้

2.1.2.5 หุ่นยนต์สมองเทียม (Intelligent Robots)

เป็นหุ่นยนต์ที่ระดับสูงที่สุด มีความสามารถในการรับรู้สิ่งแวดล้อมตรวจสอบสภาพแวดล้อมและปรับปรุงการกระทำต่าง ๆ ให้เหมาะสม โดยมันจะมีอุปกรณ์ตรวจจับข้อมูลอยู่พร้อม มีหน่วยความจำที่มากพอและมีแนวทางที่จะเป็นแบบอย่างในการทำงานมากอีกด้วย

2.2 ไมโครคอนโทรลเลอร์ MCS-51

MCS-51 เป็นไมโครคอนโทรลเลอร์ลักษณะแบบชิพเดี่ยว ที่มีข้อดีเมื่อนำมาเปรียบเทียบกับไมโครโปรเซสเซอร์ขนาด 8 บิตตระกูลอื่น ๆ ดังนี้

1. มีหน่วยความจำสำหรับเก็บข้อมูลทั่วไป (RAM) บรรจุไว้ภายใน 128-256 ไบต์
2. มีหน่วยความจำสำหรับเก็บโปรแกรมควบคุมการทำงานอยู่ภายในจำนวน 4 กิโลไบต์
3. มีวงจรตั้งเวลาวงจรนับขนาด 16 บิต 2 ตัวอยู่ภายใน
4. มีวงจรรับส่งข้อมูลอนุกรม 2 ทิศทาง
5. มีสัญญาณนาฬิกาภายในตัว
6. มีพอร์ทที่สามารถรับหรือส่งข้อมูลได้ 2 ทิศทาง จำนวน 4 พอร์ท พอร์ทละ 8 บิต

นอกจากนี้ MCS-51 ยังมีคุณสมบัติอื่น ๆ ที่น่าสนใจ คือ

- ต้องการแหล่งจ่ายไฟ 5 โวลต์เพียงชุดเดียว
- มีหน่วยความจำสำหรับเก็บโปรแกรมควบคุมการทำงานอยู่ภายในชิพ
- สามารถที่จะใช้หน่วยความจำสำหรับเก็บโปรแกรม และข้อมูลที่อยู่ภายนอกชิพได้อย่างละ 64 กิโลไบต์
- มีคำสั่งคูณและหารขนาด 8 บิตในตัวเอง
- จัดระดับความสำคัญของสัญญาณอินเตอร์รัปต์ได้ 2 ระดับ
- รับและส่งข้อมูลแบบอนุกรมได้ในตัว โดยที่สามารถกำหนดอัตราเร็วในการรับและส่งข้อมูลได้ตั้งแต่ 300 – 375 กิโลบิตต่อวินาที
- สามารถประมวลผลแบบบูลีนเพื่อใช้ในงานควบคุมโดยเฉพาะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มีรีจิสเตอร์สำหรับใช้งานเป็นไทม์เมอร์หรือเคาร์เตอร์ เพื่อที่จะใช้นับจำนวนสัญญาณนาฬิกาในชิพ หรือนับการเปลี่ยนสถานะของสัญญาณภายนอกขนาด 16 บิต จำนวน 2 ตัว เพื่อใช้สำหรับนับจำนวนพัลส์ วัดความกว้างของพัลส์หรือใช้วัดช่วงเวลา

- หน่วยความจำสำหรับเก็บข้อมูลภายในบางส่วนนั้น สามารถเข้าถึงข้อมูลได้ทั้งในระดับไบต์และบิตเพื่อให้การออกแบบโปรแกรมและการควบคุมระบบการทำงานได้ง่ายขึ้น

2.2.1 คุณสมบัติทางเทคนิคที่สำคัญของ P89C51RD2

เป็นไมโครคอนโทรเลอร์ตระกูล MCS-51 ซึ่งมีหน่วยความจำภายในโดยที่เป็นแบบ แฟลช (flash memory) ของ Philips Semiconductor ในอนุกรม P89C51RD2 ซึ่งมีด้วยกันสองเบอร์คือ P89C51RD2-HBP และ P89C51RD2BN แต่เราจะอ้างอิงถึงเบอร์ P89C51RD2BN มีคุณสมบัติที่โดดเด่นดังนี้

- เป็นไมโครคอนโทรลเลอร์ 8 บิต ที่เข้ากันได้กับไมโครคอนโทรลเลอร์ MCS-51 พื้นฐาน

- หน่วยความจำภายในตัวไมโครคอนโทรลเลอร์เป็นแบบแฟลช ทำให้สามารถลบและเขียนใหม่ได้ถึง 10,000 ครั้ง จึงสามารถใช้งานในรูปแบบไมโครคอนโทรลเลอร์ชิปเดี่ยวโดยไม่ต้องใช้หน่วยความจำภายนอก ส่งผลให้สามารถใช้งานพอร์ทอินพุตเอาต์พุตของไมโครคอนโทรลเลอร์ได้อย่างเต็มประสิทธิภาพ ขนาดของหน่วยความโปรแกรมสูงถึง 64 กิโลไบต์

- หน่วยความจำข้อมูลแรมภายในมีขนาด 1 กิโลไบต์

- สามารถโปรแกรมข้อมูลลงไปในหน่วยความจำโปรแกรมแบบในวงจรหรือในระบบ (ISP : In-system programming)

- ความถี่สัญญาณนาฬิกาสูงสุด 33 MHz ในกรณีที่ทำงานด้วยสัญญาณนาฬิกาภายใน 12 ลูกต่อแมชีนไซเกิลและ 20 MHz กรณีทำงานด้วยสัญญาณนาฬิกาภายใน 6 ลูกต่อแมชีนไซเกิล

P89C51RD2BN ได้รับการกำหนดให้ทำงานเบื้องต้นในโหมดสัญญาณนาฬิกา 12 ลูกต่อแมชีนไซเกิล สามารถเลือกเปลี่ยนเป็น 6 สัญญาณนาฬิกาต่อแมชีนไซเกิลได้ โดยสามารถเปลี่ยนกลับไปกลับมาได้ แตกต่างจาก P89C51RD2HBP ที่ค่าเบื้องเป็น 6 สัญญาณนาฬิกาต่อแมชีนไซเกิล เมื่อเปลี่ยนเป็น 12 สัญญาณนาฬิกาแล้วจะเปลี่ยนกลับไปกลับมาอีกไม่ได้

- ขาพอร์ต 8 บิต 4 พอร์ต แบบกึ่ง 2ทิศทาง (quasi-bidirectional) เป็นได้ทั้งอินพุตและเอาต์พุต

- อุปกรณ์เพอริเฟอรัลที่อยู่ภายในไมโครคอนโทรลเลอร์สามารถทำงานได้ด้วยความเร็ว 12 สัญญาณนาฬิกาต่อแมชีนไซเกิลได้แม้ว่าซีพียูจะทำงานด้วยความเร็ว 6 สัญญาณนาฬิกาภายในต่อแมชีนไซเกิลเป็นคุณสมบัติที่เพิ่มเติมเข้ามาในเบอร์ P89C51RD2BN

- มีวงจรสื่อสารอนุกรมแบบฟูลดูเพล็กซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ไทเมอร์/เคาน์เตอร์ขนาด 16 บิต 3ตัว (ไทเมอร์0 , 1 และ 2)
- มีรีจิสเตอร์ตัวชี้ตำแหน่งข้อมูลหรือ DPTR 2 ตัว
- สามารถรองรับแหล่งกำเนิดอินเตอร์รัปต์ได้ 7 ประเภท
- กำหนดนัยสำคัญของการตอบสนองอินเตอร์รัปต์ได้ 4 ระดับ
- สามารถติดต่อหน่วยความจำภายนอกได้สูงสุด 64 กิโลไบต์
- มีวอตช์ดีอกไทเมอร์
- มีโมดูลวงจรนับโปรแกรมได้ (PCA : Programmable Counter Array) ซึ่งบรรจุวงจรตรวจจับสัญญาณ (capture) เปรียบเทียบสัญญาณ (compare) วงจรมอดูเลชั่นทางความกว้างพัลส์ (PWM) และวอตช์ด็อกไทเมอร์ (watchdog timer)

ตารางที่ 2.1 แสดงรายละเอียดขั้นต้นของขาต่อใช้งานของ P89C51RD2BN

ชื่อขา	ขาที่	ชนิด	หน้าที่และการทำงาน
Vcc	40	อินพุต	ต่อไฟเลี้ยง +5V
GND	20	อินพุต	ต่อกราวด์
P0.0-P0.7	39-32	อินพุต/เอาต์พุต	-ใช้งานเป็นขาพอร์ตอินพุตเอาต์พุต ถ้าต้องการกำหนดให้ขาพอร์ต 0 ขาใดขาหนึ่งเป็นอินพุต สามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ทำให้สถานะลอย ค่าอินพุตอิมพีแดนซ์สูง สามารถใช้งานเป็นขาพอร์ตอินพุตได้ -ใช้ในการติดต่อกับขาแอดเดรสไบต์ของหน่วยความจำภายนอก (A0-A7) และขาข้อมูล (D0-D7) โดยใช้การมัลติเพล็กซ์เข้าช่วยเพื่อสลับการทำงานให้เป็นได้ทั้งขาติดต่อแอดเดรสและขาข้อมูล ในการติดต่อกับหน่วยความจำภายนอก
P1.0-P1.7	1-8	อินพุต/เอาต์พุต	-ใช้งานเป็นขาพอร์ตอินพุตเอาต์พุตสำหรับใช้งานทั่วไป -เป็นขาสัญญาณของไทเมอร์ 2 และขาสัญญาณของโมดูล PCA ดังมีรายละเอียดต่อไปนี้ T2 (P1.0: ขา 1) เป็นขาอินพุตสำหรับนับค่าของไทเมอร์ 2 และขาเอาต์พุตสัญญาณนาฬิกาโปรแกรมแบบได้ T2EX (P 1.1: ขา2) เป็นขาอินพุตสำหรับควบคุมการทำงานของไทเมอร์/เคาน์เตอร์ 2 ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

			ECI (P1.2 : ขา 3) เป็นขาอินพุตสัญญาณนาฬิกาจากภายนอกสำหรับโมดูล PCA CEX0 (P1.3: ขา 4) เป็นขาอินพุตเอาต์พุตภายนอกของวงจรตรวจจับและเปรียบเทียบสัญญาณสำหรับ PCA โมดูล 0 CEX1 (P1.4: ขา 5) เป็นขาอินพุตเอาต์พุตภายนอกของวงจรตรวจจับและเปรียบเทียบสัญญาณสำหรับ PCA โมดูล 1 CEX2 (P1.5: ขา 6) เป็นขาอินพุตเอาต์พุตภายนอกของวงจรตรวจจับและเปรียบเทียบสัญญาณสำหรับ PCA โมดูล 2 CEX3 (P1.6: ขา 7) เป็นขาอินพุตเอาต์พุตภายนอกของวงจรตรวจจับและเปรียบเทียบสัญญาณสำหรับ PCA โมดูล 3 CEX4 (P1.7: ขา 5) เป็นขาอินพุตเอาต์พุตภายนอกของวงจรตรวจจับและเปรียบเทียบสัญญาณสำหรับ PCA โมดูล 4
P2.0- P2.7	21-28	อินพุต/ เอาต์พุต	-ใช้ขาเป็นพอร์ตอินพุตเอาต์พุตสำหรับใช้งานทั่วไป -ใช้ต่อกับขาแอดเดรสไบต์สูงของหน่วยความจำภายนอก (A8-A15) เมื่อติดต่อด้วย
P3.0- P3.7	10-17	อินพุต/ เอาต์พุต	-ใช้ขาเป็นพอร์ตอินพุตเอาต์พุตสำหรับใช้งานทั่วไป -ใช้งานเป็นขาพอร์ตหน้าที่พิเศษ ดังมีรายละเอียดต่อไปนี้ RxD (P3.0:ขา 10) ใช้เป็นขาอินพุตสำหรับรับข้อมูลจากการสื่อสารแบบอนุกรม TxD (P3.1:ขา 11) ใช้เป็นขาอินพุตสำหรับส่งข้อมูลจากการสื่อสารแบบอนุกรม INT0 (P3.2:ขา12) ใช้เป็นขาอินพุตสำหรับรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 0 INT1 (P3.3:ขา 13) ใช้เป็นขาอินพุตสำหรับรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 1 T0 (P3.4:ขา 14) ใช้เป็นขาอินพุตสำหรับรับสัญญาณไทเมอร์จากภายนอกช่อง 0 T0 (P3.5:ขา 15) ใช้เป็นขาอินพุตสำหรับรับสัญญาณไทเมอร์จากภายนอกช่อง 1 WR (P3.6:ขา 16) ใช้เป็นขาสัญญาณ WR ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก RD (P3.7:ขา 17) ใช้เป็นขาสัญญาณ RD ในกรณีที่ใช้เชื่อมต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

			กับหน่วยความจำภายนอก
RESET	9	อินพุต	ใช้ในการรีเซตการทำงานของไมโครคอนโทรลเลอร์ โดยในการป้อนสัญญาณลอจิก "1" ซึ่งจะต้องใช้เวลาอย่างน้อยเป็นเวลา 2 แมชีนไซเคิลโดยที่วงจรกำเนิดสัญญาณนาฬิกายังคงทำงานต่อเนื่องไปอย่างปกติ
ALE	30	เอาต์พุต	Address Latch Enable : เป็นขาควบคุมการแลตช์ขาพอร์ต 0 เมื่อมีการใช้งานหน่วยความจำภายนอก โดยจะส่งสัญญาณพัลส์ ออกมาทุก ๆ แมชีนไซเคิล อย่างไรก็ตามสามารถดิสเอเบิลสัญญาณพัลส์นี้ได้ โดยการเซตบิต 0 ของรีจิสเตอร์ AUXR
PSEN	29	เอาต์พุต	Program Store Enable : ใช้ในการส่งสัญญาณเพื่อร้องขอติดต่อกับหน่วยความจำภายนอก เมื่อต้องการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอก ไมโครคอนโทรลเลอร์จะส่งสัญญาณออกมาที่ขานี้ 2 ครั้งในแต่ละแมชีนไซเคิล แต่ถ้าติดต่อกับหน่วยความจำข้อมูลภายนอก ขานี้จะไม่มีการส่งสัญญาณใดๆ นอกจากนี้ยังใช้ประกอบในการอ่าน-เขียนข้อมูลในหน่วยความจำโปรแกรมด้วยกระบวนการ ISP - สำหรับเบอร์ P89C51RD* ให้ต่อขานี้ลงกราวด์ แล้วป้อนไฟ ขนาด +12V (+ - 0.5V) เข้าที่ขา EA/Vpp - สำหรับเบอร์ P89C51RD2 ให้ต่อขานี้ลงกราวด์ ป้อนลอจิก "1" เข้าที่ขา P2.7 และป้อนแรงดัน +5V เข้าที่ขา EA/Vpp
EA/Vpp	31	อินพุต	External Access enable/Programming voltage input : ใช้สำหรับติดต่อหน่วยความจำโปรแกรมจากภายนอกหรือภายในตัวไมโครคอนโทรลเลอร์ "0" เลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำภายนอก "1" เลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำภายใน นอกจากนี้ที่ขานี้ยังใช้เป็นขาอินพุตสำหรับรับแรงดันสำหรับโปรแกรมหน่วยความจำภายในไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

			- สำหรับเบอร์ P89C51RD* ต้องการแรงดัน +12V (+ - 0.5V) - สำหรับเบอร์ P89C51RD2ต้องการแรงดัน +5V
XTL1	19	อินพุต	- ขาอินพุตรับสัญญาณจากวงจรขยายออสซิเลเตอร์ โดยต่อที่ขา XTAL 2 และ จากภายนอกในการใช้งานปกติ ขานี้และขา XTAL 2 ต่อเข้ากับคริสตอลและตัวเก็บประจุชดเชยค่าน้อยๆ
XTL2	18	เอาต์พุต	- ขาเอาต์พุตของวงจรขยายออสซิเลเตอร์ที่อยู่ภายในตัวของไมโครคอนโทรลเลอร์ในการใช้งานปกติขานี้และขาXTAL 1 ต่อเข้ากับคริสตอลและตัวเก็บประจุชดเชยค่าน้อย ๆ

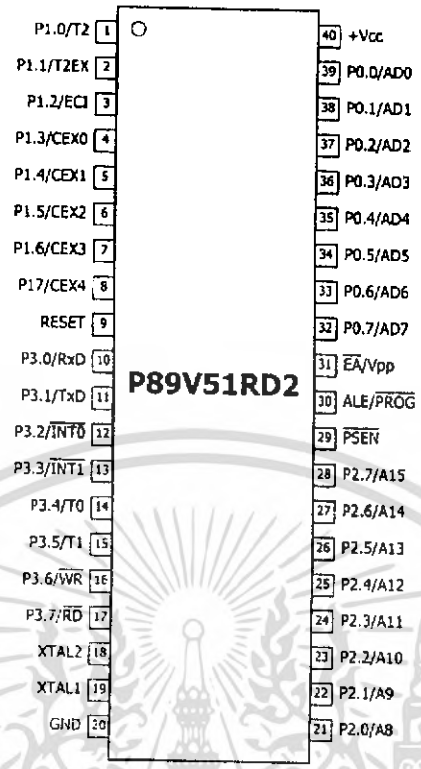
สัญญาณจากภายในจะต่อออกสู่ภายนอกทางขาของตัว P89C51RD2BN ที่มีขาอยู่ทั้งหมด 40 ขา ดังรูปที่ 2.1 ซึ่งมีการใช้งานต่างๆที่สำคัญดังนี้

- VCC ขา 40 เป็นขาที่ต้องป้อนไฟเลี้ยง +5 โวลต์ เข้าไปเพื่อให้วงจรทำงานได้
- VSS ขา 20 เป็นขาที่ต้องต่อกับกราวด์ของแหล่งจ่ายไฟ
- RST ขา 9 เป็นขารีเซ็ตที่จะทำหน้าที่รีเซ็ตการทำงานของ P89C51RD2BN ถ้าป้อนสัญญาณที่มีสภาวะลอจิก 1 ที่ขานี้จะเป็นการรีเซ็ตการทำงานกลับไปเริ่มทำงานจากคำสั่งที่อยู่ในหน่วยความจำตำแหน่ง 0000
- PSEN ขา 29 ใช้ส่งสัญญาณเพื่ออ่านโปรแกรมซึ่งจะเก็บไว้ในหน่วยความจำภายในชิพ
- XTAL 1 ขา 19 ใช้ต่อกับตริสตอลภายนอก โดยที่จะใช้เพื่อเป็นอินพุตป้อนเข้าสู่วงจรออสซิเลเตอร์
- XTAL 2 ขา 18 ใช้ต่อกับตริสตอลภายนอก โดยที่จะใช้เพื่อเป็นอินพุตป้อนเข้าสู่วงจรออสซิเลเตอร์

เมื่อเริ่มจ่ายพลังงานจะต้องมีการรีเซ็ตก่อนเสมอ ซึ่งการรีเซ็ตในขณะที่เริ่มจ่ายพลังงานจะกระทำโดยอัตโนมัติ โดยการทำงานของวงจรรีเซ็ตก็คือ เมื่อเริ่มมีการจ่ายพลังงานให้แก่ MCS-51 จะทำให้ขา RST มีสถานะเป็นหนึ่งในช่วงเวลาหนึ่งซึ่งขึ้นอยู่กับค่าตัวเก็บประจุ ความเร็วในการประจุ เพื่อให้แน่ใจว่ามีการรีเซ็ตเกิดขึ้นจริง ในช่วงเวลาเริ่มต้นเมื่อจ่ายไฟ 5 โวลต์ให้ขา VCC จะต้องให้ขารีเซ็ตมีสถานะเป็น 1 นานพอที่จะทำให้วงจรออสซิเลเตอร์เริ่มทำงานรวมกับช่วงเวลา 2 แมชีนไซเกิล

ในช่วงเริ่มต้นให้ VCC แก่ MCS-51 (ช่วง power up) VCC ควรมีค่าตามที่กำหนดภายในเวลา 10 วินาที (rise time) โดยช่วงเวลาที่วงจรออสซิเลเตอร์เริ่มต้นทำงานจะขึ้นอยู่กับความถี่ของคริสตอลที่ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 แสดงขาต่างๆของ MCS-51 เบอร์P89C51RD2

2.2.2 พอร์ทอินพุต/เอาต์พุต

PORT 0

เป็นพอร์ทขนานขนาด 8 บิต อยู่ที่ขา 39 ถึง 32 เริ่มจากบิต 0 ถึง 7 ตามลำดับพอร์ท 0 นี้ยังใช้ได้ทั้งการรับส่งตำแหน่งและข้อมูลกับหน่วยความจำหรือใช้รับส่งข้อมูลก็ได้ นอกจากนี้ยังใช้งานได้หลายอย่างดังนี้

1. ใช้สำหรับส่งค่าตำแหน่งหน่วยความจำที่ต้องการติดต่อด้วย โดย 8 พอร์ทล่างจะถูกส่งออกไปทางพอร์ท 0 และ 8 บิตบนจะถูกส่งออกไปทางพอร์ท 2
2. ใช้รับส่งข้อมูลกับ Data Memory หรือใช้รับข้อมูลจาก Program Memory
3. ใช้รับส่งข้อมูลออกจากพอร์ทโดยตรง

PORT 1

เป็นพอร์ทขนานขนาด 8 บิต อยู่ที่ขา 1 ถึง 8 เริ่มจากบิต 0 ถึงบิต 7 ตามลำดับ ใช้ทำหน้าที่เป็นตัวรับส่งข้อมูลเท่านั้น ไม่สามารถส่งตำแหน่งได้

PORT 2

เป็นพอร์ทขนานขนาด 8 บิต อยู่ที่ขา 21 ถึง 28 เริ่มจากบิต 0 ถึงบิต 7 ตามลำดับ ใช้งานเพียง 2 ลักษณะคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ใช้ส่งค่าตำแหน่งหน่วยความจำภายนอกที่ต้องติดต่อทำงานร่วมกับพอร์ท 0

2. ใช้เป็นพอร์ทรับส่งข้อมูลภายนอก

PORT 3

เป็นพอร์ทขนานขนาด 8 บิต อยู่ที่ขา 10 ถึง 17 เริ่มจากบิต 0 ถึงบิต 7 ตามลำดับ นอกจากจะใช้งานเหมือนพอร์ทอื่นแล้วยังใช้งานอื่นโดยใช้คำสั่งควบคุมดังนี้

P3.0 (RxD)	เป็นขาที่ใช้รับข้อมูลแบบอนุกรม
P3.1 (TxD)	เป็นขาที่ใช้ส่งข้อมูลแบบอนุกรม
P3.2 (INT0)	ใช้รับสัญญาณขัดจังหวะจากภายนอก
P3.3 (INT1)	ใช้รับสัญญาณขัดจังหวะจากภายใน
P3.4 (T0)	ใช้เป็นขารับสัญญาณให้เคาน์เตอร์ของไทเมอร์ 0
P3.5 (T1)	ใช้เป็นขารับสัญญาณให้เคาน์เตอร์ของไทเมอร์ 1
P3.6 (WR)	ใช้เป็นขารับสัญญาณควบคุมการเขียนข้อมูลยังหน่วยความจำสำหรับข้อมูลภายนอก
P3.7 (RD)	ใช้เป็นขารับสัญญาณควบคุมการอ่านข้อมูลยังหน่วยความจำสำหรับข้อมูลภายนอก

2.2.3 รีจิสเตอร์สำหรับใช้งานทั่วไปใน MCS-51

รีจิสเตอร์ A,B และรีจิสเตอร์ใช้งานทั่วไป R0-R7 ซึ่งอยู่ในหน่วยความจำเก็บข้อมูลทั่วไปภายในชิพบริเวณ 128 ไบต์แรก รีจิสเตอร์ใช้งานทั่วไป R0-R7 ใน MCS-51 มีด้วยกันทั้งหมด 4 กลุ่ม แต่ละกลุ่มประกอบด้วยรีจิสเตอร์จำนวน 8 ตัว (R0-R7) ซึ่งมีชื่อเรียกเหมือนกัน ดังนั้นรีจิสเตอร์ใช้งานทั่วไป R0-R7 ใน MCS-51 จึงมีทั้งหมด 32 ตัว ในการทำงานที่ขณะใดๆ รีจิสเตอร์ทั้ง 4 กลุ่ม (R0-R7) นั้นจะถูกเรียกใช้งานเพียงกลุ่มเดียวเท่านั้น โดยการเลือกใช้งานของตัวรีจิสเตอร์ R0-R7 กลุ่มใดกลุ่มหนึ่งใน 4 กลุ่ม นั้นจะกระทำโดยการเซตหรือเคลียร์บิต RS0,RS7 ในรีจิสเตอร์ใช้งานเฉพาะ PSW

2.2.4 ไทเมอร์/เคาน์เตอร์

- การทำงานเป็นตัวตั้งเวลาหรือไทเมอร์ เมื่อกำหนดให้ทำงานเป็นตัวตั้งเวลาหรือไทเมอร์ ค่าของรีจิสเตอร์จะเพิ่มขึ้นทุกๆแมชีนไซเกิลดังนั้นเมื่อทำงานเป็นไทเมอร์ รีจิสเตอร์จะทำการนับค่าของแมชีนไซเกิลนั่นเอง และเมื่อแมชีนไซเกิลประกอบด้วยคาบเวลาของวงจรกำเนิดสัญญาณนาฬิกา 6 คาบเวลา สำหรับ P89C51RD2HBP ดังนั้นอัตราในการนับของรีจิสเตอร์จึงเท่ากับ 1/6 ของความถี่สัญญาณนาฬิกาในกรณีเลือกโหมด 6 สัญญาณนาฬิกา/ไซเกิล (สำหรับ P89C51RD2BN เลือกได้ทั้ง 12 และ 6 สัญญาณนาฬิกา/ไซเกิล)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การทำงานเป็นตัวนับ (เคาน์เตอร์) ค่าของรีจิสเตอร์จะเพิ่มขึ้นก็ต่อเมื่อมีการเปลี่ยนแปลงของระดับโลจิกจาก "1" เป็น "0" เกิดขึ้นที่ขาอินพุตทางฮาร์ดแวร์ของวงจรไทเมอร์/เคาน์เตอร์ ซึ่งก็คือขา T0 (P3.4) ขาT1 (P3.5) และขา T2 (P1.0) โดยจะมีการสุ่มรับสัญญาณจากขาอินพุตในทุก ๆ คาบเวลาที่ 2 ของสเตตที่ 5 (S5P2) ในแต่ละแมชีนไซเกิล

เมื่อสัญญาณอินพุตเปลี่ยนแปลงจาก "1" เป็น "0" เป็นเวลา 1 ไซเกิล ในไซเกิลต่อมาค่าของการนับจะเพิ่มขึ้นหนึ่งค่า และจะไปปรากฏในรีจิสเตอร์ภายในคาบเวลาที่ 1 ของสเตตที่ 3 (S3P1) ของแมชีนไซเกิลต่อไปหลังจากที่ตรวจจับพบการเปลี่ยนแปลงที่ขาไทเมอร์อินพุตแล้ว เมื่อเป็นเช่นนี้กระบวนการตรวจจับการเปลี่ยนแปลงของสัญญาณอินพุตที่เข้าที่ขาของไทเมอร์ จะต้องใช้เวลา 2 แมชีนไซเกิล อัตราการนับของเคาน์เตอร์จึงเท่ากับ 1/12 ของความถี่สัญญาณนาฬิกาใน P89C51RD2 ดังนั้น ความถี่สูงสุดของสัญญาณอินพุตที่ไทเมอร์/เคาน์เตอร์ภายในไมโครคอนโทรลเลอร์ MCS-51 สามารถตรวจจับได้จึงเท่ากับความถี่ของสัญญาณนาฬิกาหารด้วย 12*ยกตัวอย่างไมโครคอนโทรลเลอร์ MCS-51 แบบแฟรชเมอร์ P89C51RD2 ซึ่งใช้ความถี่ของสัญญาณนาฬิกาได้สูงสุด 20 MHz จะสามารถตรวจจับความถี่ของสัญญาณซึ่งได้สูงสุดประมาณ 1.667 MHz

2.2.4.1 รีจิสเตอร์ไทเมอร์

มีด้วยกัน 4 ตัว คือ TL0 มีแอดเดรสอยู่ที่ 0x8A ,TH0 มีแอดเดรสอยู่ที่ 0x8C ,TL1 มีแอดเดรสอยู่ที่ 0x8B และ TH1 มีแอดเดรสอยู่ที่ 0x8D รีจิสเตอร์ทั้ง 4 ตัว จะอยู่ในพื้นที่ของรีจิสเตอร์ฟังก์ชั่นพิเศษหรือ SFR รีจิสเตอร์แต่ละตัวมีขนาด 8 บิต แต่ในการใช้งานโดยทั่วไปมักใช้งานร่วมกันโดยจัดเป็นคู่ คือ TL0 กับ TH0 รวมกันเป็นรีจิสเตอร์ไทเมอร์ 0 ขนาด 16 บิต และ TL1 กับ TH1 รวมกันเป็นรีจิสเตอร์ไทเมอร์ 1 ขนาด 16 บิต โดยใน TL0 และ TL1 เก็บข้อมูล 8 บิตล่าง ส่วน TH0 และ TH1 เก็บข้อมูลของ 8 บิตบน รีจิสเตอร์ไทเมอร์ทั้ง 2 คู่ เมื่อนำมาใช้ร่วมกันจะสามารถเก็บค่าของการนับได้สูงสุด 65,536 หรือ 0xFFFF เมื่อนับถึงค่านี้แล้วจะเริ่มนับ 0x0000 ใหม่ และเมื่อเกิดการนับรอบใหม่ บิต TF0 หรือ TF1 ในรีจิสเตอร์ TCON ที่ใช้ควบคุมการทำงานของไทเมอร์จะเกิดการเซต เพื่อแจ้งให้ทราบว่านับเกินค่าสูงสุดแล้ว การเซตบิต TF0 หรือ TF1 ขึ้นอยู่กับว่าเลือกใช้งานรีจิสเตอร์ไทเมอร์ตัวใด

2.2.4.1.1 รีจิสเตอร์ควบคุมการทำงานของไทเมอร์/เคาน์เตอร์หรือTCON (Time/Counter Control register)

เป็นรีจิสเตอร์ขนาด 8 บิต มีแอดเดรสอยู่ที่ 0x88 ในพื้นที่ของรีจิสเตอร์ SFR สามารถเข้าถึงได้ในระดับบิต มีรายละเอียดการทำงานดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

รูปที่ 2.2 รีจิสเตอร์ใช้งานเฉพาะ TCON

TF1 (Timer 1 overflow flag)

เซตด้วยกระบวนการทางฮาร์ดแวร์ซึ่งเมื่อค่าของรีจิสเตอร์ Timer 1 เกิดการนับเกินหรือเกิดโอเวอร์โฟลว การเคลียร์บิตนี้ทำได้ด้วยกระบวนการทางฮาร์ดแวร์เช่นกัน โดยบิตนี้จะเคลียร์เมื่อมีการอินเตอร์รัปต์เกิดขึ้น

TR1 (Timer 1 run control bit)

ใช้ในการเปิดปิดการทำงานของไทเมอร์1 (เอ็นเอเบิลหรือดิสเอเบิล)ทำการเซตและเคลียร์ด้วยกระบวนการทางซอฟแวร์ ถ้าต้องการให้ไทเมอร์ 1 ทำงานต้องเซตบิตนี้ให้เป็น 1

TF0 (Timer 0 overflow flag)

เซตด้วยกระบวนการทางฮาร์ดแวร์ซึ่งเมื่อค่าของรีจิสเตอร์ Timer 0 เกิดการนับเกินหรือเกิดโอเวอร์โฟลว การเคลียร์บิตนี้ทำได้ด้วยกระบวนการทางฮาร์ดแวร์เช่นกัน โดยบิตนี้จะเคลียร์เมื่อมีการอินเตอร์รัปต์เกิดขึ้น

TR0 (Timer 0 run control bit)

ใช้ในการเปิดปิดการทำงานของไทเมอร์0 (เอ็นเอเบิลหรือดิสเอเบิล) ทำการเซตและเคลียร์ด้วยกระบวนการทางซอฟแวร์ ถ้าต้องการให้ไทเมอร์ 0 ทำงานต้องเซตบิตนี้ให้เป็น 1

IE1 (External Interrupt 1 edge flag)

บิตนี้จะใช้ในกระบวนการอินเตอร์รัปต์ สามารถเซตได้ด้วยกระบวนการฮาร์ดแวร์ เมื่อสามารถตรวจตรวจจับขอบขาของสัญญาณอินเตอร์รัปต์จากภายนอกที่ขาอินพุตอินเตอร์รัปต์ที่ 1 (INT1) ได้ และจะทำการเคลียร์เมื่อมีการบริการอินเตอร์รัปต์เกิดขึ้น

IT1 (Interrupt 1 type control bit)

บิตนี้จะใช้ในกระบวนการอินเตอร์รัปต์ โดยใช้ในการเลือกลักษณะของสัญญาณอินเตอร์รัปต์จากภายนอกที่ต้องการให้ทำการตอบสนองสำหรับขาอินพุตอินเตอร์รัปต์ 1 (INT1) การเซตและเคลียร์ทำได้ด้วยกระบวนการทางซอฟแวร์

"0" เลือกขอบขาลงของสัญญาณ (foaling edge)

"1" เลือกระดับลอจิกต่ำ (low level triggered)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IE0 (External Interrupt 0 edge flag)

บิตนี้จะใช้ในกระบวนการอินเตอร์รัปต์ สามารถเซตได้ด้วยกระบวนการฮาร์ดแวร์ เมื่อสามารถตรวจตรวจจับขอบขาของสัญญาณอินเตอร์รัปต์จากภายนอกที่ขาอินพุตอินเตอร์รัปต์ที่ 0 (INT0) ได้ และจะทำการเคลียร์เมื่อมีการบริการอินเตอร์รัปต์เกิดขึ้น

IT0 (Interrupt 0 type control bit)

บิตนี้จะใช้ในกระบวนการอินเดอร์รัปต์ โดยใช้ในการเลือกลักษณะของสัญญาณอินเตอร์รัปต์จากภายนอกที่ต้องการให้ทำการตอบสนองสำหรับขาอินพุตอินเตอร์รัปต์ 0 (INT0) การเซตและเคลียร์ทำได้ด้วยกระบวนการทางซอฟแวร์

"0" เลือกขอบขาลงของสัญญาณ (foaling edge)

"1" เลือกระดับลอจิกต่ำ(low level triggered)

2.2.4.1.2 รีจิสเตอร์เลือกโหมดการทำงานของไทเมอร์/เคาน์เตอร์หรือ TMOD (Timer/Counter Mode Control Register)

เป็นรีจิสเตอร์ขนาด 8 บิต มีแอดเดรสอยู่ที่ 0x89 ในพื้นที่ของรีจิสเตอร์ SFR ไม่สามารถเข้าถึงได้ในระดับบิต แบ่งการทำงานเป็น 2 ส่วนคือ 4 บิตล่างใช้เลือกโหมดการทำงานของไทเมอร์ 0 และ 4 บิตบนใช้เลือกโหมดการทำงานของไทเมอร์ 1

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
GATE	C/T	M1	M0	GATE	C/T	M1	M0
ไทเมอร์1				ไทเมอร์0			

รูปที่ 2.3 รีจิสเตอร์ใช้งานเฉพาะ TMOD

GATE

ใช้เลือกลักษณะการควบคุมการทำงานของไทเมอร์/เคาน์เตอร์

"0" ไทเมอร์/เคาน์เตอร์จะทำงานเมื่อบิต TRx ในรีจิสเตอร์ TCON เป็น "1"

"1" ไทเมอร์/เคาน์เตอร์จะทำงานเมื่อบิต TRx ในรีจิสเตอร์ TCON เป็น "1"และสถานะลอจิกที่ขาอินพุต อินเตอร์รัปต์ INT0 และ INT1 เป็น "1"

C/T (Timer or Counter selector)

ใช้เลือกลักษณะการทำงานของไทเมอร์/เคาน์เตอร์

"0" เลือกทำงานเป็นไทเมอร์ โดยใช้สัญญาณอินพุตจากสัญญาณนาฬิกาภายใน

"1" เลือกทำงานเป็นเคาน์เตอร์ โดยรับสัญญาณอินพุตจากทางขา T0 หรือ T1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

M1,M0 (Mode selector bit)

ใช้เลือกโหมดการทำงานของไทเมอร์/เคาน์เตอร์

"00" เลือกให้ทำงานในโหมดไทเมอร์/เคาน์เตอร์ 13 บิต

"01" เลือกให้ทำงานในโหมดไทเมอร์/เคาน์เตอร์ 16 บิต

"10" เลือกให้ทำงานในโหมดไทเมอร์/เคาน์เตอร์ 8 บิต แบบตั้งค่าอัตโนมัติ

"11" สำหรับไทเมอร์ 0 เลือกให้ทำงานในโหมดไทเมอร์/เคาน์เตอร์แยกส่วนโดยแยกออกเป็นไทเมอร์/ เคาน์เตอร์ 8 บิต 2 ตัว รีจิสเตอร์ TL0 จะได้รับการควบคุมการเปิดปิดจากบิต TR0 ในรีจิสเตอร์ TCON และที่รีจิสเตอร์ TH0 ซึ่งเป็นไทเมอร์/เคาน์เตอร์ 8 บิต อีกตัวหนึ่ง จะได้รับการควบคุมจากบิต TR1 ในรีจิสเตอร์ TCON ซึ่งในกรณีของไทเมอร์ 1 เป็นการสั่งให้ไทเมอร์/เคาน์เตอร์ 1 หยุดทำงาน (ดิสเอเบิล)

2.2.5 การอินเตอร์รัปต์

การอินเตอร์รัปต์ (interrupt) เป็นกระบวนการขัดจังหวะการทำงานของไมโครคอนโทรลเลอร์ในไมโครคอนโทรเลอร์ MCS-51 มาตรฐานจะสามารถที่จะตอบสนองการอินเตอร์รัปต์ที่เกิดขึ้นได้จาก 5 แหล่งกำเนิด ประกอบด้วย

- อินเตอร์รัปต์จากภายนอกผ่านทางขา INT0
- อินเตอร์รัปต์จากภายนอกผ่านทางขา INT1
- อินเตอร์รัปต์จากไทเมอร์ 0
- อินเตอร์รัปต์จากไทเมอร์ 1
- อินเตอร์รัปต์จากพอร์ทอนุกรมภายในตัวไมโครคอนโทรลเลอร์

ในขณะที่ไมโครคอนโทรลเลอร์ P89C51RD2 สามารถกำหนดลำดับความสำคัญของการตอบสนองอินเตอร์รัปต์ได้ถึง 4 ระดับจากแหล่งกำเนิดสัญญาณอินเตอร์รัปต์ 7 แหล่ง โดยแหล่งกำเนิดสัญญาณอินเตอร์รัปต์ที่เพิ่มเติมเข้ามาคือ สัญญาณอินเตอร์รัปต์จากไทเมอร์ 2 และจากโมดูลวงจรนับแบบโปรแกรมได้หรือ PCA

มีรีจิสเตอร์ 3 ตัวที่เกี่ยวข้องคือ รีจิสเตอร์ IE , IP และIPH แสดงรายละเอียดในรูปที่ 7-1 ถึง 7-3 การกำหนดลำดับความสำคัญของการตอบสนองการอินเตอร์รัปต์จะใช้ข้อมูลในแต่ละบิตของรีจิสเตอร์ IP และ IPH ดังแสดงความสัมพันธ์ในตารางที่ 7-1 ยกตัวอย่างการเกิดอินเตอร์รัปต์จากตัวไทเมอร์ 2 มีลำดับความสำคัญสูงสุดหรือระดับ 3 ต้องกำหนดให้บิต PT2 หรือบิต 5 ของรีจิสเตอร์ IP และบิต PT2H หรือบิต 5 ของรีจิสเตอร์ IPH เป็น "1" ทั้งคู่ จึงตอบสนองการอินเตอร์รัปต์ได้จาก 6 แหล่งโดยเพิ่มการรับสัญญาณอินเตอร์รัปต์จากไทเมอร์/เคาน์เตอร์ 2 อีกหนึ่งแหล่งกำเนิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.5.1 การจัดลำดับการอินเตอร์รัปต์ในไมโครคอนโทรลเลอร์ MCS-51

เมื่อมีการอินเตอร์รัปต์ในไมโครคอนโทรลเลอร์ MCS-51 เกิดขึ้น และเอ็นเอเบิลการตอบสนองการอินเตอร์รัปต์ไว้กระบวนการหลังจากนั้นซีพียูจะกระโดดไปยังแอดเดรสในหน่วยความจำที่ซึ่งจะกำหนดไว้ ซึ่งเราจะเรียกแอดเดรสนี้ว่า **แอดเดรสอินเตอร์รัปต์เวกเตอร์** (interrupt vector address) ดังนั้นจะต้องเขียนโปรแกรมย่อยกรบริการอินเตอร์รัปต์ไว้ที่แอดเดรสอินเตอร์รัปต์ไว้ที่แอดเดรสอินเตอร์รัปต์เวกเตอร์นี้ โดยค่าของแอดเดรสอินเตอร์รัปต์เวคเตอร์จะแตกต่างกันไปในการอินเตอร์รัปต์ต่าง ๆ ดังมีรายละเอียดต่อไปนี้

	บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
IE (98H)	EA	EC	ET2	ES	ET1	EX1	ET0	EX0

"1"เอ็นเอเบิล

"0"ดิสเอเบิล

บิต	ชื่อบิต	รายละเอียด
IE.7	EA	เอ็นเอเบิลหรือดิสเอเบิลการอินเตอร์รัปต์หลัก
IE.6	EC	เอ็นเอเบิลการอินเตอร์รัปต์จากโมดูล PCA
IE.5	ET2	เอ็นเอเบิลการอินเตอร์รัปต์ไทเมอร์ 2
IE.4	ES	เอ็นเอเบิลการอินเตอร์รัปต์จากพอร์อนุกรม
IE.3	ET1	เอ็นเอเบิลการอินเตอร์รัปต์ไทเมอร์ 1
IE.2	EX1	เอ็นเอเบิลการอินเตอร์รัปต์จากสัญญาณภายนอกผ่านขา INT1
IE.1	ET0	เอ็นเอเบิลการอินเตอร์รัปต์ไทเมอร์ 0
IE.0	EX0	เอ็นเอเบิลการอินเตอร์รัปต์จากสัญญาณภายนอกผ่านขา INT0

รูปที่ 2.4 รายละเอียดของรีจิสเตอร์ IE ซึ่งใช้ในการเอ็นเอเบิลการอินเตอร์รัปต์

	บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
IE(088H)	-	PPC	PT2	PS	PT1	PX1	PT0	PX0

"1" เอ็นเอเบิล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

62700

"0" ดิสเอเบิล

บิต	ชื่อบิต	รายละเอียด
IP.7	-	-
IP.6	PPC	บิตกำหนดลำดับความสำคัญของการอินเตอร์รัปต์จากโมดูล PCA
IP.5	PT2	บิตกำหนดลำดับความสำคัญของการอินเตอร์รัปต์จากไทเมอร์ 2
IP.4	PS	บิตกำหนดลำดับความสำคัญของการอินเตอร์รัปต์จากพอร์ทอนุกรม
IP.3	PT1	บิตกำหนดลำดับความสำคัญของการอินเตอร์รัปต์ไทเมอร์ 1
IP.2	PX1	บิตกำหนดลำดับความสำคัญของการอินเตอร์รัปต์จากสัญญาณภายนอกผ่าน ขา INT1
IP.1	PT0	บิตกำหนดลำดับความสำคัญของการอินเตอร์รัปต์ไทเมอร์ 0
IP.0	PX0	บิดกำหนดลำดับความสำคัญของการอินเตอร์รัปต์จากสัญญาณภายนอกผ่านขา INT0

รูปที่ 2.5 รายละเอียดของรีจิสเตอร์ IP ซึ่งใช้ในการกำหนดลำดับความสำคัญของแหล่งกำเนิดสัญญาณอินเตอร์รัปต์

	บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
IE (087H)	-	PPCH	PT2H	PSH	PT1H	PX1H	PT0H	PX0H

"1"เอ็นเอเบิล

"0"ดิสเอเบิล

บิต	ชื่อบิต	รายละเอียด
IPH.7	-	-
IPH.6	PPCH	
IPH.5	PT2H	
IPH.4	PSH	
IPH.3	PT1H	
IPH.2	PX1H	บิตกำหนดลำดับความสำคัญสูงสุดให้แก่การอินเตอร์รัปต์จากสัญญาณ ภายนอกผ่านขา INT1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IPH.1	PT0H	
IPH.0	PX0H	บิตกำหนดลำดับความสำคัญสูงสุดให้แก่การอินเตอร์รัปต์จากสัญญาณภายนอกผ่านขา INT0

รูปที่ 2.6 รายละเอียดของรีจิสเตอร์ IPH ซึ่งใช้กำหนดลำดับความสำคัญสูงสุดให้แก่แหล่งกำเนิดสัญญาณอินเตอร์รัปต์

ตารางที่ 2.2 แสดงความสัมพันธ์ของรีจิสเตอร์ IP และ IPH ในการกำหนดระดับของลำดับความสำคัญในการตอบสนองการอินเตอร์รัปต์

บิตกำหนดลำดับความสำคัญ		ระดับของลำดับความสำคัญ
IPH.x	IP.x	
0	0	ระดับ 0 (ต่ำสุด)
0	1	ระดับ 1
1	0	ระดับ 2
1	1	ระดับ 3(สูงสุด)

การอินเตอร์รัปต์ภายนอกที่ขา INT0 มีค่าแอดเดรสอินเตอร์รัปต์เวกเตอร์อยู่ที่ 0x0003
การอินเตอร์รัปต์จากไทเมอร์ 0 มีค่าแอดเดรสอินเตอร์รัปต์เวกเตอร์อยู่ที่ 0x000B
การอินเตอร์รัปต์ภายนอกที่ขา INT1 มีค่าแอดเดรสอินเตอร์รัปต์เวกเตอร์อยู่ที่ 0x0013
การอินเตอร์รัปต์จากไทเมอร์ 1 มีค่าแอดเดรสอินเตอร์รัปต์เวกเตอร์อยู่ที่ 0x001B
การอินเตอร์รัปต์จากพอร์ทอนุกรม มีค่าแอดเดรสอินเตอร์รัปต์เวกเตอร์อยู่ที่ 0x0023

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การอินเตอร์รัปต์จากไทเมอร์ 2 มีค่าแอดเดรสอินเตอร์รัปต์เวกเตอร์อยู่ที่ 0x002B
การอินเตอร์รัปต์จากโมดูล PCA มีค่าแอดเดรสอินเตอร์รัปต์เวกเตอร์อยู่ที่ 0x0033

2.2.6 สื่อสารข้อมูลผ่านพอร์ทอนุกรมของไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์สามารถสื่อสารข้อมูลได้ทั้งรับและส่งในเวลาเดียวกัน หรือที่เรียกว่าเป็นแบบฟูลดูเพล็กซ์ (full duplex) โดยใช้ขา P3.0 ทำหน้าที่รับข้อมูล (RxD) และขาP3.1 ทำหน้าที่ส่งข้อมูล (TxD) รูปแบบการสื่อสารข้อมูลจะต้องมีรูปแบบเดียวกันทั้งคู่ อันประกอบด้วยอัตราในการถ่ายทอดข้อมูลหรืออัตราบอด (baudrate) บิตเริ่มต้น (start bit) บิตข้อมูล (data bit) บิตตรวจสอบพาริตี้ (parity bit) และบิตหยุด (stop bit)

2.2.6.1 รีจิสเตอร์ที่เกี่ยวข้องในการสื่อสารข้อมูลอนุกรม

2.2.6.1.1 รีจิสเตอร์ควบคุมการทำงานของพอร์ทอนุกรมหรือSCON (Serial port Control Register)

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI

รูปที่ 2.7 แสดงรีจิสเตอร์ที่ใช้งานเฉพาะ SCON

SM0/FE (Serial port mode bit 0/Framing error bit)

ปกติจะใช้ร่วมกับบิต SM1 เพื่อกำหนดโหมดการทำงานของพอร์ทอนุกรม การเข้าถึงบิตนี้จะเกิดขึ้นได้ก็ต่อเมื่อมีการเคลียร์บิต SMOD ซึ่งก็คือบิต 7 ของรีจิสเตอร์ PCON ในกรณีที่ใช้ความสามารถตรวจจับความผิดพลาดของเฟรมข้อมูล บิตนี้ใช้แจ้งความผิดพลาดที่เกิดขึ้น โดยจะเซตเป็น "1" ทันทีเมื่อพบว่าไม่สามารถตรวจจับบิตหยุดหรือบิตปิดท้ายของข้อมูลของพอร์ทอนุกรมได้ การเอนเอเบิลความสามารถนี้ทำได้โดยการเซตบิต SMOD ในรีจิสเตอร์ PCON การเคลียร์บิตนี้ต้องกระทำทางซอฟแวร์เท่านั้น

SM1 (Serial port mode bit 1)

ใช้ร่วมกับบิต SM0 ในการกำหนดโหมดการทำงานของพอร์ทอนุกรมในไมโครคอนโทรลเลอร์ P89C51RD2 ดังรายละเอียดดังต่อไปนี้

SM0	SM1	โหมด	รายละเอียด	อัตราบอด
0	0	0	ชิพรีจิสเตอร์	ความถี่สัญญาณนาฬิกา/6 (ในโหมด 6 ไซเกิลสัญญาณนาฬิกา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0	1	1	UART 8 บิต	ปรับค่าได้
1	0	2	UART 9 บิต	ความถี่สัญญาณนาฬิกา/32 (ในโหมด 6 ไซเกิลสัญญาณนาฬิกา)
1	1	3	UART 9 บิต	ปรับค่าได้

SM2 (Serial port mode bit 2)

ใช้ในการเอ็นเอเบิลความสามรถการรับรู้แอดเดรสในการติดต่ออัตโนมัติ เมื่อมีการเชื่อมต่อไมโครคอนโทรลเลอร์หลายตัวเข้าด้วยกัน โดยความสามรถนี้จะใช้ได้ก็ต่อเมื่อวงจรพอร์ทอนุกรมของไมโครคอนโทรลเลอร์ทำงานในโหมด 2 หรือ 3 ถ้าบิต SM2 เป็น "1" บิต R1 จะไม่เซต เว้นแต่ข้อมูลบิตที่ 9 ที่เข้ามาเป็น "1" เป็นแจ้งว่า สามารถติดต่อได้และข้อมูลที่รับเข้ามาคือ ค่าแอดเดรสที่ต้องการติดต่อด้วย

ในกรณีที่พอร์ทอนุกรมทำงานในโหมด 1 ถ้าบิด SM2 เซต บิต RI จะไม่เปลี่ยนแปลงจนกว่าจะได้รับข้อมูลบิตหยุดหรือบิตปิดท้าย และข้อมูลที่รับได้จะเป็นข้อมูลแอเดรสที่ต้องการติดต่อด้วย

ในกรณีที่วงจรพอร์ทอนุกรมทำงานอยู่ในโหมด 0 บิต SM2 นี้เป็น "0"

REN (Received enable bit)

ใช้เอ็นเอเบิลความสามารถในการรับข้อมูลของวงจรพอร์ทอนุกรม "1" เอ็นเอเบิลการรับข้อมูล "0"ดิสเอเบิลการรับข้อมูล การเซตหรือเคลียร์บิตนี้กระทำด้วยกระบวนการทางซอฟแวร์

TB8 (Transmit data bit 8)

ใช้เก็บข้อมูลบิต 8 หรือบิตที่ 9 ที่ต้องการส่งออกทางพอร์ทอนุกรม เมื่อทำงานในโหมด 2 และ 3

RB8 (Receive data bit 8)

ใช้เก็บข้อมูลบิต 8 ของข้อมูลที่รับเข้ามาทางพอร์ทอนุกรมเมื่อทำงานในโหมด 2 และ 3 ในโหมด 1 ถ้าบิต SM2=0 ข้อมูลของบิตหยุดจะเก็บไว้ที่บิดนี้ไม่ใช้งานในบิตนี้ในโหมด 0

TI (Transmit Interrupt flag)

บิตแสดงการเกิดอินเตอร์รัปต์จากการส่งข้อมูลออกทางพอร์ทอนุกรมซึ่งเมื่อทำงานในโหมด 0 บิตนี้จะเซตเมื่อมีการเริ่มต้นส่งบิตหยุดหรือบิตปิดท้าย การเคลียร์บิตนี้กระทำด้วยกระบวนการทางซอฟแวร์

RI (Receive interrupt flag)

บิตแสดงการเกิดอินเตอร์รัปต์เนื่องจากการรับข้อมูลเข้ามาทางพอร์ทอนุกรม เมื่อทำงานในโหมด 0 บิตนี้จะเซตเมื่อรับข้อมูลบิต 7 หรือบิตที่ 8 เสร็จสมบูรณ์ ตาถ้าทำงานในโหมดอื่น บิตนี้จะเซตเมื่อมีการรับบิตปิดท้ายดำเนินไปได้ครึ่งทาง นอกจากนี้การเซตบิตนี้ยังมีเงื่อนไขที่กำหนดโดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต SM2 ร่วมด้วย การเคลียร์บิตนี้ต้องกระทำด้วยกระบวนการทางซอฟแวร์ โดยส่วนใหญ่มักตั้งค่าของ SCON ดังนี้

SCON = 0x40 ; พอร์ทอนุกรมทำงานในโหมด 1 สามารถส่งข้อมูลได้อย่างเดียว

SCON = 0x50 ; พอร์ทอนุกรมทำงานในโหมด 1 สามารถรับและส่งข้อมูลได้

2.2.7 หลักการทำงานและคุณสมบัติของระบบบัส I²C

I²C ย่อมาจาก Inter-IC Communication หมายถึง การติดต่อสื่อสารระหว่างไอซีโดยบัส I²C ได้รับการพัฒนาขึ้นโดยบริษัท Phillips ด้วยจุดมุ่งหมายหลักคือ ต้องการให้ไอซีหรือโมดูลสามารถติดต่อ สั่งงาน และควบคุมภายใต้สายสัญญาณเพียง 2 เส้น เส้นหนึ่งคือสายข้อมูล และอีกเส้นหนึ่งคือ สายสัญญาณนาฬิกาที่ใช้ในการกำหนดจังหวะการทำงาน การต่อร่วมกันของอุปกรณ์บนบัส I2C ทำได้ง่ายมาก เพียงต่อสายข้อมูลและสายสัญญาณนาฬิกาของอุปกรณ์แต่ละตัวขนานหรือพ่วงกันไป ส่วนการกำหนดแอดเดรสหรือตำแหน่งสำหรับอุปกรณ์แต่ละตัว จะใช้รหัสข้อมูลและการกำหนดสภาวะลอจิกที่ขาแอดเดรสของอุปกรณ์แต่ละตัว

สายข้อมูลบนบัส I²C หรือมีชื่อเรียกอย่างเป็นทางการว่า สายข้อมูลอนุกรม หรือ SDA (Serial Data Line) ส่วนสายสัญญาณนาฬิกามีชื่อเรียกว่า สายสัญญาณนาฬิกาอนุกรม หรือ SCL (Serial Clock Line) ซึ่งทั้ง SDA และ SCL นั้นต่างเป็นสายสัญญาณ 2 ทิศทาง (Bi-Directional Line) ต้องมีการต่อตัวต้านทานพูลอัพกับแรงดัน +5 Vdc ไว้ตลอดเวลา เพื่อให้สายมีสภานะลอจิกสูงในขณะที่ไม่มีการติดต่อใช้งาน ทั้งยังช่วยในการป้องกันสัญญาณรบกวนที่อาจเข้ามาในสายสัญญาณทั้งสอง วงจรเอาต์พุตของอุปกรณ์ที่จะต่ออยู่บนบัส I²C จะต้องมีลักษณะที่เป็นวงจรเดรนเปิด (Open-Drain) หรือคอลเล็กเตอร์เปิด (Open-Collector)

อัตราการการถ่ายทอดข้อมูลบนบัส I²C สูงถึง 100 กิโลบิตต่อวินาทีในโหมดปกติและสูงถึง 400 กิโลบิตต่อวินาทีในโหมดความเร็วสูง อุปกรณ์ที่ต่อร่วมอยู่บนบัส I²C จะต้องมีค่าความจุไฟฟ้ารวมที่เกิดขึ้นระหว่างสาย SDA และ SCL ไม่เกิน 400 pF การเข้าถึงอุปกรณ์บนบัส I²C ใช้ข้อมูลสำหรับการเข้าถึง 2 ค่า คือ 7 บิต (7-bit Addressing) หรือ 10 บิต (10-bit Addressing)

บัส I²C ประกอบด้วยสายสัญญาณ 2 เส้น คือ SDA และ SCL อุปกรณ์ที่ต่อพ่วงบนบัสสามารถมีได้มากมาย ดังนั้นจึงต้องมีการกำหนดรูปแบบของการติดต่อบนบัส เพื่อให้ผู้ใช้งานทราบว่า ขณะนี้อุปกรณ์ใดติดต่อกันอยู่ และอุปกรณ์ใดเป็นตัวรับหรือตัวส่ง ซึ่งจะนิยามที่ควรรู้ ดังนี้

- อุปกรณ์ที่ทำหน้าที่ควบคุมจังหวะการติดต่อบนบัส I²C เรียกว่า มาสเตอร์ (Master)
- อุปกรณ์ที่ถูกควบคุมหรืออุปกรณ์ที่ต่อพ่วงเข้าไปบนบัส I²C เรียกว่า สเลฟ (Slave)

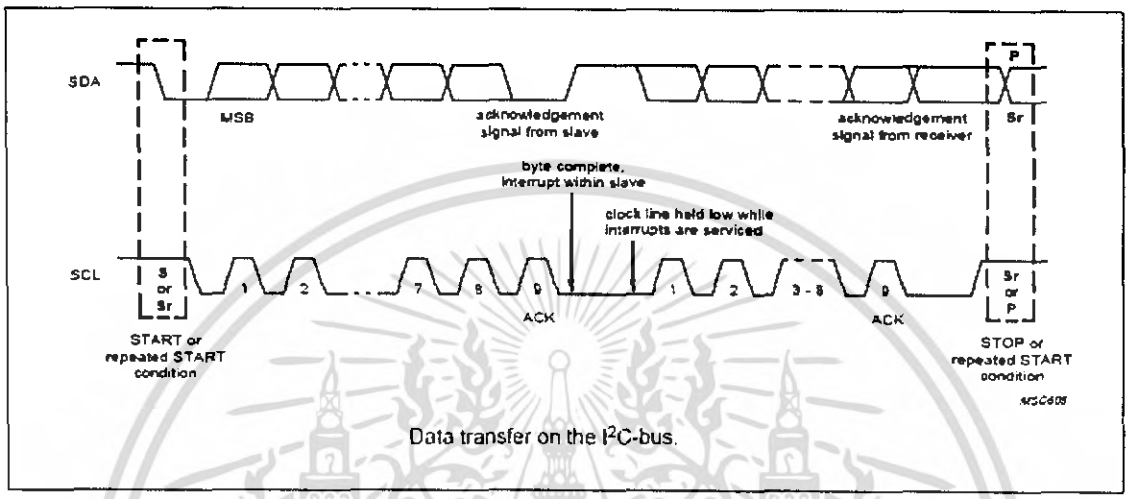
ข้อกำหนด 2 ประการที่สำคัญของการติดต่อบนบัส I²C มี ดังนี้

(1) การถ่ายทอดข้อมูลจะเกิดขึ้นได้เมื่อบัสว่างเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(2) ในระหว่างการถ่ายทอดข้อมูล เมื่อใดก็ตามที่สาย SCL มีสถานะเป็นลอจิกสูง สายข้อมูลต้องรักษาข้อมูลไว้อย่าให้เกิดการเปลี่ยนแปลงขึ้นเด็ดขาด มิฉะนั้น สัญญาณที่เกิดขึ้นจะได้รับการแปลความหมายเป็นสัญญาณควบคุมแทน

สภาวะที่เกิดขึ้นบนบัส I²C มีด้วยกัน 5 สภาวะ ดังรูปที่ 3.9 ซึ่งอธิบายได้ ดังนี้



รูปที่ 2.8 สภาวะที่เกิดบนบัส I²C

(1) บัสว่าง (Bus Not Busy) สภาวะนี้เกิดขึ้นเมื่อสถานะลอจิกบนสาย SDA และ SCL เป็นลอจิกสูงทั้งคู่นั่นหมายความว่า การถ่ายทอดข้อมูลสามารถเริ่มขึ้นได้

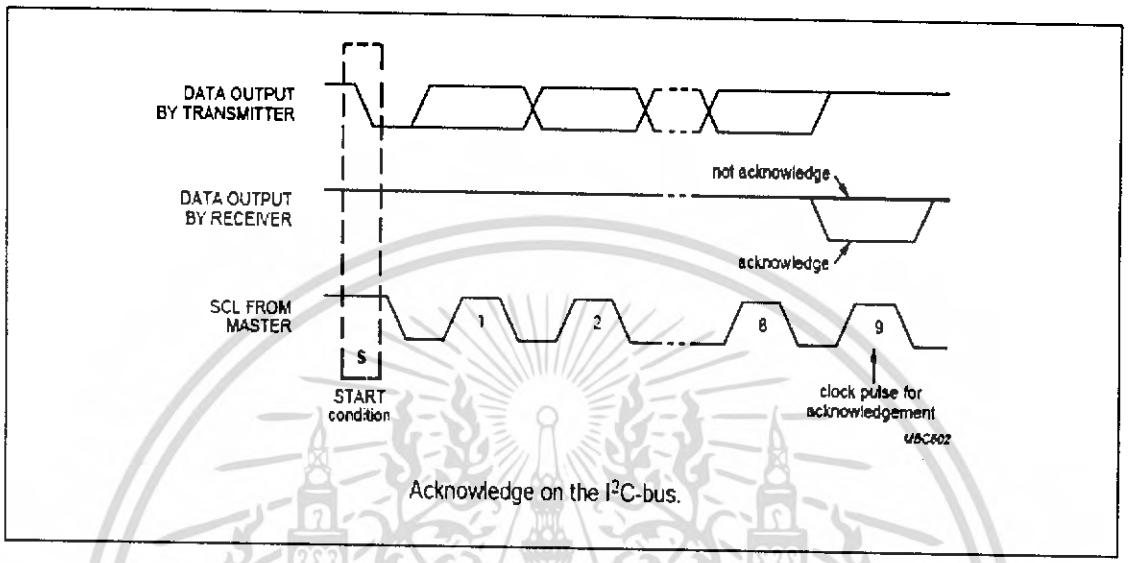
(2) เริ่มต้นการถ่ายทอดข้อมูล (Start Data Transfer) เกิดขึ้นเมื่อสาย SDA มีการเปลี่ยนแปลงระดับลอจิกจากสูงไปต่ำในขณะที่สาย SCL มีสถานะลอจิกสูง เรียกสภาวะที่เกิดขึ้นว่าสภาวะเริ่มต้น (START)

(3) ข้อมูลคำรงอยู่บนบัส (Data Valid) สภาวะนี้เกิดขึ้นถัดจากสภาวะเริ่มต้นโดยสถานะลอจิกที่เกิดขึ้นบนสาย SDA ก็คือ ข้อมูลที่ทำการถ่ายทอดเมื่อสาย SCL มีสถานะเป็นลอจิกสูงและสถานะที่สาย SDA ต้องคงที่ เพื่อให้อุปกรณ์รับรู้ข้อมูลในจังหวะนั้นว่าเป็น "0" หรือ "1" ซึ่งข้อมูลอาจเกิดการเปลี่ยนแปลงได้ในขณะที่สาย SCL มีสถานะเป็นลอจิกต่ำ แต่เมื่อใดก็ตามที่ต้องการให้เกิดการถ่ายทอดข้อมูลอย่างสมบูรณ์ สถานะลอจิกที่สาย SDA ต้องคงที่ตลอดช่วงเวลาที่สาย SCL มีสถานะเป็นลอจิกสูง หากเกิดการเปลี่ยนแปลงสถานะลอจิกของสาย SDA ในขณะที่สาย SCL มีสถานะลอจิกสูงอยู่นั้น อุปกรณ์มาสเตอร์ที่ทำการควบคุมการถ่ายทอดข้อมูล จะแปลความหมายเป็นสภาวะหยุดหรือสภาวะเริ่มต้นก็ได้ ทำให้ข้อมูลที่ทำการถ่ายทอดนั้นเกิดความผิดพลาดขึ้น

(4) รับรู้ข้อมูล (Acknowledge) เกิดขึ้นหลังจากที่การถ่ายทอดข้อมูลจากตัวส่งมายังตัวรับเกิดขึ้นอย่างสมบูรณ์โดยตัวส่งจะทำการส่งข้อมูลมา 1 บิตเรียกว่า บิตรับรู้ (Acknowledge bit) มี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สถานะเป็นลอจิกสูง หลังจากที่ส่งข้อมูลมาครบถ้วน ส่วนอุปกรณ์มาสเตอร์จะทำการส่งสัญญาณรับรู้พิเศษซึ่งสัมพันธ์กับสัญญาณนาฬิกาอุปกรณ์สเลฟที่ถูกอ้างถึงในการติดต่อหรือกำลังติดต่ออยู่ ในขณะนั้นก็จะกำเนิดสัญญาณรับรู้ที่มีสถานะลอจิกต่ำเพื่อตอบสนองให้ทราบว่าได้รับข้อมูลเรียบร้อยแล้ว ดังรูปที่ 2.9



รูปที่ 2.9 Acknowledge บนสาย I²C

(5) หยุดการถ่ายทอดข้อมูล (Stop Data Transfer) เกิดขึ้นเมื่อสาย SDA มีการเปลี่ยนแปลงระดับลอจิกจากต่ำไปสูงในขณะที่สาย SCL มีสถานะลอจิกสูง เรียกสภาวะที่เกิดขึ้นว่า สภาวะหยุด (STOP)

2.2.7.1 การทำงานบนบัส I²C

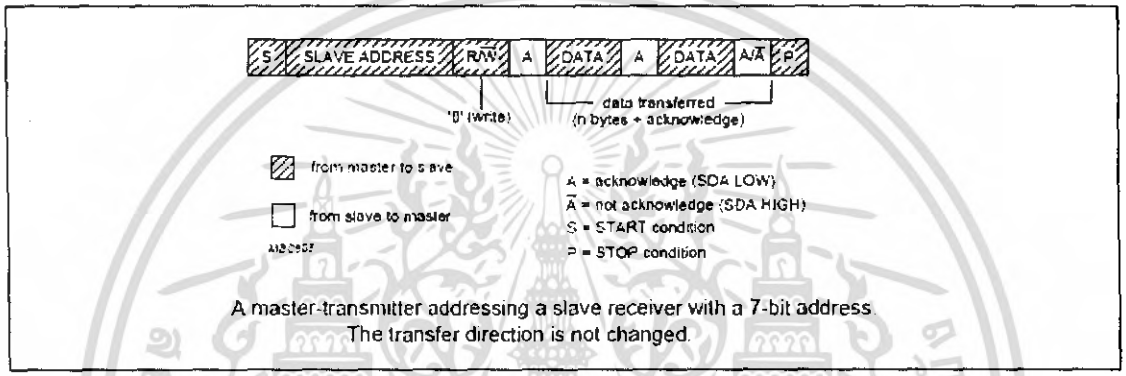
การทำงานบนบัส I2C จะเริ่มต้นด้วยการเข้าถึงอุปกรณ์เสียก่อน โดยการเข้าถึงอุปกรณ์บนบัส I²C นั้นจะใช้การเข้าถึงแบบ 7 บิต หรือ 10 บิต ก็ได้ ในกรณีที่มีอุปกรณ์ต่ออยู่บนบัสไม่มากนัก จะใช้การเข้าถึงแบบ 7 บิต แต่ในกรณีที่ซึ่งมีอุปกรณ์มากก็จำเป็นต้องใช้การเข้าถึงแบบ I0 บิต หลังจากที่ติดต่ออุปกรณ์แต่ละตัวได้เรียบร้อยแล้ว ก็จะเริ่มต้นการถ่ายทอดข้อมูล

2.2.7.2 การเข้าถึงแบบ 7 บิต (7-bit Addressing)

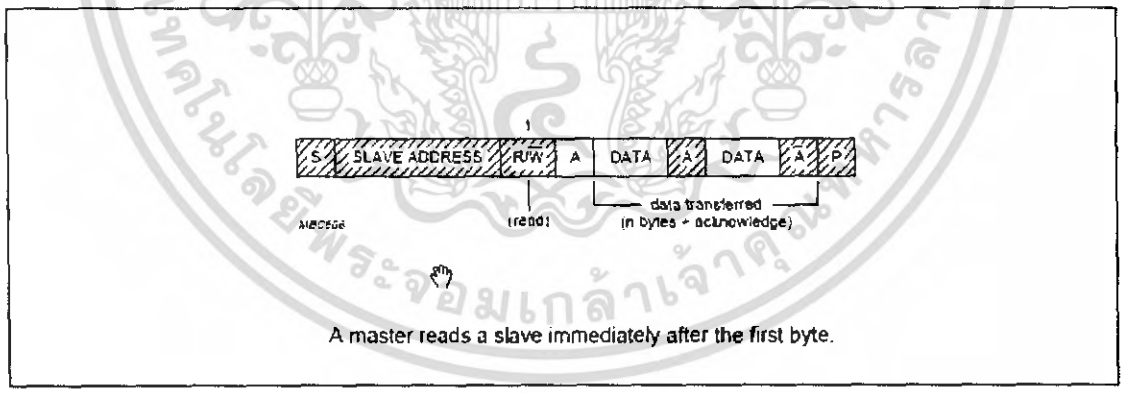
ข้อมูลไบต์แรกที่เกิดขึ้นหลังจากสภาวะเริ่มต้นคือ ข้อมูลที่ใช้ในการอ้างอิงถึงแอดเดรสของอุปกรณ์สเลฟที่ต้องการติดต่อ โดยจะแบ่งเป็นบิตกำหนดคงที่ (Fixed Address bit) จำนวน 4 บิต ซึ่งข้อมูลนี้อุปกรณ์แต่ละตัวจะถูกกำหนดมาจากผู้ผลิต ไม่สามารถเปลี่ยนแปลงแก้ไขได้ ถัดมาอีก 3 บิต จะเป็นบิตกำหนดแอดเดรสที่สามารถโปรแกรมได้ (Programmable Address bit)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยผู้ใช้งานจะต้องกำหนดสถานะลอจิกให้แก่ขา A0 , A1 , A2 ของอุปกรณ์ที่มีการเชื่อมต่อบนบัส I²C ส่วนในบิต LSB จะเป็นบิตที่ใช้กำหนดการอ่านหรือเขียนข้อมูลกับอุปกรณ์สเลฟตัวนั้นๆ หากบิต LSB มีค่าเป็น "0" จะหมายถึงต้องการเขียนข้อมูลไปยังอุปกรณ์สเลฟนั้น ถ้าเป็น "1" จะเป็นการอ่านข้อมูลจากอุปกรณ์สเลฟ ข้อมูลในไบต์ต่อมาคือ ข้อมูลควบคุม (Control byte)ในอุปกรณ์แต่ละตัวนั้นจะมีการกำหนดข้อมูลควบคุมที่แตกต่างกันไป ข้อมูลในไบต์ต่อมาคือ ข้อมูลที่ทำการถ่ายทอดจริง (Data) ซึ่งหลังจากที่มีการถ่ายทอดข้อมูลในแต่ละไบต์แล้ว อุปกรณ์สเลฟที่ได้รับการติดต่อต้องส่งสัญญาณรับรู้ตอบกลับมาด้วยทุกครั้ง โดยที่อาจจะมีการรับ-ส่งข้อมูลหลายๆไบต์ติดต่อกันก็ได้ ตามรูปที่ 3.11และ 3.12 ตามลำดับ จากนั้นจะจบการติดต่อกับอุปกรณ์สเลฟที่อ้างถึงด้วยสภาวะหยุด



รูปที่ 2.10 การส่งข้อมูลแบบ 7 บิตของอุปกรณ์มาสเตอร์



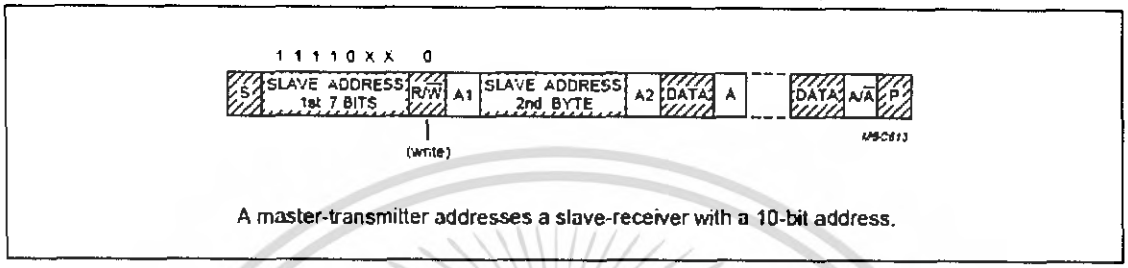
รูปที่ 2.11 การอ่านสัญญาณ Acknowledge ของอุปกรณ์มาสเตอร์แบบ 7 บิต

2.2.7.3 การเข้าถึงข้อมูลแบบ 10 บิต (10-bit Addressing)

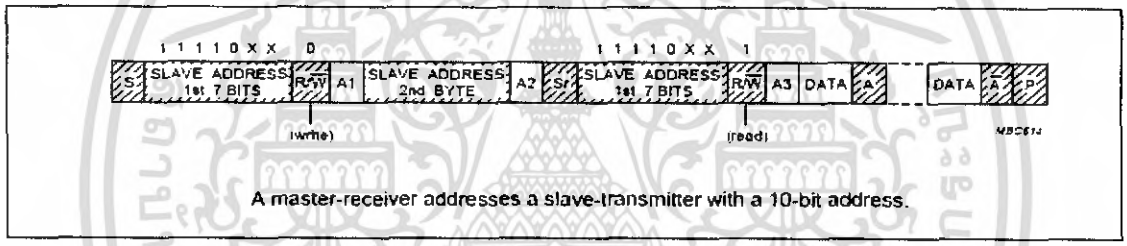
จะเพิ่มเติมจากการเข้าถึงแบบ 7 บิตอีกเล็กน้อย คือ โดยในไบต์แรกหลังจากสภาวะเริ่มต้น ต้องกำหนดให้ 5 บิตบน มีข้อมูลเป็น 11110 ส่วนอีก 2 บิตถัดมาเป็นบิตแอดเดรสของอุปกรณ์ที่ต้องการติดต่อ ในบิต LSB ของข้อมูลไบต์แรกยังคงเป็นการกำหนดว่า ต้องการอ่านหรือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เขียนกับข้อมูลกับอุปกรณ์สเลฟที่ต้องการติดต่อด้วย ข้อมูลไบต์ต่อมาเป็นข้อมูลแอดเดรสในไบต์ที่ 2 ของอุปกรณ์สเลฟที่ต้องการติดต่อด้วย จากนั้นข้อมูลไบต์ถัดไปจึงเป็นข้อมูลควบคุม และก็จะตามด้วยไบต์ของข้อมูลที่ทำการถ่ายทอดจริงและเช่นเดียวกับการเข้าถึงแบบ 7 บิต หลังจากที่มีการถ่ายทอดข้อมูลในแต่ละไบต์แล้ว อุปกรณ์สเลฟที่ได้รับการติดต่อต้องส่งสัญญาณรับรู้ตอบกลับมาด้วยทุกครั้ง โดยที่อาจจะมีการรับ-ส่งข้อมูลหลายๆไบต์ติดต่อกันก็ได้ ตามรูปที่ 2.12และ2.13 ตามลำดับจากนั้นจะจบการติดต่อกับอุปกรณ์สเลฟที่อ้างถึงด้วยสภาวะหยุด



รูปที่ 2.12 การส่งข้อมูลแบบ 10 บิตของอุปกรณ์มาสเตอร์



รูปที่ 2.13 การอ่านสัญญาณ Acknowledge ของอุปกรณ์มาสเตอร์แบบ 10 บิต

2.2.7.4 การเขียนโปรแกรมเพื่อติดต่อกับบัส I²C

การเขียนโปรแกรมเพื่อติดต่อกับบัส I²C นั้น จะต้องเขียนโปรแกรมเพื่อสร้างสภาวะที่เกิดขึ้นบนบัสทั้ง 5 รูปแบบ คือ บัสว่าง (Bus Not Busy) สภาวะเริ่มต้น (START) ข้อมูลดำรงอยู่บนบัส (Data Valid) รับรู้ข้อมูล (Acknowledge) และสภาวะหยุด (STOP) ซึ่งสามารถเขียนโปรแกรมแยกเป็นส่วนต่างๆได้ ดังนี้

```
sbit SDA=P2^0;                    กำหนด P2.0 เป็นสาย SDA
sbit SCL=P2^1;                    กำหนด P2.1 เป็นสาย SCL
void I2C_delay (void)             ฟังก์ชั่นหน่วงเวลา
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        unsigned char i;
        for(i=0;i<20;i++);
}

void I2C_clk (void)                          ฟังก์ชั่นสร้างสัญญาณนาฬิกา (clock) 1 ลูก
{
        SCL=0;
        I2C_delay();
        SCL=1;
        I2C_delay();
        SCL=0;
        I2C_delay();
}

void I2C_start (void)                        ฟังก์ชั่นสร้างสภาวะเริ่มต้น (START)
{
        SCL=0;
        SDA=1;
        SCL=1;
        SDA=0;
        I2C_delay();
}
void I2C_stop (void)                         ฟังก์ชั่นสร้างสภาวะหยุด (STOP)
{
        SCL=0;
        SDA=0;
        SCL=0;
        SDA=1;
        I2C_delay();
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void I2C_ack (void)                    ฟังก์ชั่นสร้างสัญญาณรับรู้ข้อมูล(Acknowledge bit)
                                       มีสถานะเป็นลอจิกต่ำ
{
        SDA=0;
        I2C_clk();
}


void I2C_Nack (void)                   ฟังก์ชั่นสร้างสัญญาณ Not Acknowledge
{                                      มีสถานะเป็นลอจิกสูง
        SDA=1;
        I2C_clk();
}
bit I2C_write (unsigned char dat)      ฟังก์ชั่นเขียนค่าข้อมูลลงบัส I²C
{
        bit send;
        unsigned char i;
        SCL=0;
        for(i=0;i<8;i++)
{                                      วนส่งค่าข้อมูลทีละบิต จนครบทั้ง 8 บิต
        send=dat&0x80;
        SDA=send;
        I2C_clk();
        dat=dat<<1;
 }
        SDA=1;                         สร้างสัญญาณเพื่อรอรับบิตรับรู้ข้อมูล
        I2C_delay();                   (Acknowledge bit)จากตัวรับข้อมูล
        SCL=1;
        I2C_delay();
        send=SDA;                      เก็บค่าสถานะบิตรับรู้ข้อมูล(Acknowledge bit)
        return(send);                  ส่งค่าสถานะบิตรับรู้ข้อมูล(Acknowledge bit)
 }                                     กลับไปยังส่วนที่เรียกใช้ โดยถ้าบิตรับรู้ข้อมูลมี
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สถานะเป็นลอจิกต่ำ แสดงว่าการส่ง-รับข้อมูล
กับตัวรับข้อมูลทำได้สำเร็จแต่ถ้าบิตรับรู้ข้อมูล
มีสถานะเป็นลอจิกสูงแสดงว่ามีความผิดพลาด
ในการส่งข้อมูล

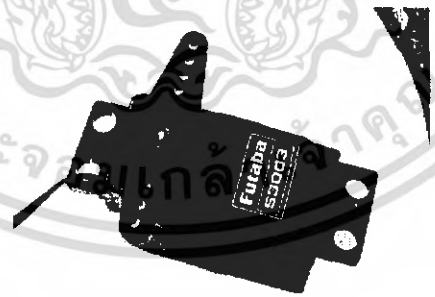
```
unsigned char I2C_read (void)                ฟังก์ชั่นอ่านค่าข้อมูลจากบัส I²C
{
        bit receive;
        unsigned char i;
        unsigned char dat=0x00;
        SDA=1;
        SCL=0;

        for(i=0;i<8;i++)                     วนรับค่าข้อมูลทีละบิต จนครบทั้ง 8 บิต
{
        SCL=1;
        receive=SDA;
        dat=dat<<1;
        dat=dat|receive;
        SCL=0;
}
        SDA=0;                               ส่งสัญญาณบิตรับรู้ข้อมูล (Acknowledge bit)
        I2C_clk();                           มีสถานะเป็นลอจิกต่ำ เพื่อให้ตัวส่งข้อมูลรู้ว่า
                                             ว่าการรับข้อมูลเป็นไปอย่างถูกต้อง
        return(dat);                         คืนค่าข้อมูลที่อ่านได้กลับไปยังส่วนที่เรียกใช้
}
```

ไอซี PCF8591 เป็นไอซีแปลงสัญญาณอะนาลอกเป็นดิจิตอล 4 ช่องทางและแปลงสัญญาณดิจิตอลเป็นอะนาลอก 1 ช่องทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

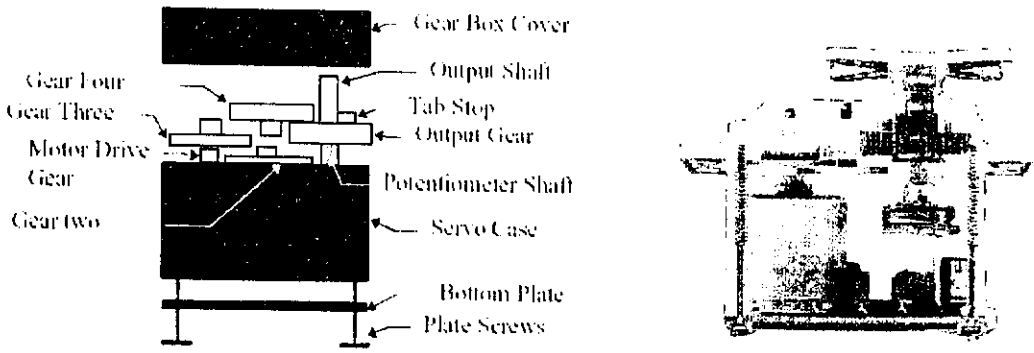
2.3 เซอร์โวมอเตอร์ (Servo motor)

เซอร์โวมอเตอร์ คือ มอเตอร์ไฟฟ้ากระแสตรง (DC motor) ที่ถูกประกอบรวมกับชุดเกียร์และส่วนควบคุมต่าง ๆ ไว้ในโมดูลเดียวกัน โดยมอเตอร์ชนิดนี้จะมีสายต่อใช้งานเพียง 3 เส้นเท่านั้น คือ VCC , GND และสายสัญญาณควบคุม (Control Line) ซึ่งสามารถควบคุมให้มอเตอร์หมุนซ้ายหรือขวาได้จากสายสัญญาณเพียงเส้นเดียว โดยสัญญาณที่ใช้ควบคุมนี้จะเป็นสัญญาณพัลส์วิดมอด (PWM) แบบ TTL Level ระดับแรงดันที่จะจ่ายให้กับมอเตอร์นี้จะอยู่ในช่วงประมาณ 4 ถึง 6 โวลท์ ขึ้นอยู่กับคุณสมบัติของมอเตอร์แต่ละตัว ข้อดีของมอเตอร์ชนิดนี้คือ จะมีขนาดเล็กน้ำหนักเบา ให้แรงบิดสูง กินพลังงานน้อย และสามารถควบคุม ด้วยแรงดันลอจิกที่เป็น TTL ได้โดยตรงไม่จำเป็นต้องต่อวงจรขับ (Driver) อื่นๆเพราะมอเตอร์ชนิดนี้จะมีวงจรควบคุมบรรจุไว้ภายในอยู่แล้ว ซึ่งมอเตอร์ชนิดนี้สามารถควบคุมให้หมุนไปในตำแหน่ง หรือทิศทางองศาที่ต้องการได้ โดยอาศัยสัญญาณความกว้างพัลส์ ที่ป้อนให้มอเตอร์ แต่เซอร์โวมอเดอร์นี้จะหมุนได้แค่ในช่วงประมาณ 180° หรือครึ่งรอบเท่านั้น หรือบางรุ่นอาจหมุนได้ถึง 210° แต่จะไม่สามารถหมุนเป็นวงรอบได้ เนื่องจากโครงสร้างภายในจะประกอบด้วย ตัวต้านทานชนิดปรับค่าได้ (VR) ที่ทำหน้าที่ตรวจสอบการหมุนของมอเตอร์และตัวต้านทานนี้จะถูกยึดติดกับแกนหมุนของมอเตอร์ ซึ่งจากการที่ตัวต้านทานปรับค่านี้ไม่สามารถหมุนเป็นวงรอบได้ ดังนั้น เซอร์โวมอเตอร์จึงถูกออกแบบให้หมุนได้เพียงแค่ประมาณ 180 องศา หรือครึ่งรอบเท่านั้น เพื่อป้องกันความเสียหายที่จะเกิดขึ้นกับตัวต้านทานปรับค่าได้ แต่ถ้าหากเราต้องการให้มอเตอร์หมุนเป็นวงรอบ (360องศา) นั้นก็สามารถทำได้ โดยจะต้องปรับแต่ง (Modify) ดัดแปลงชิ้นส่วนบางอย่างของมอเตอร์ ซึ่งวิธีการต่างๆจะได้กล่าวไว้ในภายหลัง

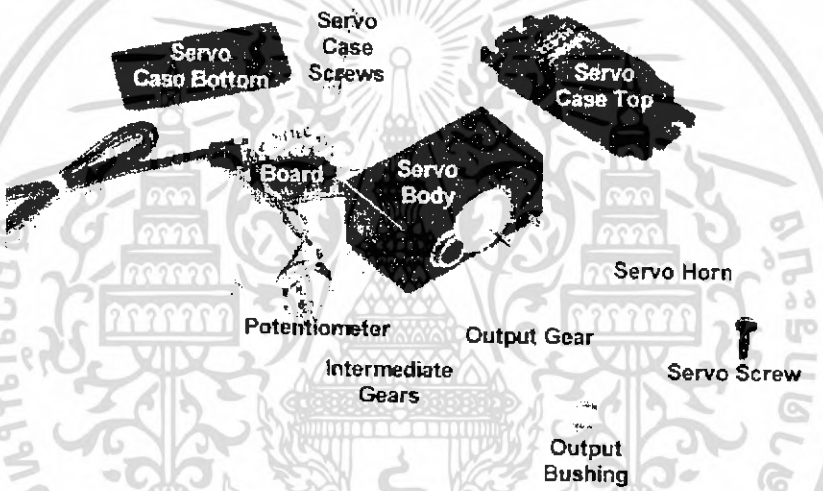


รูปที่ 2.14 แสดงเซอร์โวมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 แสดงส่วนประกอบต่างๆของเซอร์โวมอเตอร์

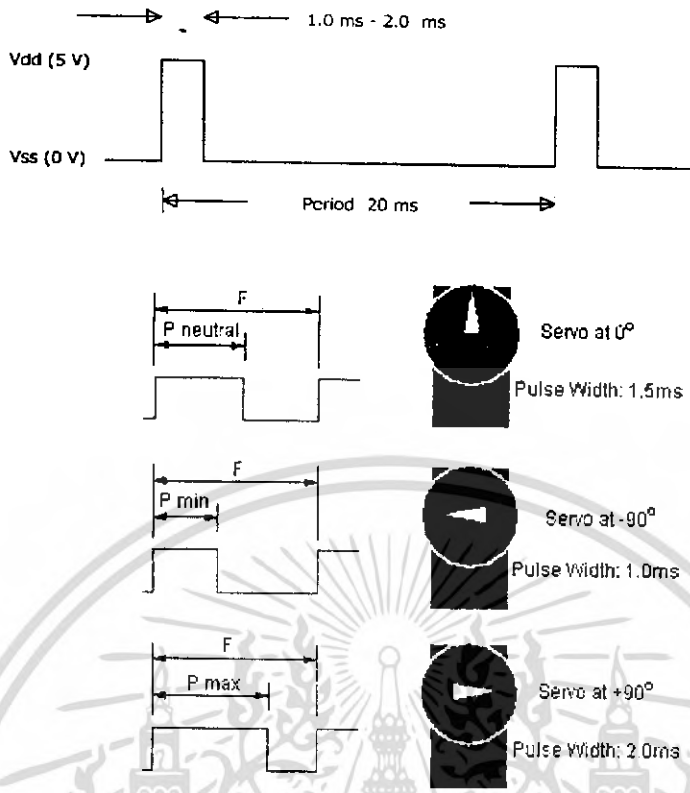


รูปที่ 2.16 แสดงชิ้นส่วนต่างๆของเซอร์โวมอเตอร์

2.3.1 หลักการทำงานของเซอร์โวมอเตอร์

การควบคุมการทำงานของเซอร์โวมอเตอร์ ทำได้โดย การป้อนสัญญาณความกว้างพัลส์ให้กับมอเตอร์ซึ่งตำแหน่งและทิศทางการหมุนของมอเตอร์นี้จะขึ้นอยู่กับขนาดของความกว้างของพัลส์นั้นๆ โดยทั่วไปแล้วความกว้างของสัญญาณพัลส์จะมีจุดให้อ้างอิง 3 จุด ดังรูป คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.17 แสดงความกว้างของสัญญาณพัลส์ที่เวลาต่าง ๆ

- สัญญาณความกว้างพัลส์ขนาด 1.5 ms จะควบคุมให้เซอร์โวมอเตอร์หมุนไปอยู่ที่ตำแหน่งมุม 0 องศา หรือ จุดกึ่งกลางของมอเตอร์

- สัญญาณความกว้างพัลส์ขนาด 1 ms จะควบคุมให้เซอร์โวมอเตอร์หมุนไปอยู่ที่ตำแหน่งมุม -90 องศา หรือ จุดกึ่งกลางของมอเตอร์

- สัญญาณความกว้างพัลส์ขนาด 2 ms จะควบคุมให้เซอร์โวมอเตอร์หมุนไปอยู่ที่ตำแหน่งมุม +90 องศา หรือ จุดกึ่งกลางของมอเตอร์

***หมายเหตุ** ค่าความกว้างพัลส์ และระยะองศาการหมุนของมอเตอร์ ที่อธิบายด้านบน นั้นเป็นเพียงค่าประมาณเท่านั้นทั้งนี้ระยะการหมุน และขนาดของพัลส์ที่ควบคุมการทำงานของมอเตอร์ในแต่ละยี่ห้ออาจจะไม่เท่ากัน ดังนั้นในการใช้งานจึงควรศึกษารายละเอียดของมอเตอร์ในแต่ละรุ่นที่นำมาใช้ซึ่งโดยปกติแล้วรายละเอียดต่างๆของมอเตอร์มักจะมีติดมากับตัวมอเตอร์นั้น ๆ อยู่แล้ว

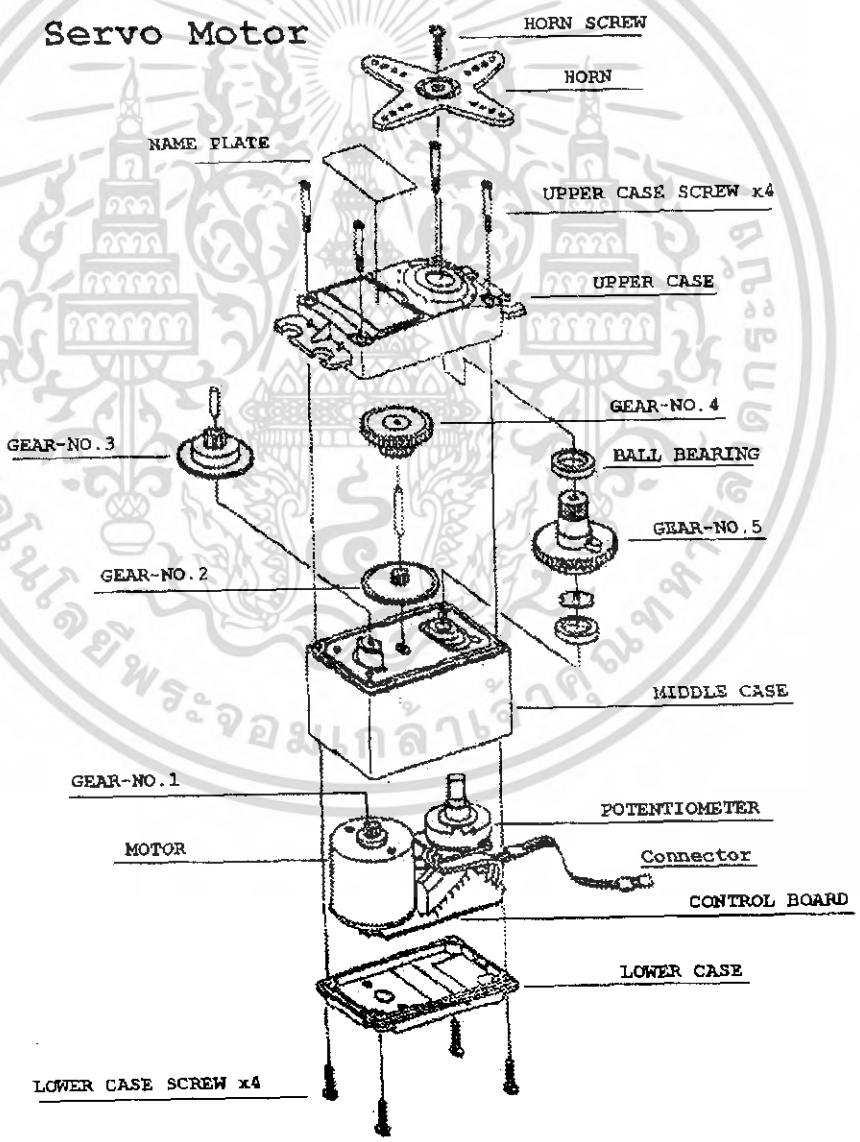
สำหรับเซอร์โวมอเตอร์ยี่ห้อ GWS และ HITEC นั้น จะใช้ระบบเฟืองที่ต่างกันโดยทำให้มีทิศทางการหมุนที่ต่างกันโดยจะตรงข้ามกัน เช่น ส่งสัญญาณพัลส์ 1 ms มอเตอร์ GWS จะหมุนทวนเข็มนาฬิกา ส่วนมอเตอร์ของ HITEC จะหมุนในทิศทางตามเข็มนาฬิกา เป็นต้น

ส่วนการที่จะควบคุมให้มอเตอร์หมุนเป็นมุมอื่นๆ นั้นก็สามารถทำได้โดยการป้อนสัญญาณพัลส์เป็นระดับความกว้างต่างๆโดยอ้างอิงจากจุด ทั้ง 3 จุดที่กล่าวมานี้ ตัวอย่างเช่น ถ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต้องการให้มอเตอร์หมุนไปที่มุม -45 องศา เราก็จะต้องป้อนสัญญาณพัลส์ที่มีความกว้าง 1.25 ms เป็นต้น และสัญญาณพัลส์นี้จะต้องจ่ายให้มอเตอร์ทุก ๆ 20 ms (Period) เพื่อรักษาสภาพตำแหน่งของมอเตอร์ไว้

โดยหลักการก็คือ จะอาศัยการเปรียบเทียบช่วงเวลาของความกว้างพัลส์ที่จ่ายที่ให้กับมอเตอร์ทางขาสัญญาณควบคุมกับค่าเวลาของวงจร RC ภายในบอร์ดควบคุม ในตัวของมอเตอร์ ซึ่งค่าเวลาของวงจร RC นี้จะมีการเปลี่ยนแปลงตามการหมุนของมอเตอร์ เนื่องจาก ตัวต้านทานปรับค่าจะถูกยึดติดกับแกนหมุนของมอเตอร์ ซึ่งการหมุนของมอเตอร์จะทำให้ค่าความต้านทานของตัวต้านทานปรับค่าได้ (VR) เปลี่ยนแปลงไป เป็นผลทำให้ค่าเวลาของวงจรRCเปลี่ยนแปลงจนกระทั่งค่าเวลาความกว้างพัลส์ของวงจรRCเปลี่ยนแปลงจนมีค่าเท่ากับสัญญาณพัลส์ทางขาควบคุม (Control line) มอเตอร์จึงจะหยุดหมุน



รูปที่ 2.18 แสดงชิ้นส่วนต่าง ๆ ของเซอร์โวมอเตอร์อย่างละเอียด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 โมดูลเข็มทิศดิจิตอล (Digital Compass Module)

2.2.1 คุณสมบัติ

- ใช้ไฟเลี้ยง +5V ต้องการกระแสไฟฟ้า 20mA

- ใช้ตัวตรวจจับสนามแม่เหล็กเบอร์ KMZ51 ของ Philips จำนวน 2 ตัวเพื่อให้สามารถตรวจจับสนามแม่เหล็กโลกได้อย่างสมบูรณ์และมีความละเอียดมากเพียงพอ

- ความละเอียดของมุม 0.1 องศา โดยประมาณหลังการปรับแต่ง

- เอาต์พุตแบบสัญญาณพัลส์ ความกว้าง 1 ถึง 37 มิลลิวินาที โดยมีอัตราจะเพิ่มขึ้นครั้งละ 0.1 มิลลิวินาที

- เอาต์พุตข้อมูลดิจิตอลผ่านการติดต่อระบบบัส I²C รองรับสัญญาณนาฬิกาที่ความถี่สูงถึง 1 MHz โดยให้ข้อมูล 2 รูปแบบคือ 0-255 และ 0-3599

- ขนาดเล็กเพียง 32x35 มิลลิเมตร

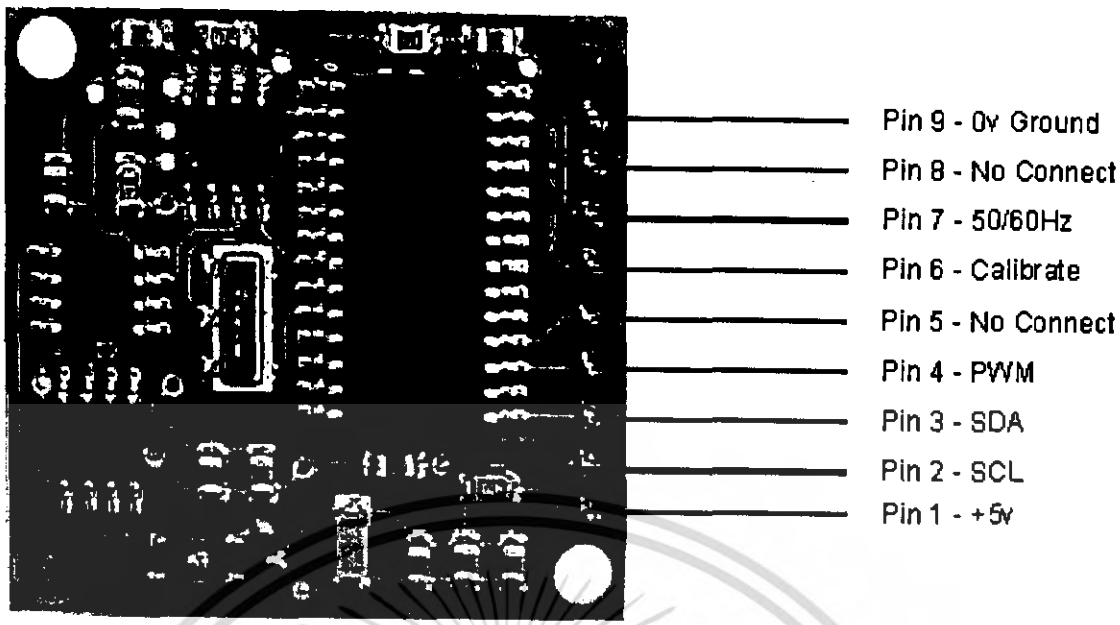
- สื่อสารกับไมโครคอนโทรลเลอร์ยอดนิยมได้ทุกตระกูล อาทิ เบสิกแสตมป์ 2SX/2P,PIC,MCS-51,PSoC,68HC11 ทั้งผ่านระบบบัส I²C และด้วยการวัดสัญญาณพัลส์

โมดูลเข็มทิศดิจิตอล CMPS03 เป็นผลงานของ Devantech (www.radio-electronics.co.uk) ออกแบบมาเพื่อช่วยในการกำหนดทิศทางการเคลื่อนที่ของหุ่นยนต์อัตโนมัติ และนำมาใช้ในการสร้างเครื่องมือวัดและตรวจสอบทิศระบบอิเล็กทรอนิกส์ โดยหัวใจสำคัญของโมดูล CMPS03 คือ ตัวตรวจจับสนามแม่เหล็กเบอร์ KMZ51 ของฟิลิปส์ (Philips) จำนวน 2 ตัว เพื่อให้มีความไวเพียงพอในการตรวจจับสนามแม่เหล็กโลก (Earth magnetic field) และไมโครคอนโทรลเลอร์เพื่อรับสัญญาณจากตัวตรวจจับมาประมวลผลเป็นข้อมูลดิจิตอลและสัญญาณพัลส์แจ้งผลการวัดทิศทาง

2.4.2 ตำแหน่งขาและการต่อใช้งาน

ในรูปที่ 2.18 แสดงรูปร่างหน้าตาและการจัดขาของCMPS03 โมดูลเข็มทิศดิจิตอล จะเห็นว่าเป็นแผงวงจรที่มีตัวคอนเน็กเตอร์ต่อออกมาเพื่อแสดงการเชื่อมต่อไปใช้งาน อย่างไรก็ตามเพื่ออำนวยความสะดวกแก่ผู้ใช้งานกับบอร์ดควบคุมหุ่นยนต์ของบริษัท อินโนเวทิฟเอ็กเพอริเมนต์ จำกัด (i-nex : เป็นตัวแทนจำหน่ายสินค้าของ Devantech ในประเทศไทยอย่างเป็นทางการ) จึงได้พัฒนาบอร์ดอะแดปเตอร์รุ่น ADX-CMPS03 มาติดตั้ง ลงบนบอร์ด ADX-CMPS03 ได้จัดเตรียมคอนเน็กเตอร์ PCB 3 ขาตัวผู้สำหรับเชื่อมต่อกับบอร์ดควบคุมหุ่นยนต์ และคอนเน็กเตอร์ IDC ตัวเมียแถวเดี่ยว 4ขาสำหรับเสียบสายต่อวงจรเบอร์ AWG#22 เพื่อต่อกับแผงต่อวงจรหรือเบรดบอร์ด นอกจากนั้นยังมีสวิตซ์กดสำหรับปรับตั้งค่า (calibration) เพื่อกำหนดตำแหน่งอ้างอิง โดยวงจรบอร์ด ADX-CMPS03 แสดงในรูปที่ 2.19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.19 แสดงบอร์ดโมดูลเข็มทิศ

2.4.3 การปรับแต่งค่าของโมดูล CMPS03 ผ่านทางระบบบัส I²C

การปรับแต่งค่าทำได้โดยการส่งค่า 0xFF ไปยังรีจิสเตอร์ 15 ของโมดูล CMPS03 โดยที่จะต้องส่งค่า 4 ครั้งและจะต้องระบุทิศทางหลัก ๆ 4 ทิศทางเช่นเดียวกันกับการกำหนดค่าด้วยสวิตช์ได้โดยตรงมีขั้นตอนดังนี้

(1) วางโมดูล CMPS03 ขนานกับพื้น หันหน้าของโมดูลไปทางทิศเหนือ จากนั้นเขียนค่า 255(0xFF) ไปยังรีจิสเตอร์ 15

(2) วางโมดูล CMPS03 ขนานกับพื้น หันหน้าของโมดูลไปทางทิศตะวันออก จากนั้นเขียนค่า 255(0xFF) ไปยังรีจิสเตอร์ 15

(3) วางโมดูล CMPS03 ขนานกับพื้น หันหน้าของโมดูลไปทางทิศใต้ จากนั้นเขียนค่า 255(0xFF) ไปยังรีจิสเตอร์ 15

(4) วางโมดูล CMPS03 ขนานกับพื้น หันหน้าของโมดูลไปทางทิศตะวันตก จากนั้นเขียนค่า 255(0xFF) ไปยังรีจิสเตอร์ 15

หลังการปรับแต่งค่าแล้วค่าที่ปรับแต่งจะเก็บไว้ที่หน่วยความจำอีอีพรอม ดังนั้นแม้ไม่จ่ายไฟให้กับตัวบอร์ด ข้อมูลที่ปรับแต่งแล้ว จะยังคงอยู่ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 แสดงตำแหน่งรีจิสเตอร์ของโมดูลเข็มทิศ

ตำแหน่งรีจิสเตอร์	รายละเอียด
0	ตัวเลขแสดงรุ่นบอร์ด CMPS03
1	ส่งค่าตำแหน่งแบบหยาบ (0-255)
2,3	ส่งค่าตำแหน่งแบบละเอียดด้วยตัวเลข 16 บิต (0-3599) สามารถแปลงค่าเพื่อแสดงองศา 0-359.9 องศาได้โดยตรง
4,5	สำหรับตรวจสอบค่าภายใน โดยจะแสดงค่าความต่างของSensor1 เป็นตัวเลข 16 บิตแบบคิดเครื่องหมาย
6,7	สำหรับตรวจสอบค่าภายใน โดยจะแสดงค่าความต่างของSensor2 เป็นตัวเลข 16 บิตแบบคิดเครื่องหมาย
8,9	แสดงค่าตัวเลขการปรับแต่งภายใน (calibration value1)เป็นตัวเลข 16 บิตแบบคิดเครื่องหมาย
10,11	แสดงค่าตัวเลขการปรับแต่งภายใน (calibration value2)เป็นตัวเลข 16 บิตแบบคิดเครื่องหมาย
12,13	ไม่ใช้งานอ่านค่าเป็น 0
14	ไม่ใช้งาน ไม่ได้กำหนดค่าไว้
15	คำสั่งสำหรับปรับแต่งค่า โดยเมื่อต้องการปรับแต่งค่าต้องเขียนข้อมูล 255 เข้าที่รีจิสเตอร์ตำแหน่งนี้

2.5 เครือข่ายไวร์เลสแลน (Wireless Lan)

ครือข่ายไวร์เลสแลนได้มีการถูกพัฒนาขึ้นมาเป็นครั้งแรกในยุค 1980 โดยใช้ความถี่ย่าน 900 เมกะเฮิรตซ์ โดยใช้ความถี่ย่าน ISM Band และด้วยความที่ระบบไวร์เลสนี้ใช้ความถี่เดียวกับระบบโทรศัพท์แบบเซลลูลาร์ ทำให้สามารถใช้งานอุปกรณ์เครื่อง รับ-ส่งได้พร้อมกัน ทำให้สะดวกและประหยัด ทำให้การพัฒนาระบบเครือข่ายแต่เนื่องจากว่าบางประเทศได้มีการสงวนย่านความถี่ 900 เมกะเฮิรตซ์ นี้ไปใช้กับระบบโทรศัพท์มือถือ เครือข่ายไวร์เลสแลน ระบบนี้จึงไม่เป็นที่นิยม

ในช่วงยุคปี 1990 เครือข่ายไวร์เลสแลนย่าน 2.4 กิกะเฮิรตซ์ ก็ได้ถูกพัฒนาขึ้น ก็เพราะความถี่ย่านนี้ เป็นความถี่ที่สูงมาก พัฒนาระบบเครื่องรับส่งได้ยาก ทำให้ไม่มีผู้นิยมที่จะใช้ ต่อมาในปี 1992 องค์กรชื่อว่า IEEE จึงเข้ามาทำหน้าที่กำหนดมาตรฐานเครือข่ายไวร์เลสแลน แล้วตั้งมาตรฐาน 802 .11 เพื่อใช้ระบุมาตรฐานไวร์เลสแลนรุ่นต่าง ๆ ออกมา

มาตรฐานของเครือข่ายไวร์เลสแลนนั้นสามารถแบ่งได้เป็น 3 กลุ่มหลัก ๆ ตามคลื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความถี่ที่ใช้กัน กลุ่มแรกความถี่ 2.4 กิกะเฮิรตซ์ กลุ่มที่สองความถี่ 5 กิกะเฮิรตซ์ ส่วนกลุ่มสุดท้ายใช้แสงอินฟาเรดเพื่อการติดต่อ

2.5.1 เครือข่ายไร้สายตามมาตรฐาน IEEE 802.11

2.5.1.1 มาตรฐาน IEEE 802.11 b

เป็นมาตรฐานเครือข่ายไวร์เลสแลนที่มีผู้ใช้มากที่สุด โดยได้มีการใช้งานในย่านความถี่ 2.4 กิกะเฮิรตซ์ อัตราความเร็วสูงสุดอยู่ที่ 11 เมกะบิตต่อวินาที ระยะการทางสูงสุดอยู่ที่ประมาณ 100 เมตร

2.5.1.2 มาตรฐาน *IEEE 802.11 g*

เป็นมาตรฐานเครือข่ายไวร์เลสแลนที่ ใช้งานในย่านความถี่2.4 กิกะเฮิรตซ์ อัตราความเร็วสูงสุดอยู่ที่ 54 เมกะบิตต่อวินาที ระยะการทางสูงสุดอยู่ที่ประมาณ 100 เมตร

2.5.1.3 มาตรฐาน IEEE 802.11 a

เป็นมาตรฐานตัวเก่าที่ไม่ค่อยจะได้รับความนิยม เนื่องจากการใช้งานในย่านความถี่ 5 กิกะเฮิรตซ์ อัตราความเร็วสูงสุดอยู่ที่ 54 เมกะบิตต่อวินาที ซึ่งเป็นย่านที่ไม่ได้เปิดใช้งานอย่างเสรี ในทุกประเทศ และยังไม่สามารถ ใช้งานร่วมกับ อุปกรณ์ตามมาตรฐาน 802.11g และ 802.11b ได้

2.5.2 อุปกรณ์ที่ใช้ในเครือข่ายไร้สาย

2.5.2.1 แลนการ์ดไร้สาย (Wireless Lan Card) เป็นอุปกรณ์ที่ติดตั้งใส่คอมพิวเตอร์ทำให้สามารถสื่อสารข้อมูลถึงการได้โดยไม่ต้องใช้สายสัญญาณเป็นสื่อกลาง เหมือนกับระบบอีเธอร์เนท (Ethernet Lan) หน้าที่หลักของแลนการ์ดไร้สายคือ แปลงข้อมูลดิจิตอลที่ได้จากการประมวลผลของเครื่องคอมพิวเตอร์ให้เป็นคลื่นวิทยุ แล้วส่งผ่านเสาอากาศ (attenna) แพร่กระจายออกไป และในทางกลับกันก็จะทำหน้าที่รับเอาคลื่นวิทยุที่แพร่กระจายออกมาจากอุปกรณ์ไร้สายอื่นๆแปลงกลับเป็นข้อมูลดิจิตอลให้เครื่องคอมพิวเตอร์ประมวลผลต่อไป แลนการ์ดไร้สายที่ผลิตออกมามีหลากหลายรูปแบบตามลักษณะช่องเชื่อมต่อของคอมพิวเตอร์ เช่นแบบ PCI, PCMCI , USB และ Compact Flash

2.5.2.2 อุปกรณ์เข้าใช้งานเครือข่าย (Wireless Access Point) เป็นอุปกรณ์ที่ทำหน้าที่เสมือนฮับ เชื่อมเครื่องคอมพิวเตอร์เข้ากับอุปกรณ์ไวร์เลสแลนอื่นๆเข้าด้วยกันเป็นเครือข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อีกทั้งเป็นสะพานเชื่อมต่อเครือข่ายไวร์เลสแลน เข้ากับ เครือข่ายอีเธอร์เน็ท ทำให้อุปกรณ์บนระบบทั้งสองสามารถสื่อสารข้อมูลถึงกันได้

2.5.2.3 สะพานเชื่อมโยงไร้สาย (Wireless Bridge) เป็นอุปกรณ์ที่ใช้เป็นตัวกลางเชื่อมโยงระบบเครือข่ายอีเธอร์เน็ทแลนตั้งแต่สองระบบขึ้นไปเข้าด้วยกันแทนการใช้สายสัญญาณ ข้อมูลที่สื่อสารระหว่างเครือข่ายอีเธอร์เน็ทแลนจะถูกเปลี่ยนเป็นคลื่นวิทยุและส่งไปยังปลายทางโดยสะพานเชื่อมโยงไร้สาย การเชื่อมโยงระหว่างระบบอีเธอร์เน็ทแลนด้วย สะพานเชื่อมโยงไร้สาย สามารถแบ่งเป็นสองแบบดังนี้ 1. การเชื่อมโยงแบบจุดไปจุด 2 . การเชื่อมโยงแบบจุดไปหลายจุด

2.5.2.4 อุปกรณ์ Wireless Brodband Router ถูกพัฒนาขึ้นมาเพื่อตอบสนองผู้ใช้งานเครือข่ายไวร์เลสแลนที่ต้องการเข้าระบบอินเตอร์เน็ทความเร็วสูงผ่านคู่สายโทรศัพท์ หรือเคเบิลทีวี และอุปกรณ์ Wireless Broadband ได้นำเอาเทคโนโลยี Broadband Router ซึ่งมีฟังก์ชั่นการทำงานที่เป็นตัวค้นหาเส้นทาง ,NAT (Network Address Translation) , Firewall , VPN ฯลฯ มาผสมผสานเข้ากับ อุปกรณ์เข้าใช้งานเครือข่าย (Access Point)

1. อุปกรณ์ Wireless PrintServer อุปกรณ์นี้ทำหน้าที่เป็นเครื่องพิมพ์กับ ผู้ใช้งานเครื่องคอมพิวเตอร์ไร้สาย เมื่อมีการสั่งพิมพ์เอกสารจากเครื่องคอมพิวเตอร์ไร้สาย ข้อมูลจะถูกแปลงให้เป็นคลื่นและส่งแพร่กระจายไปบนเครือข่ายไวร์เลสแลนจนกระทั่งไปถึง Wireless Print Server คลื่นจะถูกแปลงกลับให้เป็นข้อมูลต้นฉบับพิมพ์ออกเครื่องพิมพ์

2. อุปกรณ์ Wireless Signal Booster เป็นอุปกรณ์ที่ทำหน้าที่ ขยายสัญญาณ ที่ส่งออกมาจาก อุปกรณ์เข้าใช้งานเครือข่าย (Access Point) ให้มีกำลังสูงขึ้น ผลคือสามารถส่งสัญญาณออกไปไกลมากขึ้น

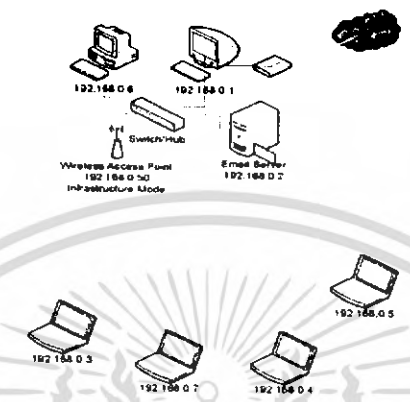
ในการรับ-ส่งสัญญาณของระบบไวร์เลสแลนนั้นจะใช้ความถี่เดียวกัน ซึ่งอุปกรณ์ทุกๆตัว ไม่ว่าจะเป็นอุปกรณ์เข้าใช้งานเครือข่าย (Access Point) หรือ การ์ดไวร์เลสแลนก็จะรับ และส่งข้อมูลที่ความถี่นี้ และจะต้องผลัดกันรับ-ส่งข้อมูลไม่สามารถส่งข้อมูลพร้อมกันได้ ทำให้เกิดการชนของข้อมูลในอากาศ ซึ่งการส่งแบบนี้ก็คือวิธีการแบบ ฮาลฟ์ดูเพล็กซ์ (Half-Duplex Mode) และโปรโตคอลที่ใช้คือ CSMA/CA (Carrier Sense Multiple Acccess With Collision Avoidance) ซึ่งมีหลักการทำงาน คือ เมื่อผู้ส่งต้องการส่งข้อมูลก็จะมีการส่งสัญญาณร้องขอ (Request To Send) ฝ่ายรับก็ต้องตรวจสอบช่องสัญญาณว่าว่างพร้อมจะส่งหรือไม่แล้วตอบกลับด้วยสัญญาณพร้อมให้ส่ง (Clear To Send) จากนั้นก็เริ่มส่งข้อมูล เมื่อฝ่ายรับได้รับข้อมูลครบก็ตอบกลับว่าได้รับข้อมูลแล้วด้วยสัญญาณ Acknowledge จากที่กล่าวมานั้น ทั้งการส่งข้อมูลแบบ ฮาลฟ์ดูเพลกซ์ และโปรโตคอลแบบ CSMA/CA จะส่งผลทำให้ความเร็วในการใช้งานจริงจะมีค่าต่ำกว่าค่าที่กำหนดไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เช่นมาตรฐาน IEEE 802.11g/a ซึ่งมีความเร็วตามที่ระบุไว้ 54 เมกะบิตต่อวินาที แต่เมื่อเวลาใช้งานจริงจะมีความเร็วอยู่ที่ประมาณ 26-27 เมกะบิตต่อวินาทีเท่านั้น

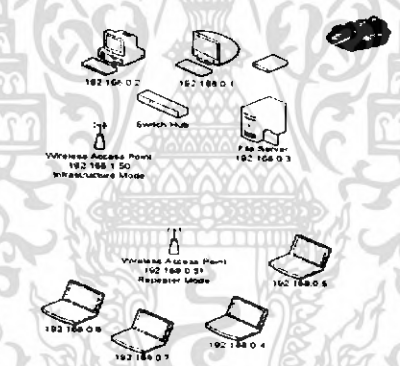
2.5.3 รูปแบบการเชื่อมต่อต่างๆของเครือข่ายไร้สาย

1. Access Point Mode



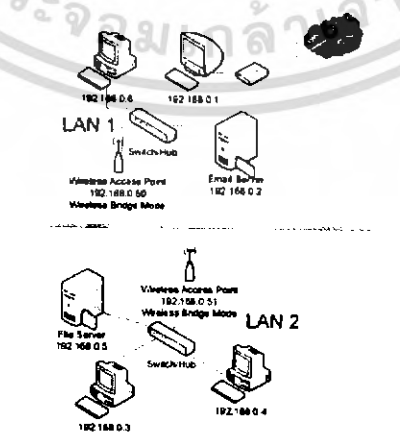
รูปที่ 2.20 Access Point Mode

2. Repeater Mode



รูปที่ 2.21 Repeater Mode

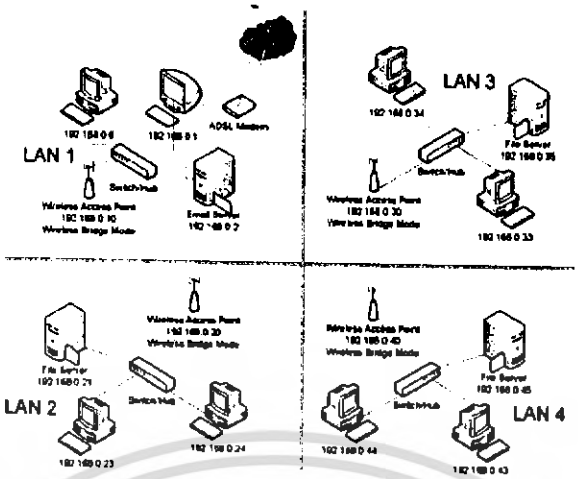
3. Wirelessbridge Mode



รูปที่ 2.22 Wirelessbridge Mode

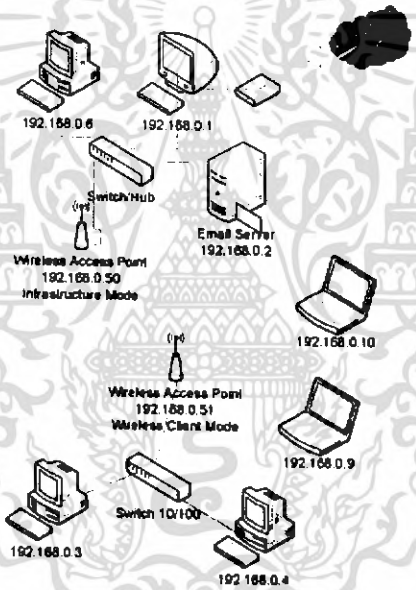
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. **Multiportbridge Mode**



รูปที่ 2.23 Multiportbridge Mode

5. **Wireless Client Mode**



รูปที่ 2.24 Wireless Client Mode

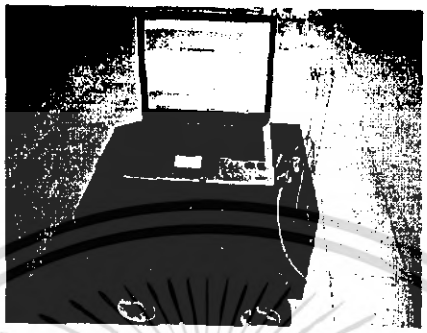
2.5.4 ลักษณะเครือข่ายไร้สายที่ใช้ในการทดลองหุ่นยนต์เคลื่อนที่

ในเครือข่ายไร้สายของหุ่นยนต์เคลื่อนที่นี้จะใช้ อุปกรณ์เข้าใช้งานเครือข่าย (Access point) ยี่ห้อ Linksys รุ่น WRT54G

รูปที่ 2.25 แสดงอุปกรณ์เข้าใช้งานเครือข่าย (Access point) ยี่ห้อ Linksys รุ่น WRT54G

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และในเครือข่าย จะประกอบด้วยเครื่องคอมพิวเตอร์ลูกข่าย (Client) และคอมพิวเตอร์ แม่ข่าย (Server) โดยคอมพิวเตอร์ลูกข่ายนี้ จะเป็นคอมพิวเตอร์ใดๆก็ได้ที่อยู่ในเครือข่าย ส่วนคอมพิวเตอร์แม่ข่ายจะใช้คอมพิวเตอร์โน้ตบุค ยี่ห้อ Toshiba รุ่น Satellite A 70 ติดไว้ที่ตัวหุ่นยนต์เคลื่อนที่



รูปที่ 2.26 แสดงการติดตั้งคอมพิวเตอร์แม่ข่ายกับตัวหุ่นยนต์

2.5.5 การโปรแกรมคอมพิวเตอร์

งานวิจัยนี้จะควบคุมหุ่นยนต์เคลื่อนที่โดยใช้โปรแกรม Visual Basic 6 และใช้ control ชื่อ Winsock ในการทำงานด้านเครือข่ายใช้โปรโตคอล TCP/IP โดยในส่วนของโปรแกรมจะแบ่งได้เป็น 2 ส่วนหลักคือ 1. ส่วนลูกข่าย (Client) 2 . ส่วนแม่ข่าย (Server)

2.5.6 พื้นฐานต่างๆของการเชื่อมโยง TCP

TCP ช่วยสร้าง Application ที่ใช้ Protocal ในการเชื่อมโยงกับคอมพิวเตอร์ที่อยู่ห่างไกลโดยใช้การเชื่อมโยงที่คอมพิวเตอร์ 2 เครื่องสามารถส่งผ่านข้อมูลซึ่งกันและกัน จึงเหมาะที่จะใช้งานในด้านการสื่อสารโดยตรงระหว่างผู้ใช้และส่งผ่าน File ขนาดใหญ่ การสร้าง Application ของ Server จะกำหนด Port (Property Local Port) ให้รับฟังและเรียกใช้ Method Listen เมื่อ Client ร้องขอการเชื่อมโยงทำให้เกิดเหตุการณ์ Connect Request เมื่อมีการเชื่อมโยงคอมพิวเตอร์ทั้งคู่ สามารถรับและส่งข้อมูลจะเรียกใช้Method Send Data เมื่อมีได้รับข้อมูลเหตุการณ์ Data Arrival จะเกิดขึ้นซึ่งจะเรียกใช้ Method Get Data ภายในเหตุการณ์ Data Arrival เมื่อนำข้อมูลออกมาแสดง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.7 Winsock Control

คอนโทรลที่มีชื่อว่า Winsock ที่มากับ Visual Basic เป็นเครื่องมือที่มีประโยชน์ในการพัฒนาโปรแกรมบน Internet เมื่อเปรียบเทียบกับคอนโทรล ActiveX ตัวอื่น ๆ Winsock จะเป็นตัวที่ถูกใช้งานอย่างมาก

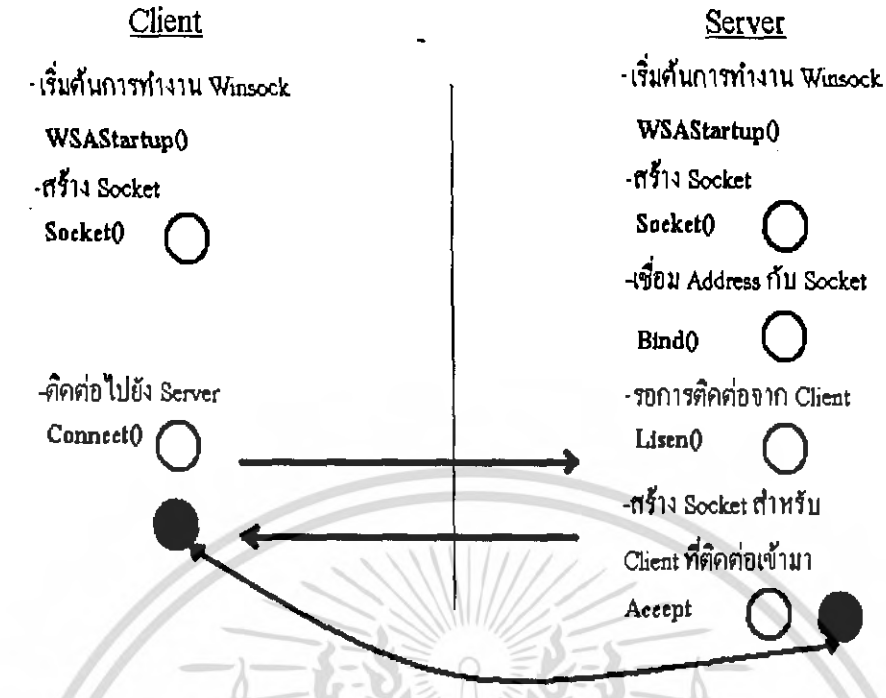
ไคลเอ็น (Client) และ เซิร์ฟเวอร์ (Server) ไคลเอ็นจะเป็นเครื่องคอมพิวเตอร์ที่จะร้องขอการบริการหรือข้อมูล จากเครื่องเซิร์ฟเวอร์ โดยที่จะผ่านทางโปรแกรมต่าง ๆ เช่น Web Browse , Outlook Express , mIRC ซึ่งเครื่องเซิร์ฟเวอร์ จะเป็นเครื่องที่ซึ่งให้บริการด้านต่างๆ แก่เครื่องที่เป็นไคลเอ็น เช่น Web Server , Mail server , IRC Server โดยมากโปรแกรมทางอินเตอร์เนตจะถูกติดตั้งลงในเครื่องที่เป็นไคลเอ็น ทำให้เครื่องที่เป็นไคลเอ็นมีความสามารถในการ รับส่ง อีเมล์ เปิดชมเว็บไซต์ ทำการติดต่อสื่อสารกับกลุ่มข่าว (news groups) ต่างๆและสามารถดาวโหลดไฟล์ที่ต้องการ ส่วนโปรแกรมอีกประเภท จะต้องติดตั้งกับเครื่องที่เป็นเซิร์ฟเวอร์ โปรแกรมประเภทนี้จะให้บริการแก่ไคลเอ็นที่ร้องขอใช้บริการ เครื่องเซิร์ฟเวอร์สามารถที่จัดการกับเครื่องไคลเอ็นได้หลายเครื่องพร้อมกัน และยังสามารถจัดการกับงานต่างๆที่อยู่บนเครื่องไปพร้อมกัน

เทคโนโลยีซ็อกเก็ต (Socket) ที่มีใช้งานอยู่ในปัจจุบันบนอินเตอร์เน็ตที่มีความเสถียรภาพ Socket ที่มีการใช้งานอยู่บน MS Windows จะถูกเรียก Window Socket หรือจะเรียกสั้นๆว่า วินซ็อก (Winsock) ความหมายของซ็อกเก็ต คือ เครื่องมือของโปรแกรมที่จะถูกใช้ ในการส่งและรับ ข้อมูลผ่านทางหมายเลขพอร์ทของ TCP/IP ที่กำหนด โปรแกรมจะสร้างซ็อกเก็ตได้ตามที่ต้องการเพื่อใช้ในการทำงาน แต่ 1 ซ็อกเก็ต จะต้องทำงานกับ 1 พอร์ทของ TCP/IP เท่านั้น

2.5.7.1 การสร้าง Client/server โดยใช้ Winsock

การสร้าง Client/Server โดยใช้ฟังก์ชันของWinsock สามารถทำได้ดังรูปที่ 2.27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.27 การสร้าง Client/Server โดยใช้ Winsock

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบและการสร้าง

3.1 ส่วนประกอบระบบของหุ่นยนต์เคลื่อนที่อัตโนมัติ

1. หุ่นยนต์เคลื่อนที่
2. ไมโครคอนโทรลเลอร์ ยี่ห้อ Phillip รุ่น P89C51RD2
3. เราท์เตอร์ ยี่ห้อ Linksys รุ่น WRT54G
4. โฟโต้ทรานซิสเตอร์ ชนิด PNP
5. จานเข้ารหัส ชนิด 60 ช่อง
6. โมดูลเข็มทิศแม่เหล็ก CMPS03
7. เซอร์โวมอเตอร์ยี่ห้อ Futaba รุ่น S3003 จำนวนสองตัว
8. กล้องเว็บแคมยี่ห้อ Logitech
9. ชุดพาวเวอร์ซัพพลาย
10. คอมพิวเตอร์โน้ตบุคยี่ห้อ Toshiba ใช้เป็นเครื่องแม่ข่าย และโปรแกรมควบคุม
11. คอมพิวเตอร์ส่วนบุคคลทั่วไปใช้เป็นเครื่องลูกข่าย และโปรแกรมควบคุม

3.2 ไมโครคอนโทรลเลอร์

เป็นไมโครคอนโทรลเลอร์ยี่ห้อ Philip รุ่น P89C51RD2 ซึ่งมีรายละเอียดดังนี้

- เป็นไมโครคอนโทรลเลอร์ขนาด 8 บิต
- มีหน่วยความจำโปรแกรมสูงถึง 64 กิโลไบต์
- หน่วยความจำข้อมูลแรมภายในมีขนาด 1 กิโลไบต์
- ความถี่สัญญาณนาฬิกาสูงสุด 33 MHz ซึ่งในกรณีที่จะทำงานด้วยสัญญาณนาฬิกาภายในที่ใช้ 12 ลูกต่อแมชีนไซเกิลและ 20 MHz จะเห็นได้ว่าในกรณีที่ทำงานด้วยสัญญาณนาฬิกาภายในที่ใช้ 6 ลูกต่อแมชีนไซเกิล
- ขาพอร์ต 8 บิต 4 พอร์ตแบบ 2 ทิศทาง (quasi-bidirectional) ซึ่งจะสามารถเป็นได้ทั้งอินพุตและเอาต์พุท
- มีวงจรสื่อสารอนุกรมแบบฟูลดูเพล็กซ์
- ไทเมอร์/เคาน์เตอร์ขนาด 16 บิต 3 ตัว
- สามารถรองรับแหล่งกำเนิดอินเตอร์รัปต์ได้ 7 ประเภท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มีโมดูลวงจรที่นับโปรแกรมได้ (PCA : Programmable Counter Array) ซึ่งบรรจุวงจรตรวจจับสัญญาณ (Capture) ที่ใช้ในการเปรียบเทียบสัญญาณ (Compare) และวอตช์ด็อกไทม์เมอร์ (Watchdog timer)

การจัดสรรบิตต่างๆสำหรับเป็นอินพุต และเอาท์พุทในแต่ละพอร์ตของโครงงาน

พอร์ต 0	P0.0-0.7	ไม่ได้ต่อใช้งาน
พอร์ต 1	P1.0-1.2	ไม่ได้ต่อใช้งาน
	P1.3	ใช้เป็นขาตรวจสัญญาณขอบขาลงจากโฟโต้ทรานซิสเตอร์
	P 1.4	ใช้เป็นขาตรวจสัญญาณขอบขึ้นลงจากโฟโตทรานซิสเตอร์
	P1.5	ใช้เป็นสัญญาณที่ส่งไปควบคุมเซอร์โวมอเตอร์ขึ้น-ลง
	P1.6	ใช้เป็นสัญญาณที่ส่งไปควบคุมเซอร์โวมอเตอร์ซ้าย-ขวา
	P1.7	ไม่ได้ต่อใช้งาน
พอร์ต2	P2.0	ใช้เป็นขา SDA ติดต่อกับเข็มทิศแม่เหล็ก
	P2.1	ใช้เป็นขา SCA ติดต่อกับเข็มทิศแม่เหล็ก
	P2.2-2.7	ไม่ได้ต่อใช้งาน
พอร์ต3	P3.0-3.7	ไม่ได้ต่อใช้งาน

ในส่วนของการทำงานนั้นจะใช้ โฟโต้ทรานซิสเตอร์ ร่วมกับจานเข้ารหัส เพื่อวัดระยะทาง ซึ่งเอาท์พุทจากโฟโตทรานซิสเตอร์ จะมีลักษณะเป็นพัลส์ ไมโครคอนโทรลเลอร์จะทำการนับที่ขอบขาลง ค่าพัลส์ที่ได้จะส่งไปให้ส่วนของโปรแกรมVB คำนวณต่อไป ส่วนทางด้านการวัดทิศทางโมดูลเข็มทิศจะทำการส่งเอาท์พุทมาเป็นค่าตัวเลขฐานสิบซึ่งไมโครคอนโทรลเลอร์จะทำการนับค่าและส่งให้โปรแกรมVB ต่อไป และในส่วนของการควบคุมเซอร์โวมอเตอร์จะใช้โมดูล PCA ส่ง Output ที่เป็นสัญญาณพัลส์ออกไปจากนั้นนำไปผ่านไอซี 74LS244N ซึ่งภายในเป็นวงจรบัฟเฟอร์เพื่อช่วยเพิ่มกระแส และจากนั้นจึงค่อยนำไปเข้า เซอร์โวมอเตอร์

3.3 ส่วนวัดระยะทาง

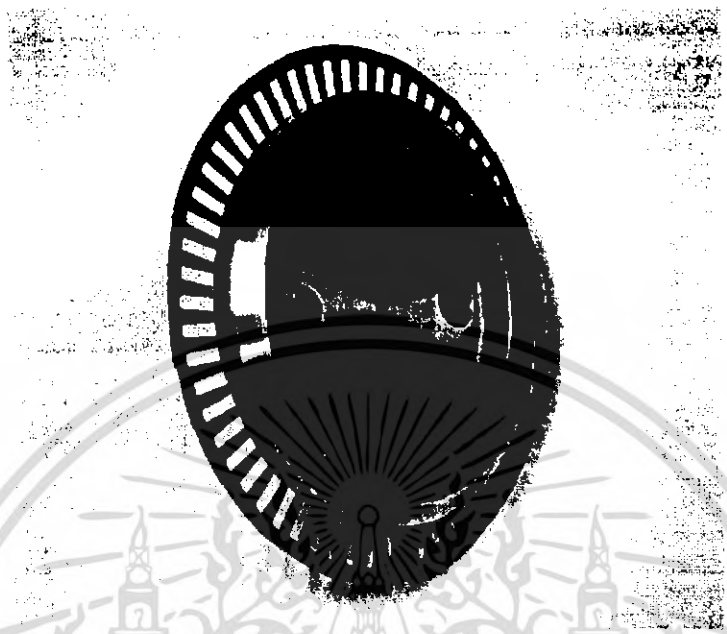
ส่วนวัดระยะทางในหุ่นยนต์นี้ประกอบไปด้วย

1. จานเข้ารหัส (Encoder)
2. โฟโตทรานซิสเตอร์ (Photo Transistor)

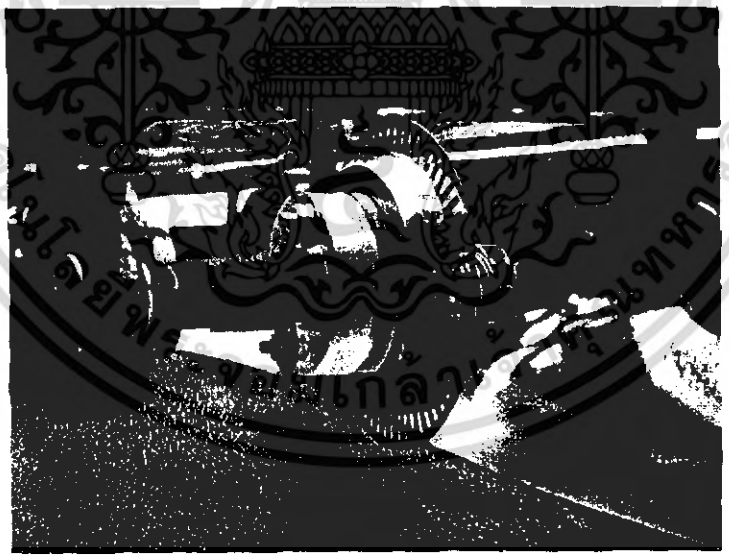
จานเข้ารหัสนี้จะมีลักษณะดังรูปที่ 3.1 โดยจะมีทั้งหมด 60 ช่องเล็ก จากการทดลองจะได้ว่า จานเข้ารหัสขับไป1 ช่องเล็กจะเทียบการเคลื่อนที่เป็นระยะทางของหุ่นยนต์ = 0.007833333 เมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรือเมื่อจานเข้ารหัสหมุนไป 1 รอบ จะทำให้หุ่นยนต์เคลื่อนที่ไปเป็นระยะทาง 0.45 เมตร และจะทำการติดตั้งเข้ากับบริเวณ เพลาซ้ายของหุ่นยนต์ดังรูปที่ 3.2



รูปที่ 3.1 จานเข้ารหัส (Encoder)

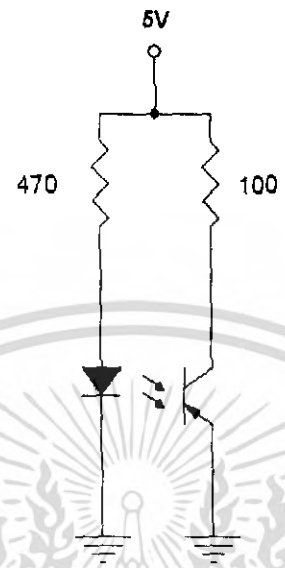


รูปที่ 3.2 การติดจานเข้ารหัสเข้ากับเพลารถ

Photo Transistor เป็นTransistorชนิด PNP จะมีลักษณะดังรูปที่ 3.4 ซึ่งจะประกอบไปด้วยสายไฟสี่เส้นอันได้แก่ 1. VCC 5V. 2 . Ground 3. Ground 4. Output โดยจะให้ Output = 5V. เมื่อไม่มีอะไรมากั้นระหว่าง led และ Transistor แต่เมื่อมีสิ่งกีดขวางจะให้ Output = 0 V. ดังนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะมีการนำเอาขอบขาลงของสัญญาณ มาใช้งานในการวัดระยะทางต่อไป ซึ่งการต่อวงจรจะเป็นไปดังรูปที่ 3.3



รูปที่ 3.3 วงจรโฟโต้ทรานซิสเตอร์

รูปที่ 3.4 โฟโต้ทรานซิสเตอร์ (Photo Transistor)

3.4 ส่วนวัดทิศทาง

ในส่วนของการวัดทิศทางจะนำเอาโมดูลเข็มทิศแม่เหล็ก CMPS03 ซึ่งจะทำงานบนระบบบัสที่ชื่อว่า I²C ใช้สายสองเส้นในการติดต่อกับ MCS-51 ได้แก่ สาย SCL หรือสายสัญญาณนาฬิกา และสายที่ 2 สาย SDA หรือสายที่ทำหน้าที่รับและส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.1 การอ่านค่าทิศทางเป็นข้อมูลดิจิตอลผ่านระบบบัส I²C

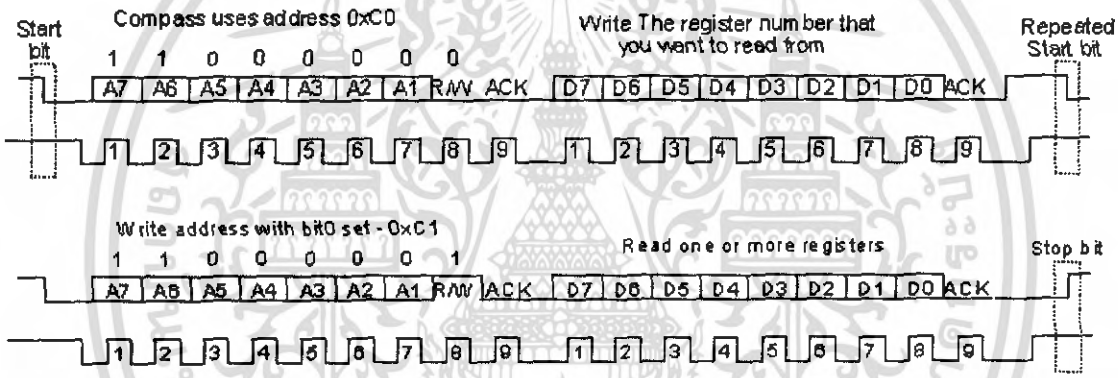
การอ่านค่าจากโมดูล CMPS03 ให้ได้ค่าที่มีความแม่นยำสูงควรเลือกเอาต์พุตข้อมูลดิจิตอลผ่านระบบบัส I²C โดยโมดูล CMPS03 จะสามารถส่งข้อมูลของตำแหน่งออกมาที่ความละเอียดสูงสุด 0.1 องศาโดยไม่จำเป็นต้องมีการคำนวณหรือแปลงค่าใด ๆ อีก

3.4.2 รูปแบบการการสื่อสารข้อมูลบัส I²C

บัส I²C จะทำการเชื่อมต่อกับไมโครคอนโทรลเลอร์ได้โดยใช้สายสัญญาณ 2 เส้น ซึ่งจะได้แก่ขา SDA (รับและส่งข้อมูล) และ SCL (ขาสัญญาณนาฬิกา) โดยขาสัญญาณทั้งสองจะต้องต่อตัวต้านทานพูลอัปต่อไว้เพื่อกำหนดสถานะโลจิก "1" ให้กับระบบบัส

3.4.3 ลำดับขั้นการติดต่อ

ค่าแอดเดรสของโมดูล CMPS03 คือ $C0 สำหรับการส่งข้อมูล และ $C1 สำหรับการอ่านค่าข้อมูล โดยขั้นตอนการติดต่อกับโมดูล CMPS03 เพื่ออ่านข้อมูลมีดังนี้



รูปที่ 3.5 แสดงไทมิ่งไดอะแกรมของการติดต่อสื่อสารกับโมดูล CMPS03 ผ่านระบบบัส I²C

1. ส่งบิตเริ่มต้นหรือสตาร์ทบิต (Start Bit) เพื่อแจ้งให้ระบบบัส I²C เตรียมพร้อมรับข้อมูล
2. ส่งค่าแอดเดรส $C0 เพื่อระบุว่าต้องการติดต่อเพื่อเขียนข้อมูลไปยังกับโมดูล CMPS03
3. ส่งค่าตำแหน่งรีจิสเตอร์ภายในโมดูล CMPS03 ที่ต้องการอ่านค่า โดยมีรายละเอียดดังแสดงในตารางที่ 2.3
4. ส่งค่าแอดเดรส $C1 เพื่อระบุว่าต้องการอ่านค่าข้อมูลจากโมดูล CMPS03
5. อ่านค่าข้อมูลจากโมดูล CMPS03 มาเก็บไว้ในหน่วยความจำ
6. ส่งบิตหยุดหรือสต๊อป (Stop) เพื่อหยุดการสื่อสารข้อมูล และจะต้องกำหนดให้บัสอยู่ในสถานะบัสว่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 การทำงานของ VB

ในโครงงานนี้ใช้ภาษา Visual Basic เพื่อทำหน้าที่ติดต่อระหว่างผู้ใช้งานและตัวหุ่น โดยจะใช้คอนโทรล winsock เพื่อติดต่อผ่านระบบเครือข่ายซึ่งเป็นระบบเครือข่ายแบบไร้สาย และใช้คอนโทรล mscomm เพื่อติดต่อกับตัวหุ่นผ่านสาย RS-232 ซึ่งเป็นการติดต่อผ่านพอร์ตอนุกรม สามารถแบ่งการทำงานได้เป็นสองส่วน คือ ส่วนของเครื่องแม่ข่าย (Server) และเครื่องลูกข่าย (Client)

3.5.1 คอนโทรล Winsock

คอนโทรลนี้มีไว้เพื่อพัฒนาระบบติดต่อสื่อสารผ่านระบบเครือข่าย ซึ่งสามารถใช้ได้ทั้งระบบเครือข่ายและระบบเครือข่ายไร้สาย โดยคอนโทรล winsock หนึ่งตัวซึ่งสามารถติดต่อกับเครื่องคอมพิวเตอร์อื่นในระบบได้ 1 เครื่อง แต่เมื่อสร้างคอนโทรลอาร์เรย์ winsock แล้วจะสามารถติดต่อกับเครื่องคอมพิวเตอร์ในระบบได้มากกว่า 1 เครื่อง

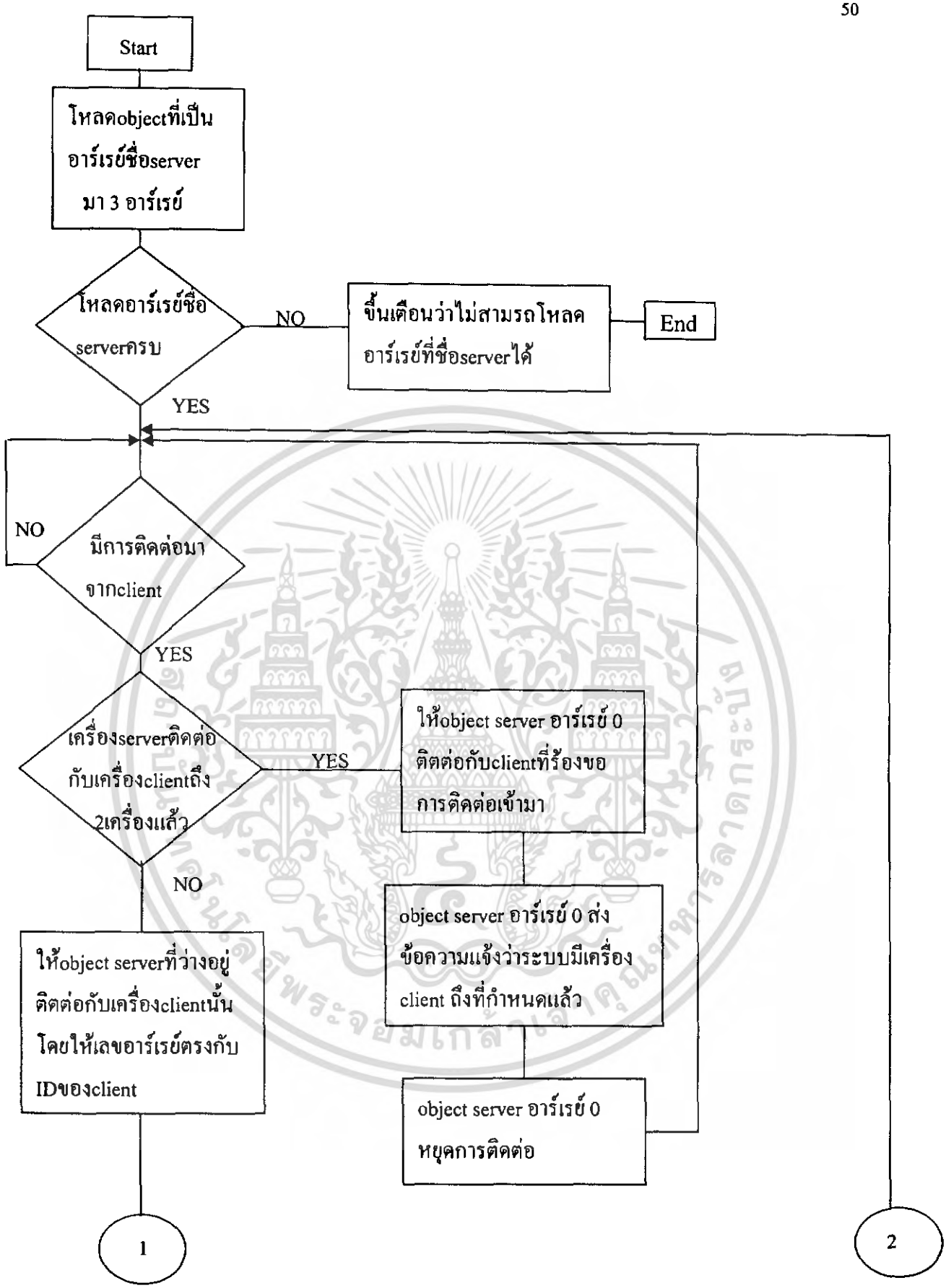
3.5.2 การทำงานของเครื่องแม่ข่าย

เครื่องแม่ข่ายจะทำหน้าที่ติดต่อกับตัวหุ่นโดยตรง โดยจะทำหน้าที่ติดต่อโดยตรงกับไมโครคอนโทรลเลอร์ที่ทำการติดตั้งอยู่ที่ตัวหุ่น 2 ตัว และติดต่อสื่อสารกับเครื่องลูกข่ายที่อยู่ในระบบ พร้อมทั้งจัดลำดับความสำคัญในการทำงานของเครื่องลูกข่าย

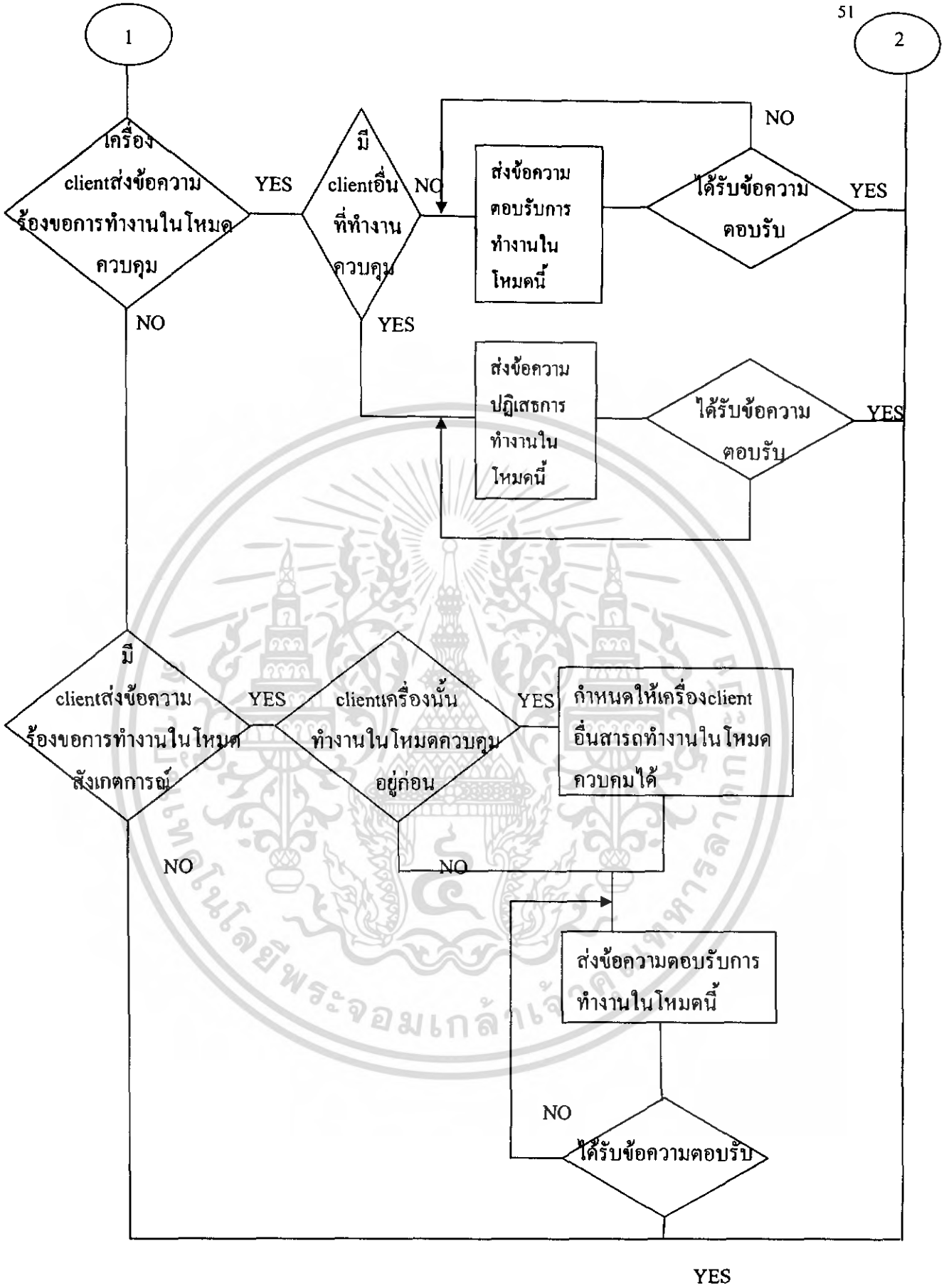
3.5.3 การติดต่อกับเครื่องลูกข่าย

เมื่อเริ่มต้นการทำงานเครื่องแม่ข่ายจะเปิดคอนโทรล winsock ขึ้นมารอรับการติดต่อจากเครื่องลูกข่าย หลังจากที่เครื่องลูกข่ายติดต่อมาแล้วจะแจ้งหมายเลข ID ซึ่งเป็นหมายเลขประจำตัวของเครื่องลูกข่ายแต่ละเครื่องกลับไป โดยที่หมายเลข ID นี้จะเอาไว้ชี้ตัวอาร์เรย์คอนโทรล winsock ที่เอาไว้ติดต่อกับเครื่องลูกข่ายเครื่องนั้น ๆ หลังจากนั้นจะทำการรอรับข้อมูลที่ร้องขอการทำงานจากเครื่องลูกข่ายว่าจะทำงานในโหมดควบคุมหรือโหมดสังเกตการณ์ โดยเครื่องแม่ข่ายจะจัดสรรให้เครื่องลูกข่ายทำงานในโหมดควบคุมเพียงเครื่องเดียว และเมื่อจัดสรรไปแล้วเครื่องลูกข่ายเครื่องอื่นจะเข้ามาทำงานในโหมดนี้ไม่ได้ แต่เครื่องลูกข่ายสามารถทำงานในโหมดสังเกตการณ์ได้มากกว่าหนึ่งเครื่องในเวลาเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 โฟลว์ชาร์ตแสดงการทำงานของ เมื่อเครื่องแม่ข่ายได้รับการร้องขอการติดต่อจากเครื่องลูกข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

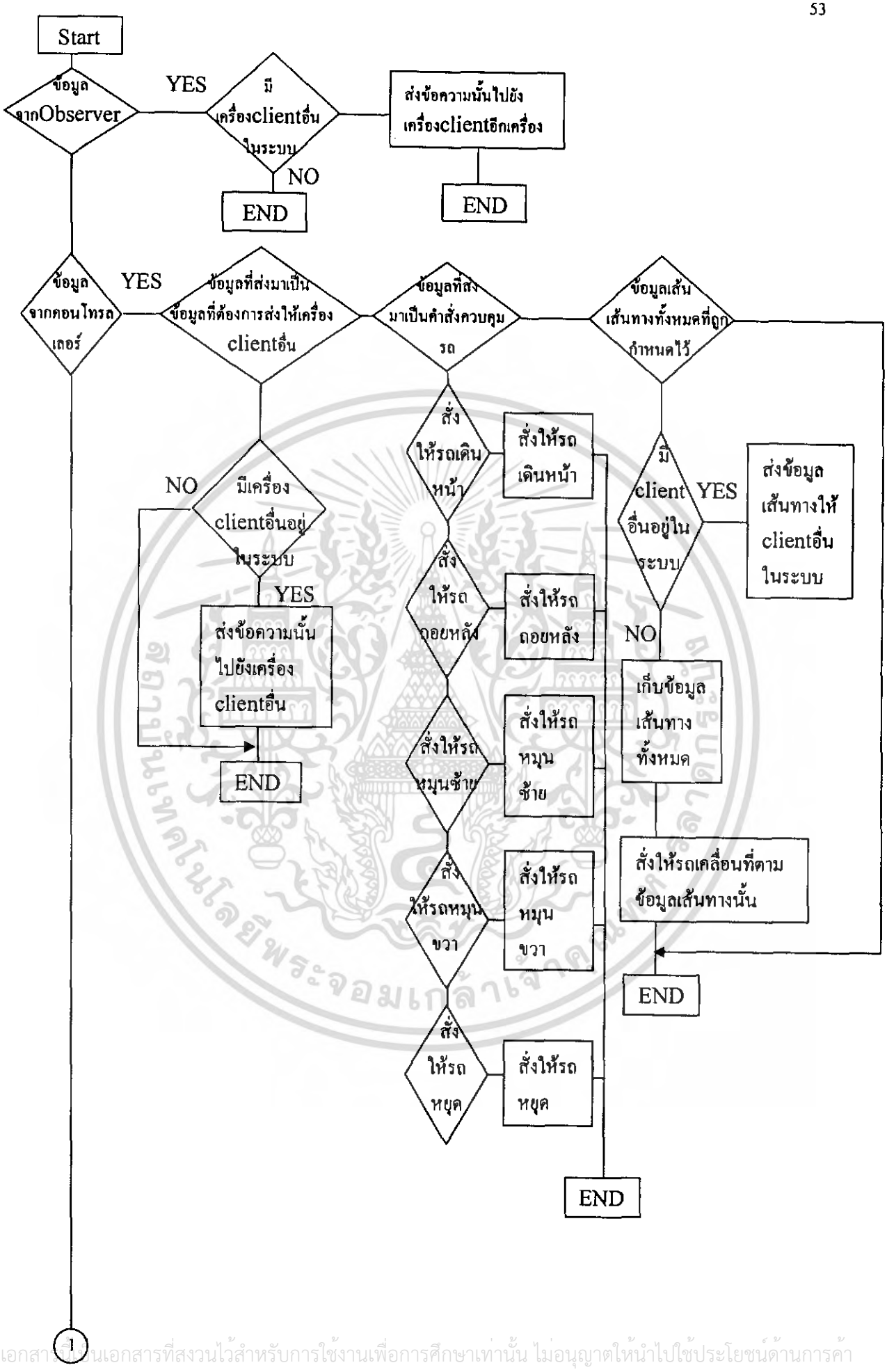
เมื่อเครื่องลูกข่ายใดที่ต้องการทำหน้าที่ควบคุม หรือทำหน้าที่สังเกตการณ์ จะส่งข้อความที่ระบุการร้องขอการทำงานในโหมดนั้นๆ และเครื่องแม่ข่ายจะทำการส่งข้อความตอบรับการทำงานในโหมดนั้นๆ เครื่องลูกข่ายจึงจะสามารถทำงานในโหมดที่ร้องขอไปได้ แต่ในบางครั้งที่ระบบการสื่อสารมีปัญหา ทำให้ร้องขอการทำงานหรือข้อความตอบรับสูญหายทำให้เครื่องลูกข่ายไม่สามารถทำงานได้ จึงพัฒนาโดยการให้เครื่องลูกข่ายส่งข้อความตอบรับการได้รับข้อความหลังจากได้รับสัญญาณการตอบรับการทำงานจากเครื่องแม่ข่ายแล้ว ถ้าเครื่องแม่ข่ายไม่ได้รับสัญญาณตอบรับการได้รับข้อความภายในเวลาที่ตั้งไว้เครื่องแม่ข่ายจะทำการส่งข้อความนั้น ๆ อีกครั้ง

นอกจากนั้นเครื่องแม่ข่ายจะตรวจสอบสถานะการติดต่อของเครื่องลูกข่ายต่างๆอยู่ตลอดเวลา ว่ามีเครื่องใดหลุดไปจากการติดต่อสื่อสารหรือยัง เมื่อตรวจพบว่าเครื่องที่ทำหน้าที่ควบคุมหลุดไปจากการติดต่อสื่อสารไม่ว่าด้วยเหตุใดก็ตามเครื่องแม่ข่ายจะจัดสรรให้เครื่องลูกข่ายอีกเครื่องที่ทำหน้าที่สังเกตการณ์ทำงานในโหมดควบคุมโดยอัตโนมัติ

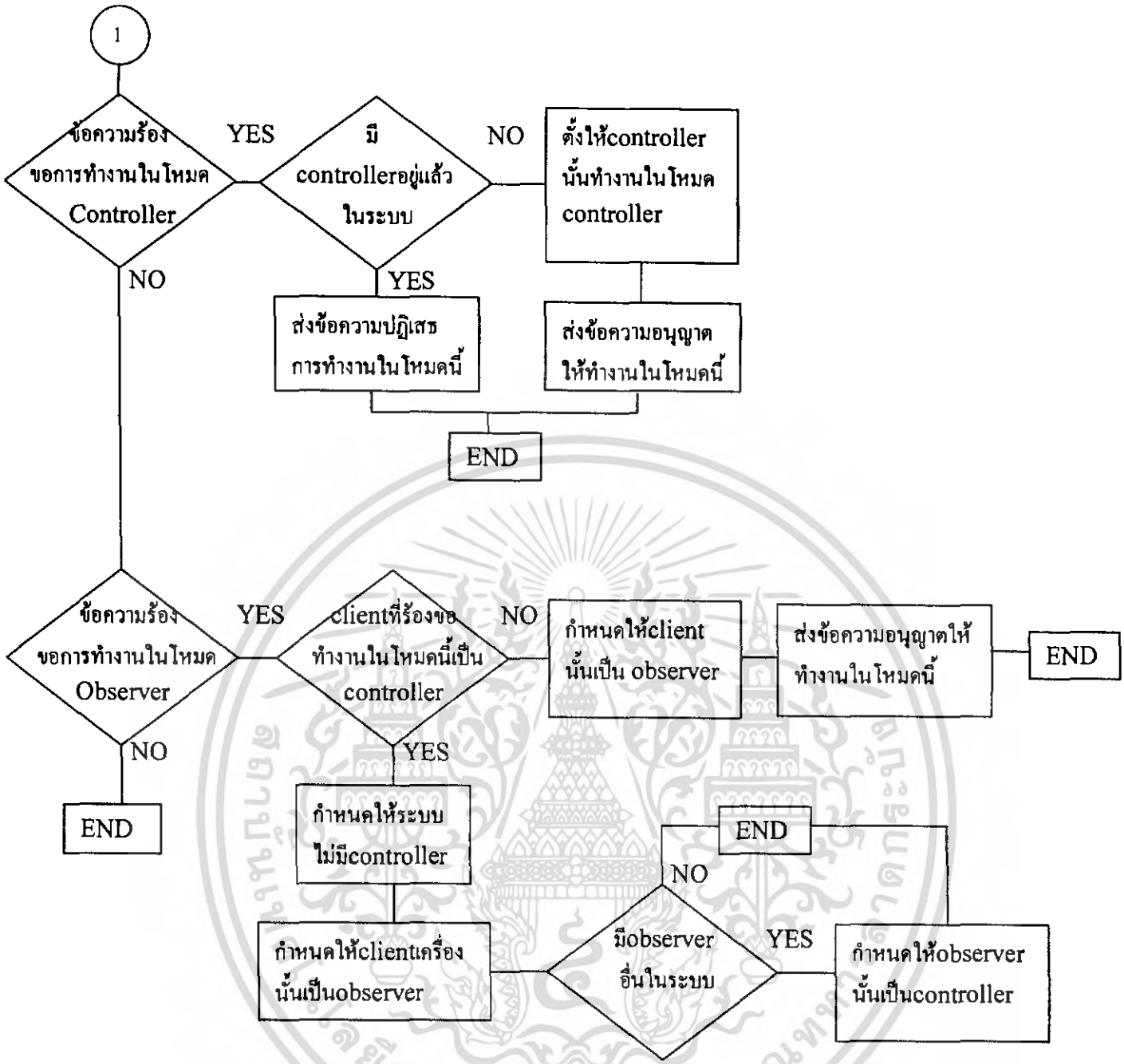
ข้อมูลที่ได้รับจากเครื่องลูกข่ายจะมีอยู่ 2 ประเภท คือ

1. ข้อมูลที่เป็นคำสั่ง โดยเป็นคำสั่งให้รถเคลื่อนที่หรือคำสั่งให้หมุนกล้อง คำสั่งนี้จะได้รับจากเครื่องลูกข่ายที่ทำงานในโหมดควบคุมเท่านั้น
2. ข้อมูลที่เป็นข้อมูลที่ต้องการส่งให้เครื่องลูกข่ายเครื่องอื่น เมื่อได้รับแล้วจะทำการส่งต่อไปยังเครื่องลูกข่ายที่ต้องการส่งถึงพร้อมทั้งแจ้งไปด้วยว่ามาจากเครื่องที่มี ID อะไรและทำงานในโหมดไหน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 โฟลว์ชาร์ตแสดงการทำงานของ เครื่องแม่ข่ายเมื่อได้รับข้อมูลจากเครื่องลูกข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

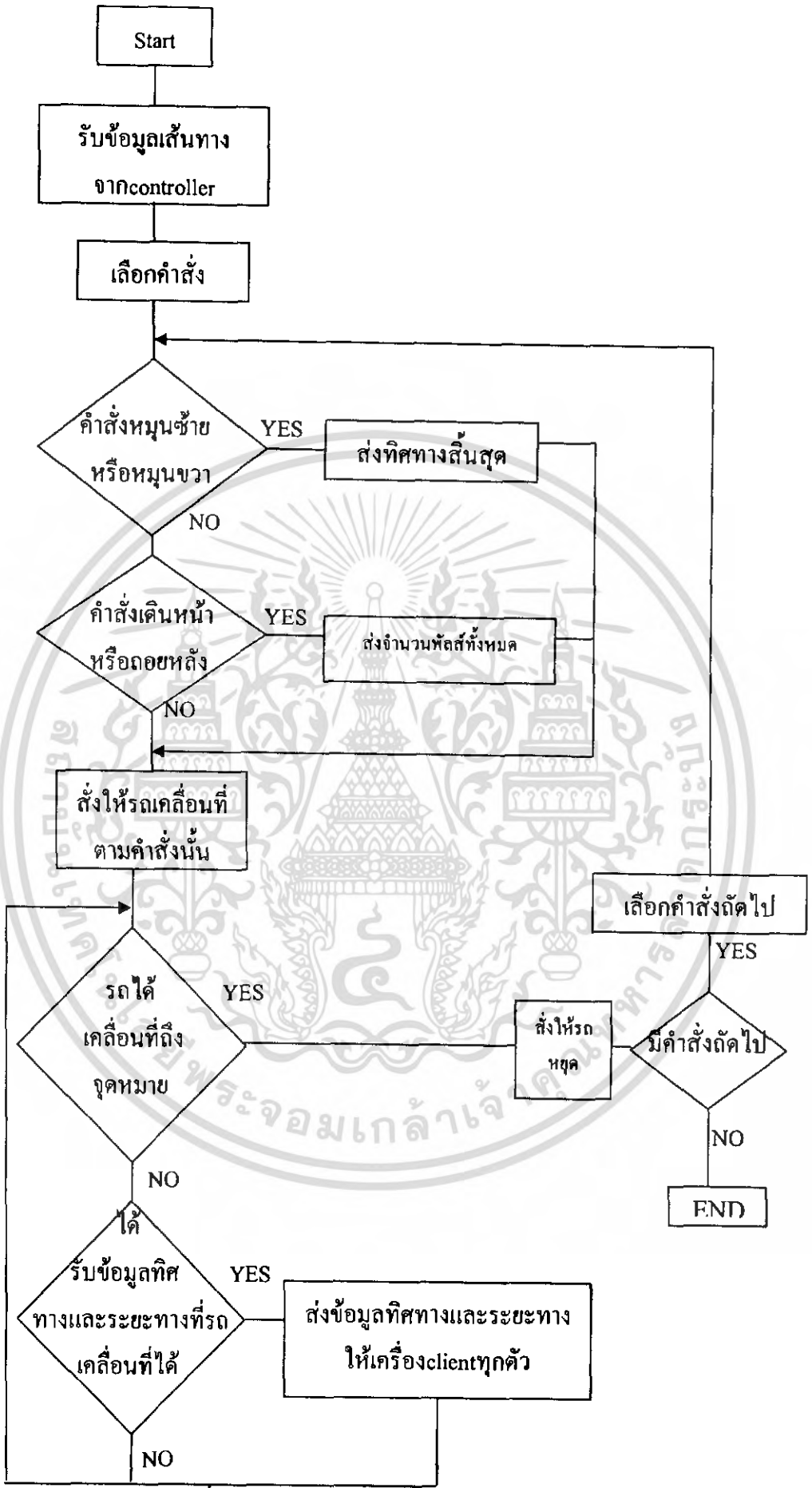
3.5.4 การติดต่อกับไมโครคอนโทรลเลอร์

ในโครงงานนี้ใช้ไมโครคอนโทรลเลอร์ 2 ตัวเพื่อควบคุมการทำงานของตัวหุ่น ซึ่งโดยภาพรวมแล้วเครื่องแม่ข่ายจะทำการสั่งให้ตัวหุ่นเคลื่อนที่ไปตามต้องการ ซึ่งสามารถแบ่งคำสั่งควบคุมออกได้เป็นสองประเภท คือ คำสั่งควบคุมแบบ manual และคำสั่งควบคุมแบบ Automatic ซึ่งผู้ใช้งานจะสามารถสั่งงานได้เมื่อทำงานในโหมดควบคุมเท่านั้น

- คำสั่งควบคุมแบบ manual ผู้สั่งงานต้องสั่งตัวหุ่นโดยตรงให้เคลื่อนที่ไปตามต้องการ ไม่ว่าจะเดินหน้า ถอยหลัง หมุนซ้ายหรือหมุนขวา

- คำสั่งควบคุมแบบ Automatic ผู้สั่งงานจะต้องกำหนดคำสั่งเอาไว้ก่อนหน้าตามลำดับ โดยผู้ใช้งานจะต้องระบุระยะทางและองศาที่ต้องการหมุนและช่วงเวลาที่ต้องการให้รถหยุดอยู่กับที่ให้เรียบร้อยก่อนการส่ง หลังจากเครื่องแม่ข่ายได้รับคำสั่งแล้วจะส่งเส้นทางเดินนั้นให้เครื่องลูกข่ายอื่น ๆ หลังจากนั้นเครื่องแม่ข่ายจะทำการจดจำคำสั่งเหล่านั้นเอาไว้และสั่งให้รถเคลื่อนที่ตามลำดับคำสั่งนั้น ระหว่างที่รถเคลื่อนที่จะรอรับระยะทางที่รถเคลื่อนที่ไปได้และระยะทางจากไมโครคอนโทรลเลอร์เพื่อส่งต่อไปยังเครื่องลูกข่าย และเมื่อรถเคลื่อนที่ได้ตามระยะทางหรือทิศทางที่กำหนดไว้ตอนต้นไมโครคอนโทรลเลอร์จะส่งคำสั่งหยุดมาและเครื่องแม่ข่ายจะสั่งให้ตัวหุ่นหยุดเคลื่อนที่ทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 โฟลว์ชาร์ตแสดงการทำงานของ Automatic Mode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถแบ่งการทำงานของไมโครคอนโทรลเลอร์ทั้งสองได้ดังนี้

- ไมโครคอนโทรลเลอร์ตัวแรกที่จะทำหน้าที่ในการสั่งงานตัวหุ่น ตรวจสอบสถานะของตัว Photo Diode และตรวจวัดอุณหภูมิ เครื่องแม่ข่ายจะรับคำสั่ง เครื่องแม่ข่ายจะรอรับข้อมูลจากไมโครคอนโทรลเลอร์และจะส่งต่อไปยังเครื่องลูกข่ายทุกเครื่องที่อยู่ในระบบ และสั่งงานไปให้ตัวหุ่นเคลื่อนที่

- ไมโครคอนโทรลเลอร์ตัวที่สองจะทำหน้าที่สั่งงานเซอร์โวมอเตอร์เพื่อนำไปหมุนกล้อง ตรวจนับจำนวน pulse ที่จะนับได้จาก encoder และวัดทิศทางจากตัวโมดูลเข็มทิศดิจิตอล เครื่องแม่ข่ายจะรอรับทิศทางและระยะทางที่ตัวหุ่นเคลื่อนที่ไปได้และทำการเก็บค่าเอาไว้ พร้อมทั้งส่งข้อมูลต่อไปให้เครื่องลูกข่ายทุกเครื่องที่อยู่ในระบบ และสั่งงานให้เซอร์โวมอเตอร์หมุน

3.5.5 การทำงานของเครื่องลูกข่าย

เครื่องลูกข่ายจะทำหน้าที่ติดต่อสื่อสารกับผู้ที่ใช้งานโดยตรง โดยจะมีความสามารถทำงานได้ทั้ง 2 โหมดดังที่ได้กล่าวไปแล้ว เครื่องลูกข่ายจะรับข้อมูลจากเครื่องแม่ข่ายและแสดงผลให้ผู้ใช้งานได้รับรู้ถึงสถานการณ์ทำงานต่าง ๆ ของตัวหุ่น และรับคำสั่งต่าง ๆ จากผู้ใช้งานส่งไปยังเครื่องแม่ข่ายเพื่อทำการควบคุมตัวหุ่นต่อไป นอกจากนี้ในการส่งข้อมูลเส้นทางที่ต้องการให้หุ่นเดินทางไปนั้นสามารถกำหนดเส้นทางได้ก่อนโดยโปรแกรมสามารถแสดงเส้นทางที่ได้กำหนดเอาไว้ พร้อมทั้งผู้ใช้สามารถกำหนดสเกลที่ใช้วาดเส้นทางได้ 6 สเกล คือ 1 cm = 0.5 m,1 cm = 1 m, 1cm = 2 m , 1 cm = 5 m , 1 mm = 1m 1 mm = 2 m โดยผู้ใช้งานสามารถปรับเปลี่ยนหรือแก้ไขได้ก่อนสั่งให้หุ่นเคลื่อนที่ นอกจากนี้หลังจากที่มีการเคลื่อนที่ไปแล้ว โปรแกรมจะรับค่าทิศทางและระยะทางที่หุ่นเคลื่อนที่ไปแล้วจากเครื่องแม่ข่ายและแสดงผลออกมาเป็นรูปภาพเปรียบเทียบกับเส้นทางที่กำหนดเอาไว้ก่อนหน้าได้ โดยจะแสดงออกมาตามสเกลที่กำหนด

3.5.6 โปรแกรมการทำงาน

เมื่อเข้าสู่โปรแกรมการทำงานของเครื่องลูกข่าย จอมอนิเตอร์จะแสดงหน้าจอของโปรแกรมควบคุมดังรูป 3.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.9 แสดงหน้าจอ Disconnected

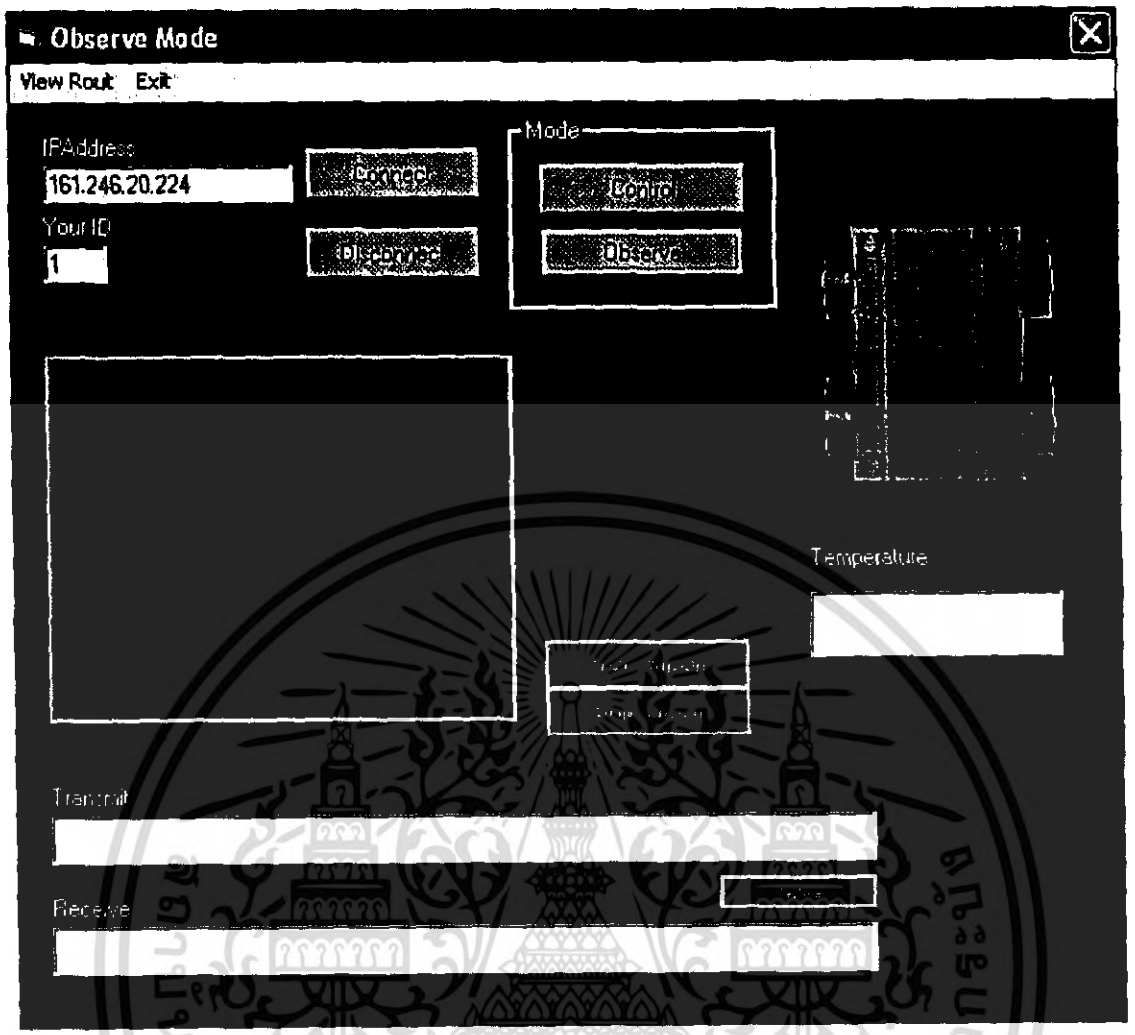
หลังจากเข้าสู่โปรแกรมการทำงานแล้วให้กำหนดหมายเลข IP ของเครื่องแม่ข่ายที่แถบข้อความด้านบนซ้ายแล้วกดปุ่มจาก Connect เพื่อติดต่อกับเครื่องแม่ข่าย จอมอนิเตอร์จะแสดงหน้าจอดังรูป 3.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.10 แสดงหน้าจอ Connected

หลังจากติดต่อกับเครื่องแม่ข่ายแล้วหมายเลข ID จะปรากฏขึ้นที่ช่องแสดงเลข ID หลังจากนั้นกดปุ่มเลือกโหมดการทำงาน โดยสามารถเลือกเข้าสู่การทำงานในโหมดควบคุมหรือสังเกตการณ์ได้โดยกดปุ่ม Control หรือ Observe เมื่อกดปุ่ม Observe จะเข้าสู่หน้าจอการทำงานในโหมดสังเกตการณ์ หรือกดปุ่ม Control เพื่อเข้าสู่โหมดควบคุม

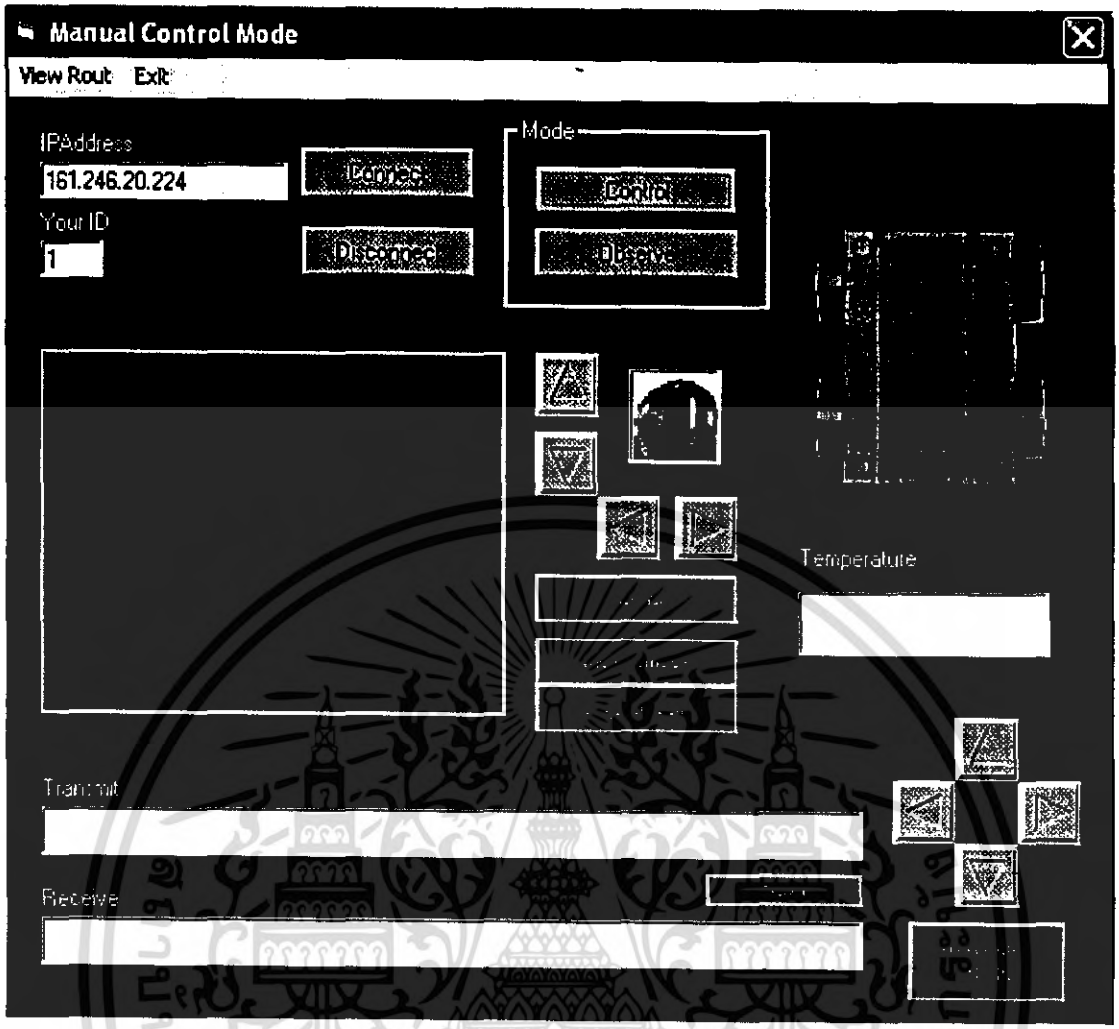
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 แสดงหน้าจอ Observed Mode

ในโหมดนี้จะปรากฏหน้าจอที่เอาไว้แสดงภาพที่ถ่ายมาได้จากกล้อง webcam ,อุณหภูมิที่ตัวหุ่นวัดได้ที่แถบข้อความด้านขวาของหน้าจอการทำงาน และในส่วนล่างสุดของโปรแกรมการทำงานจะแสดงแถบข้อความที่แสดงข้อมูลที่ส่งมาจากเครื่องลูกข่ายเครื่องอื่น หลังจากเข้าสู่โหมดนี้แล้วสามารถเปลี่ยนโหมดการทำงานเข้าสู่โหมดควบคุมโดยกดปุ่ม control หรือสามารถเข้าสู่โหมดนี้ได้ทันทีหลังจากติดต่อกับเครื่องแม่ข่ายแล้วโดยกดปุ่มเดียวกัน โปรแกรมการทำงานจะแสดงหน้าจอดังรูป 3.12

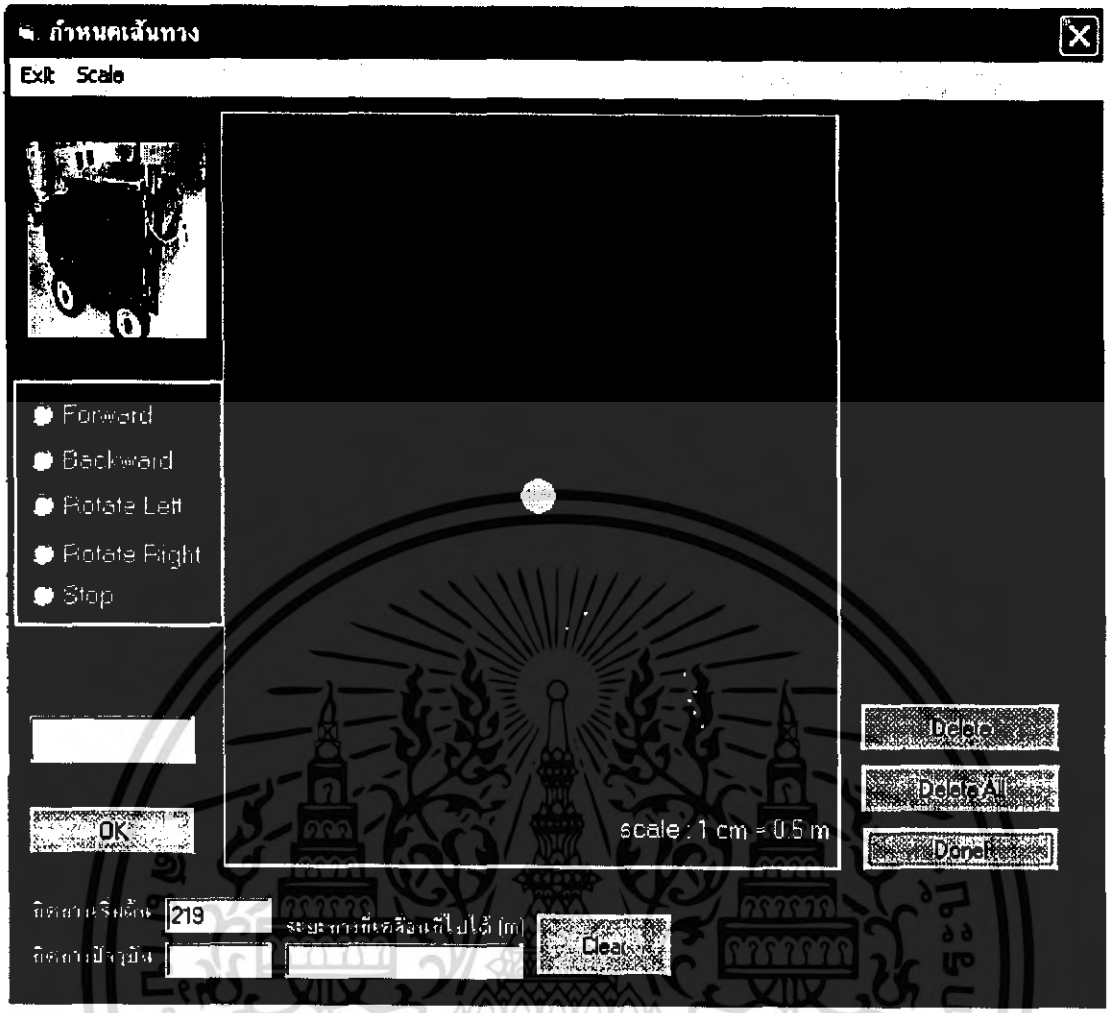
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 แสดงหน้าจอControl Mode

หน้าจอในโหมดนี้จะแตกต่างกับโหมดสังเกตการณ์ตรงที่สามารถบังคับหุ่นได้โดยกดที่ปุ่มลูกศรด้านล่างขวาของหน้าจอ และถ้ากดปุ่ม Automatic Mode จะเข้าสู่หน้าจอการทำงานดังรูป 3.13

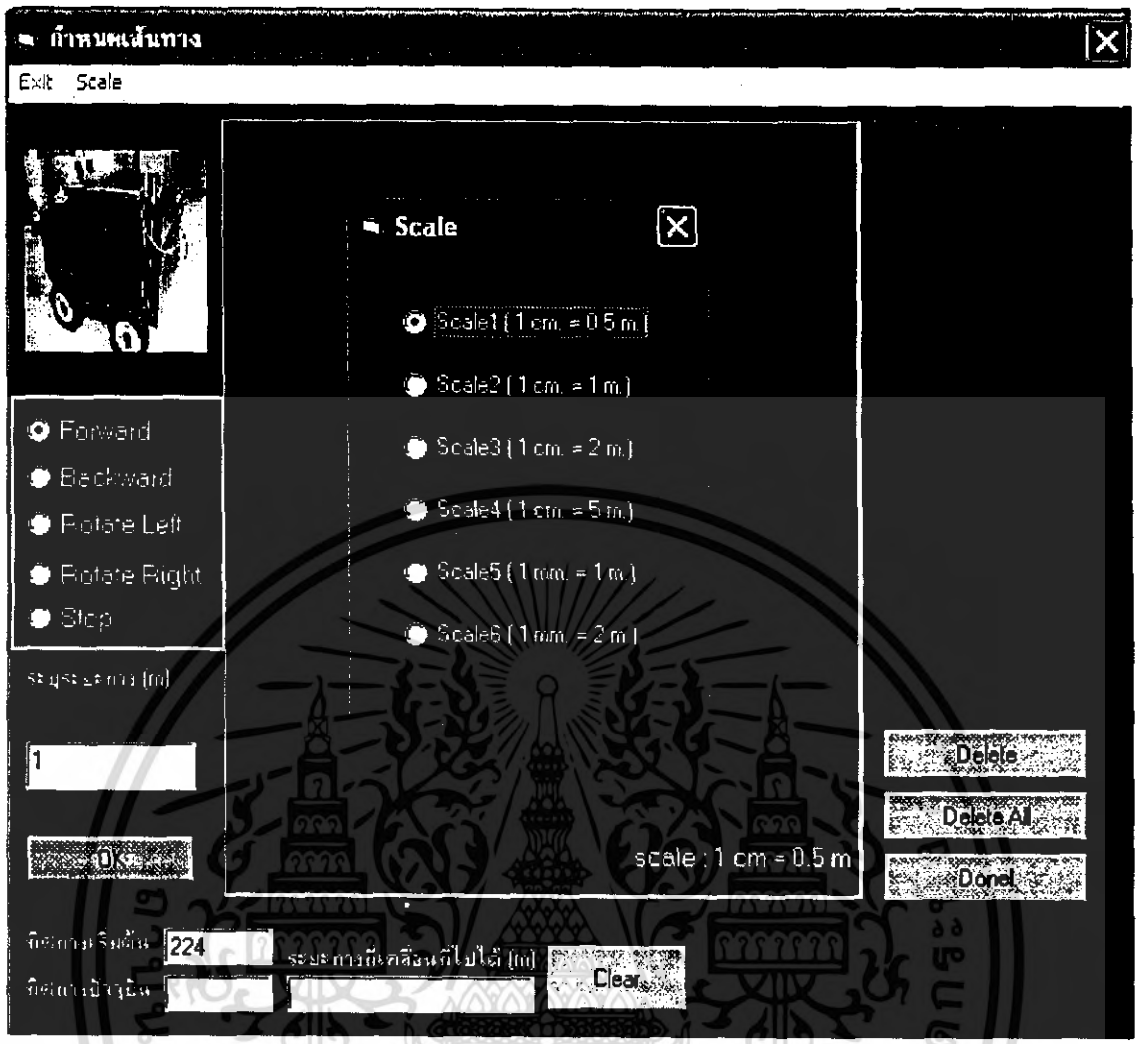
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 แสดงหน้าจอ Automatic Mode

ในหน้าจอนี้ผู้ใช้งานสามารถกำหนดเส้นทางของตัวหุ่นได้ล่วงหน้าโดยเลือกจากปุ่ม option ด้านซ้ายว่าต้องการให้ตัวหุ่นเคลื่อนที่อย่างไรและมีรายละเอียดการเคลื่อนที่อย่างไร เช่น ต้องการให้รถเดินหน้า 5 m ให้เลือก Forward แล้วใส่ตัวเลข 5 ลงในแถบข้อความด้านล่างแล้วจึงกดปุ่ม OK หน้าจอจะแสดงรูปเส้นทางที่ผู้ใช้ได้กำหนดไว้ และผู้ใช้ยังสามารถลบเส้นทางใดๆ โดยกดปุ่ม Delete หรือลบทั้งหมดโดยกดปุ่ม Delete All นอกจากนี้ผู้ใช้ยังสามารถเปลี่ยนสเกลที่แสดงเส้นทางของตัวหุ่นได้โดยกดที่ Scale ที่แถบเมนูด้านบนซ้าย โปรแกรมจะแสดงหน้าจอให้ผู้ใช้สามารถเลือกสเกลที่ใช้วาดได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.14 แสดงหน้าจอ Scale

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

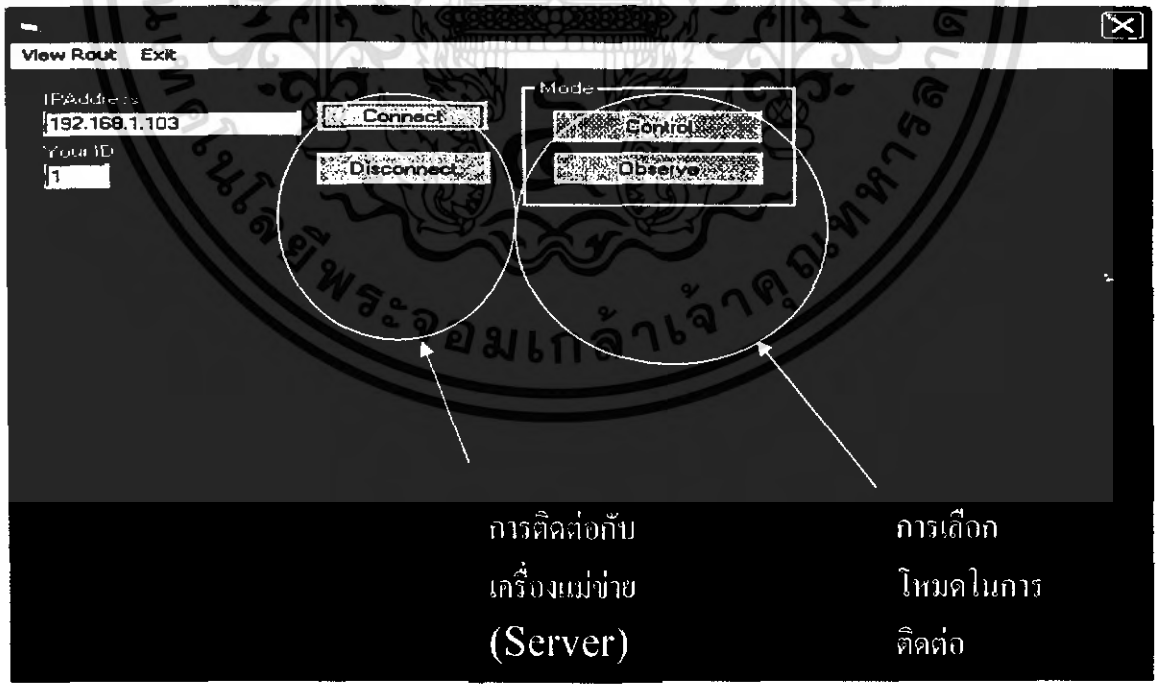
ผลการทดลอง

4.1 การติดต่อสื่อสารระหว่างคอมพิวเตอร์กับหุ่นยนต์สำรวจ

การติดต่อสื่อสารเพื่อรับ-ส่งข้อมูลต่างๆนั้นใช้การรับส่งแบบไร้สาย โดยใช้การติดต่อสื่อสารระบบไวร์เลสแลน มาตรฐานเครือข่ายไวร์เลสแลนที่ใช้งาน คือ มาตรฐาน IEEE 802.11 gในย่านความถี่2.4 กิกะเฮิรตซ์ อัตราความเร็วสูงสุดอยู่ที่ 54 เมกะบิตต่อวินาที ระยะการทางวัดได้สูงสุดอยู่ที่ประมาณ 100 เมตร ในส่วนของภาพที่จะปรากฏบนหน้าจอมอนิเตอร์นั้น เป็นภาพที่ส่งมาจากเครื่องส่งสัญญาณภาพบนตัวหุ่นยนต์เคลื่อนที่ (Mobile Robot) ซึ่งก็คือกล้องเว็บแคม (Webcam) ช่วยในการมองเห็นวัตถุ และทิศทางสำหรับใช้ในการสำรวจ

4.2 การทำงานของโครงงาน

หลังจากที่เปิดเครื่องรับส่งสัญญาณที่คอมพิวเตอร์และตัวหุ่นยนต์เรียบร้อยแล้ว เมื่อเข้าสู่โปรแกรม Mobile Robot ซึ่งเป็นโปรแกรมควบคุมการทำงานของหุ่นยนต์เคลื่อนที่ (Mobile Robot) จะปรากฏหน้าจอของโปรแกรมดังรูปที่ 4.1 เพื่อทำการติดต่อกับเครื่องแม่ข่ายเมื่อติดต่อ (Connect) แล้ว จะต้องเลือกว่าจะทำงานในระบบควบคุม (Control) หรือในระบบสังเกตการณ์ (Observe)



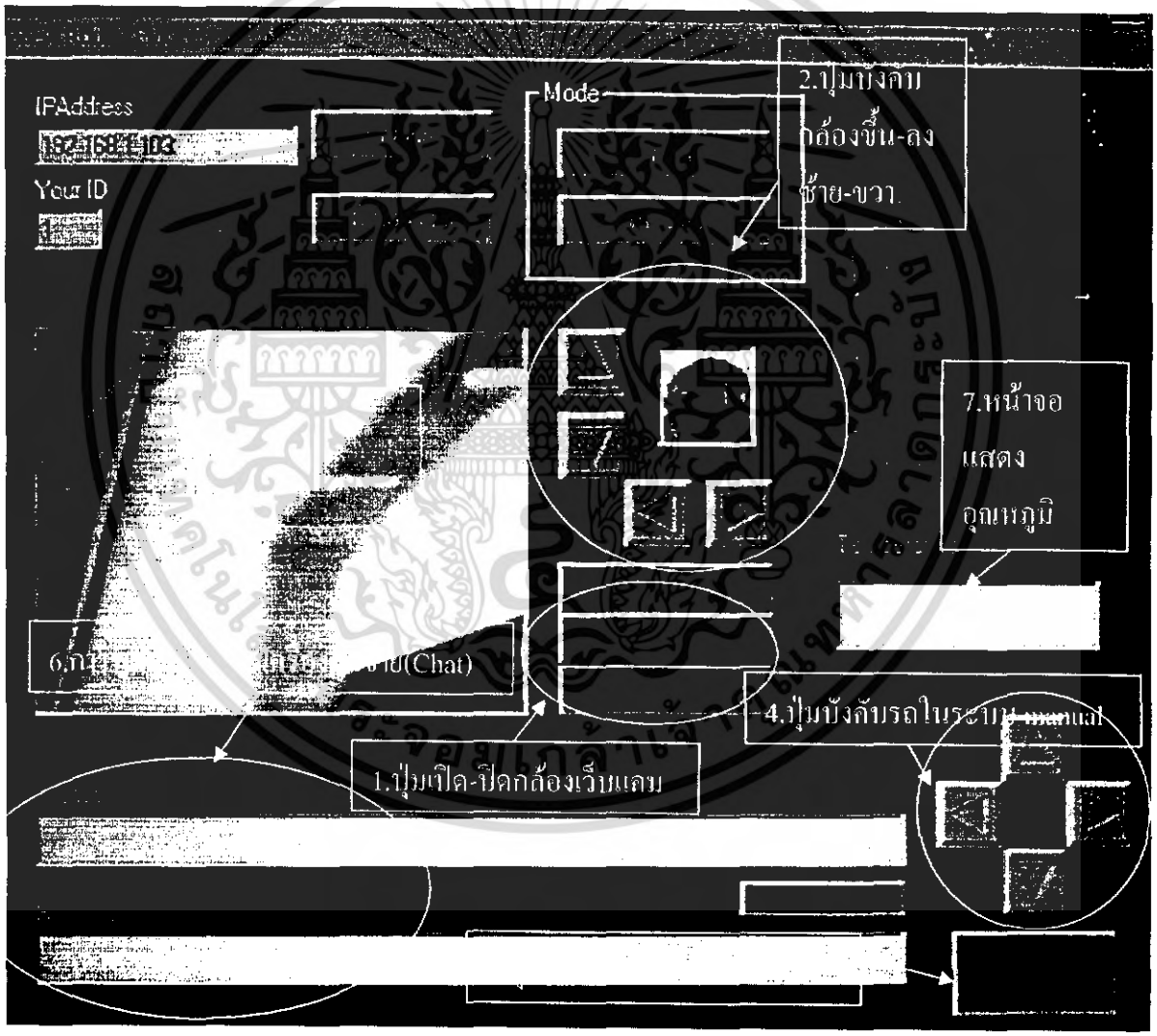
รูปที่ 4.1 แสดงหน้าจอการติดต่อและเลือกโหมดการทำงานของเครื่องลูกข่าย (Client)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.1 การทำงานโหมด Control

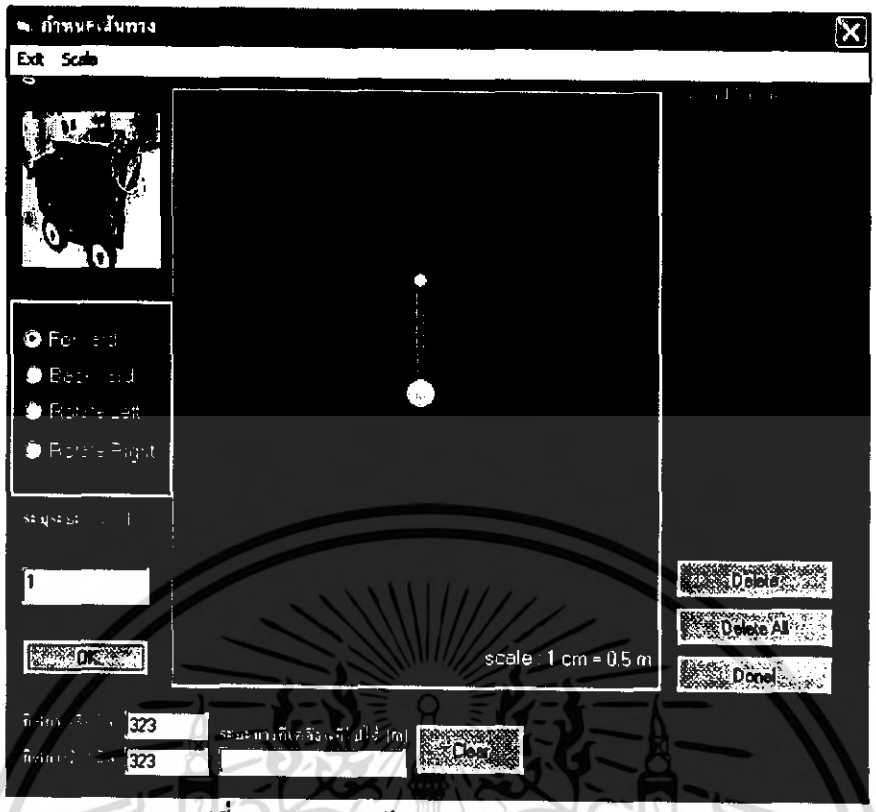
หลังจากเข้าสู่หน้าจอของการควบคุมในโหมด Control เรียบร้อยแล้ว แสดงได้ดังรูปที่ 4.2 ซึ่งจะมีการควบคุมต่างๆ ดังนี้

1. การเปิด-ปิดกล้องเว็บแคม
2. การบังคับกล้องให้เคลื่อนที่ ซ้าย-ขวา ขึ้น-ลง
3. หน้าจอแสดงภาพจากกล้องเว็บแคม
4. การบังคับรถ ให้เคลื่อนที่ในระบบmanual
5. การบังคับรถในระบบ Automatic
6. การพูดคุยระหว่างเครื่องลูกข่าย (Chat)
7. หน้าจอแสดงอุณหภูมิ



รูปที่ 4.2 แสดงหน้าจอในโหมด Control

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



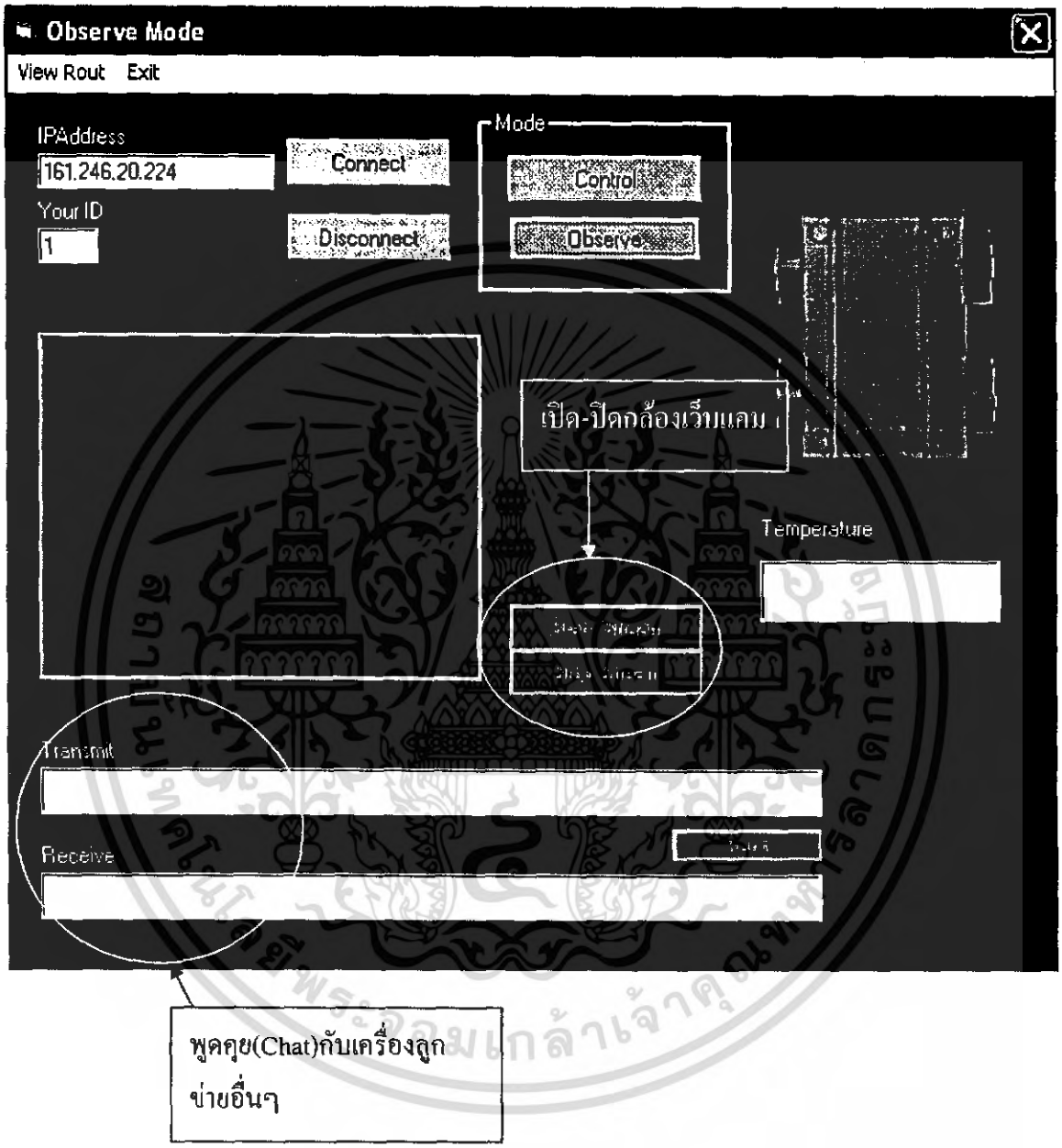
รูปที่ 4.3 แสดงหน้าจอการทำงานในระบบ Automatic

รูปที่ 4.4 แสดงหน้าจอในโหมด Automatic เมื่อสั่งให้ทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 การทำงานโหมด Observe

การทำงานในระบบสังเกตการณ์ (Observe) จะไม่สามารถควบคุมการเคลื่อนที่ของหุ่นยนต์หรือการบังคับกล้องได้ ดังรูปที่ 4.5 ในระบบสังเกตการณ์นี้จะสามารถเปิด-ปิดกล้องและทำการพูดคุยกับเครื่องลูกข่ายเครื่องอื่นๆได้



รูปที่ 4.5 แสดงหน้าจอในโหมดสังเกตการณ์ (Observe)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การทดลอง

4.3.1 การทดสอบความสามารถในการเคลื่อนที่

ในการทดลองนี้จะทำการวัดระยะการเคลื่อนที่ของหุ่นยนต์ในระบบ Automatic โดยทำการวัดการเคลื่อนที่ เดินหน้า ถอยหลัง เลี้ยวซ้าย เลี้ยวขวา อย่างละ 5 ครั้ง ดังตารางที่ 4.1

ตารางที่ 4.1 แสดงการเคลื่อนที่ในโหมดAutomatic

โหมด	กำหนด (เมตร)	ระยะที่เคลื่อนที่ได้ (เมตร)	ค่าคลาดเคลื่อน (เมตร)
เดินหน้า	1	0.97	0.03
	1	0.98	0.02
	1	0.96	0.04
	1	0.975	0.025
	1	0.93	0.07
ถอยหลัง	1	0.92	0.08
	1	0.95	0.05
	1	0.955	0.045
	1	0.975	0.025
	1	0.93	0.07
โหมด	กำหนด (องศา)	มุมที่เคลื่อนที่ได้ (องศา)	ค่าคลาดเคลื่อน (องศา)
เลี้ยวซ้าย	90	92	2
	90	92	2
	90	92	2
	90	93	3
	90	93	3
เลี้ยวขวา	90	89	1
	90	83	7
	90	86	4
	90	87	3
	90	88	2

จากการทดสอบความสามารถในการเคลื่อนที่ หุ่นยนต์สามารถเคลื่อนที่ในแนวเส้นตรงคือเดินหน้าและถอยหลังได้ค่าที่ใกล้เคียงกับค่าที่กำหนดโดยมีค่าความคลาดเคลื่อนในทศนิยมตำแหน่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่สองหรือในหน่วยเซนติเมตรนั่นเอง ส่วนการเลี้ยวมีค่าคลาดเคลื่อนเฉลี่ยเลี้ยวซ้ายอยู่ที่ 2.4 องศา และเลี้ยวขวาค่าคลาดเคลื่อนอยู่ที่ 3.4 องศา

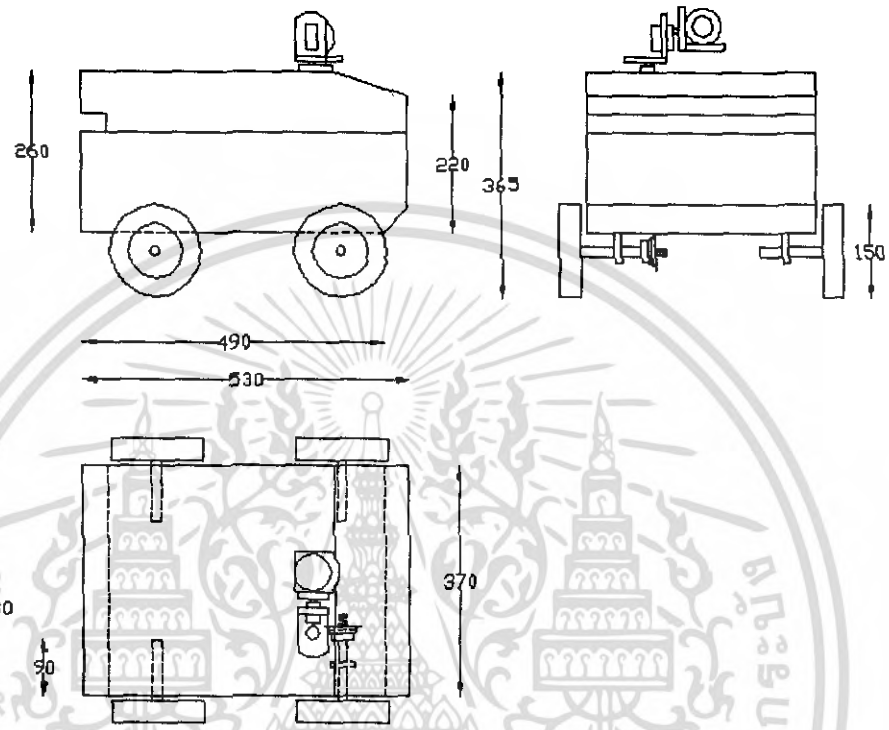
4.4 ข้อมูลจำเพาะของหุ่นยนต์เคลื่อนที่

ส่วนประกอบต่างๆที่ประกอบอยู่บนหุ่นยนต์เคลื่อนที่

1. บอร์ดไมโครคอนโทรลเลอร์(MCS-51)	1 ชุด
2. ชุดควบคุมและขับมอเตอร์	1 ชุด
3. กล้องเว็บแคม	1 ตัว
4. มอเตอร์ดีซี สำหรับการขับเคลื่อน	2 ตัว
5. ชุดควบคุมเซอร์โวมอเตอร์ สำหรับบังคับกล้องเว็บแคม	1 ชุด
6. เซอร์โวมอเตอร์ สำหรับบังคับกล้องเว็บแคม	2 ตัว
7. บอร์ดโมดูลเข็มทิศสำหรับควบคุมการเลี้ยว	1 ชุด
8. ชุดควบคุมระยะทางการเคลื่อนที่	1 ชุด
9. Encoderและเซนเซอร์	1 ตัว
10. แบตเตอร์รี่แห้งชนิดประจุใหม่ ขนาดแรงดัน 12 V 5 AH	1 ลูก
11. แบตเตอร์รี่แห้งชนิดประจุใหม่ ขนาดแรงดัน 6 V 5 AH	2 ลูก
12. คอมพิวเตอร์โน้ตบุ๊ค สำหรับเป็นเครื่องแม่ข่าย (Server)	1 เครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความกว้างของตัวรถ	37 cm
ความยาวของตัวรถ	53 cm
ความสูงของตัวรถ	36.5 cm



รูปที่ 4.6 แสดงรายละเอียดโครงสร้างของหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผล ปัญหาและข้อเสนอแนะ

5.1 สรุปผล

จากการทดสอบการควบคุมให้หุ่นยนต์เคลื่อนที่ทำตามคำสั่งต่างๆ ได้แก่ การควบคุมการเคลื่อนที่ทั้งแบบกำหนดชุดคำสั่งล่วงหน้า โดยจะประกอบไปด้วยการเคลื่อนที่ไปข้างหน้า ถอยหลัง เลี้ยวซ้าย และเลี้ยวขวา ผลปรากฏว่าหุ่นยนต์สามารถเคลื่อนที่ได้ตามชุดคำสั่งที่กำหนดไว้ล่วงหน้าได้ตามลำดับ แต่ว่ามีความคลาดเคลื่อนในระยะการเคลื่อนที่และองศาที่หมุนไป สำหรับการควบคุมเซอร์โวมอเตอร์เพื่อหมุนกล้อง webcam ซึ่งสามารถทำได้อย่างถูกต้อง แต่การแสดงภาพที่ถ่ายได้จากกล้อง webcam พบว่ายังมีปัญหาในเรื่องการส่งข้อมูล ทำให้ภาพที่ได้มีความล่าช้ากว่าปกติ

5.2 ปัญหา

1. ระยะทางการเคลื่อนที่ที่ไม่เป็นไปตามที่ตั้งเอาไว้ สาเหตุอาจเนื่องมาจาก
   - ความไวในการรับข้อมูลผ่านพอร์ตอนุกรม RS-232
   - โซ่ที่ใช้หมุนเฟืองที่ติดกับล้อหย่อนทำให้เมื่อสั่งรถหยุด รถจะเคลื่อนที่ต่อไปได้อีกระยะหนึ่ง
   - เนื่องจากการหมุนล้อด้านตรงกันข้ามกับทิศที่จะหมุนควรจะเคลื่อนที่เร็วกว่าอีกด้านจึงจะทำให้จุดศูนย์กลางการหมุนอยู่ที่กลางหุ่นยนต์ แต่เนื่องจากหุ่นยนต์เคลื่อนที่ที่นำมาพัฒนาไม่ได้ถูกออกแบบมาให้มีจุดศูนย์กลางอยู่กับที่เวลาหมุน จึงทำให้เกิดการไถลเวลาหมุน
   - ล้อแต่ละล้อรับน้ำหนักไม่เท่ากัน ทำให้เวลาหมุนเกิดการเลื่อนไถล
2. องศาที่หมุนไปไม่เป็นไปตามที่ตั้งเอาไว้ สาเหตุอาจเนื่องมาจาก
   - เนื่องจากสัญญาณแม่เหล็กรบกวนไม่ว่าจากมอเตอร์ หรือแหล่งกำเนิดสนามแม่เหล็กอื่นๆ มีผลต่อการอ่านทิศทางของเข็มทิศดิจิตอล ทำให้ไม่สามารถติดตั้งเข็มทิศไว้บริเวณตรงกึ่งกลางรถได้เนื่องจากอยู่ใกล้มอเตอร์
3. การแสดงภาพที่ถ่ายได้จากกล้อง webcam ยังมีความล่าช้ากว่าที่ควรจะเป็น เนื่องจากความเร็วที่ใช้รับส่งข้อมูลของระบบเครือข่ายไร้สายบางครั้งไม่เพียงพอที่จะส่งภาพไปแสดงผลที่อีกเครื่องได้ทัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 ข้อเสนอแนะ

1. ปรับโครงสร้างชุดขับเคลื่อนให้โซ่มีความตึงพอดี
2. ปรับการติดตั้งล้อเพื่อให้แต่ละล้อรับน้ำหนักเท่า ๆ กัน
3. เปลี่ยนชุดขับเคลื่อนเพื่อความคุมความเร็วเวลาหมุนได้
4. จัดระเบียบการรับส่งข้อมูลให้ดีขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. นคร ภักดีชาติ , ธีรบูลย์ หล่อวิเชียรรุ่ง , ชัยวัฒน์ ลิ้มพรจิตรวิไล , ปฏิบัติการไมโครคอนโทรลเลอร์ MCS-51 ด้วยโปรแกรมภาษาซี ,Innovative Experiment Press
2. CMPS03 Digital Compass Module โมดูลเข็มทิศดิจิตอล,,inex INNOVATIVE EXPERIMENT
3. คู่มือการใช้งาน SERVO MOTOR,ETT CO., LTD.
4. www.inex.co.th/
5. www.ett.co.th/

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

โปรแกรมควบคุมการทำงานในส่วนของไมโครคอนโทรลเลอร์

```
#include<reg51rx.h>
#include<i2c.h>
#include<stdio.h>
#define compass_id 0xc0
unsigned char a = 0;
unsigned int setpoint[3];
unsigned int rotate ;
unsigned int digi2;
unsigned int digi1 ;
unsigned int point;
unsigned int bid  = 0;
unsigned int DAT;
unsigned char index = 0;
unsigned char c;
unsigned char check = 0;
unsigned int checkpoint;
 int display;
 unsigned  int cnt  ;
 int x1;
sbit inA = P2^5;
sbit inB = P2^4;
sbit inC = P2^3;
sbit inD = P2^2;
sbit en2 = P2^6;
sbit en1 = P2^7;
 int sum;
code unsigned int pulse1[25] =
{251,250,249,248,247,246,245,244,243,242,241,240,239,238,237,236,235,234,233,2
32,231,230,229,228,227};
code unsigned int pulse2[25]
={251,250,249,248,247,246,245,244,243,242,241,240,239,238,237,236,235,234,233,
232,231,230,229,228,227};

//////////////////////////////////////////////////////////////////

void superhold ( float hold )
{
 int i,j;
  for(i=0;i<hold;i++)
    {
     checkpoint = read_angle();
     if(checkpoint == point)
      {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 en1 = 0;
 en2 = 0;
 inA = 0;
 inB = 0;
 inC = 0;
 inD = 0;
 check = 1;
     }
     }
 for(j=0;j<2000;j  ++);
}

//////////////////////////////////////////////////////////////

void delay(unsigned char tick)
{unsigned char i,j;
for(i=0;i<tick;i++)
for(j=0;j<200;j++);
}

//////////////////////////////////////////////////////////////

int read_angle()
{
unsigned int datah = 0;
unsigned int datal = 0;
I2c_Start();
I2c_Write(compass_id);
I2c_Write(0x02);

I2c_Start();
I2c_Write(compass_id+1);
datah =i2c_Read();
I2c_Ack();
datal=I2c_Read();
I2c_Nack();
I2c_Stop();
datah = (datah<<8) + (datal);
return(datah/10);
}
//////////////////////////////////////////////////////////////
unsigned int counter (void)
{

sum = (cnt - x1);
  x1 = cnt ;
  return(sum);


 }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
////////////////////////////////////////////////////////////

void send_data()
{
  display =read_angle();
  if(display > 255)
  {
   SBUF = 'O';
   TI = 0;
     checkpoint = read_angle();
   if(checkpoint == point)
    {
  en1 = 0;
  en2 = 0;
  inA = 0;
  inB = 0;
  inC = 0;
  inD = 0;
  check = 1;
      }
    superhold(8);
      checkpoint = read_angle();
   if(checkpoint == point)
    {
  en1 = 0;
  en2 = 0;
  inA = 0;
  inB = 0;
  inC = 0;
  inD = 0;
  check = 1;
      }
    SBUF = display;
    TI = 0;
      checkpoint = read_angle();
    if(checkpoint == point)
     {
   en1 = 0;
   en2 = 0;
   inA = 0;
   inB = 0;
   inC = 0;
   inD = 0;
   check = 1;
      }
     superhold(8);
   }

   else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  {
   SBUF = 'N';
   TI = 0;
     checkpoint = read_angle();
   if(checkpoint == point)
    {
en1 = 0;
en2 = 0;
inA = 0;
inB = 0;
inC = 0;
inD = 0;
check = 1;
    }
   superhold(8);
     checkpoint = read_angle();
   if(checkpoint == point)
    {
en1 = 0;
en2 = 0;
inA = 0;
inB = 0;
inC = 0;
inD = 0;
check = 1;
    }
   SBUF = display;
   TI = 0;
     checkpoint = read_angle();
   if(checkpoint == point)
    {
en1 = 0;
en2 = 0;
inA = 0;
inB = 0;
inC = 0;
inD = 0;
check = 1;
    }
   superhold(8);

  }


}

//////////////////////////////////////////////////////////////////////

void CommandCam(void)  interrupt 4
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
        if(RI)
        {

                RI = 0;
                DAT = SBUF;
          if(a > 0)
                 { setpoint[a-1] = DAT - 48;
                  a++;
                    if(a > 3)
         {
                      a = 0;
                      point = ((setpoint[0]*100) + (setpoint[1]*10) + (setpoint[2]*1));
                      }
                  }

                switch(DAT)
                {
                case 'W' :  a++;
                        break;

                case 'C':
                        digi1 = 11;
                        digi2 = 13;
                        break;
                case 'U':
                        digi1 ++;
                        if (digi1 >= 24)
                                digi1 = 24;
                        break;
                case 'D':
                        digi1 --;
                        if (digi1 <= 1)
                                digi1 = 1;
                        break;
                case 'l':
                        digi2 ++;
                        if (digi2 >= 24)
                                digi2 = 24;
                        break;
                case 'r':
                        digi2 --;
                        if (digi2 <= 1)
                                digi2 = 1;
                        break;
                case 'A':
                        cnt = 0;
                        x1 = 0;
                        break ;
                case 'F':{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    inA = 1; // p2.5
        inB = 0; // p2.4
        inC = 0; // p2.3
        inD = 1; // p2.2
        en1 = 1; // p2.6
        en2 = 1; // p2.7
        }
        break;
    case 'B': {
                    inA = 0;
        inB = 1;
        inC = 1;
        inD = 0;
        en1 = 1;
        en2 = 1;
        }
        break;
            case 'L': {
                    inA = 0;
        inB = 1;
        inC = 0;
        inD = 1;
        en1 = 1;
        en2 = 1;
        }
        break;
            case'R' : {
                    inA = 1;
        inB = 0;
        inC = 1;
        inD = 0;
        en1 = 1;
        en2 = 1;
        }
        break;
            case'T' :{
                    inA = 0;
        inB = 0;
        inC = 0;
        inD = 0;
        en1 = 0;
        en2 = 0;
        }
        break;
            }
CCAP2L = pulse1[digi1];
    CCAP2H = pulse1[digi1];
    CCAP3L = pulse2[digi2];
    CCAP3H = pulse2[digi2];
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if(TI)
        {
                TI=0;
        }
}

//////////////////////////////////////////////////////////////////

 void PCA_interrupt(void) interrupt 6
{
 if(CCF0 &(~bid))
 {
    CCF1 = 0;
    bid = 1;
    cnt++;
    CCF0 = 0;

  }

  if(CCF1 & bid )
  {
    CCF0 = 0;
    bid = 0;
    CCF1 = 0;
  }
}


//////////////////////////////////////////////////////////////////

void main (void)
{
 digi1 =11;
 digi2 =13;
 TMOD = 0x22;
 CMOD = 0x04;
 SCON = 0x50;
 TH1 = 0xFA;
 TL1 = 0xFA;
 TH0 = 0x70;
 TL0 = 0x70;

//////////////////// PCA PART ////////////////////////////////

  CH = 0x00;
  CL = 0x00;
  CCAPM0 = 0x11;
  CCAPM1 = 0x21;
  CCAPM2 = 0x42;
  CCAPM3 = 0x42;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CCAP2L = 240;
CCAP2H = 240;
CCAP3L = 238;
CCAP3H = 238;

////////////////////////////////////////////////////////////////

EA = 1;
EC = 1;
ES = 1;
TF1 = 0;
TF0 = 0;
RI = 0;
TI = 0;
PPC = 1;
IPH = 0x40;
cnt = 0 ;
x1  = 0;
TR1 = 1;
TR0 = 1;
CCON = 0x40;
display = read_angle();
checkpoint = read_angle();
sum  = 0;
en1 = 0;
en2 = 0;
inA = 0;
inB = 0;
inC = 0;
inD = 0;
 while(1)
{
  checkpoint = read_angle();
  if(checkpoint == point)
   {
en1 = 0;
en2 = 0;
inA = 0;
inB = 0;
inC = 0;
inD = 0;
check = 1;
    }
  send_data();
   checkpoint = read_angle();
   if(checkpoint == point)
    {
en1 = 0;
en2 = 0;
inA = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
inB = 0;
inC = 0;
inD = 0;
check = 1;
    }
 SBUF = counter();
   checkpoint = read_angle();
  if(checkpoint == point)
   {
en1 = 0;
en2 = 0;
inA = 0;
inB = 0;
inC = 0;
inD = 0;
check = 1;
    }
 superhold(8);
    checkpoint = read_angle();
  if(checkpoint == point)
   {
en1 = 0;
en2 = 0;
inA = 0;
inB = 0;
inC = 0;
inD = 0;
check = 1;
    }
 SBUF = 'E';
   checkpoint = read_angle();
   if(checkpoint == point)
    {
en1 = 0;
en2 = 0;
inA = 0;
inB = 0;
inC = 0;
inD = 0;
check = 1;
    }
 superhold(70);
   checkpoint = read_angle();
   if(checkpoint == point)
    {
en1 = 0;
en2 = 0;
inA = 0;
inB = 0;
inC = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
inD = 0;
check = 1;
    }
    if(check==1)
    {
    check = 0;
    SBUF = 'B';
    point = 400;
    superhold(8);
    }

}
}
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้