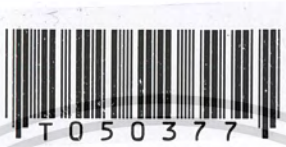
สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

หุ่นยนต์นักสืบ

A Spy Robot



โดย

นายอนุพล อุไรรัตน์

นายอลงกรณ์ สง่าเพ็ชร์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เลขหมู่.....
เลขทะเบียน 50377
วัน,เดือน,ปี 13 พ.ค. 2547

.b.....
.i.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A Spy Robot

BY

MR. ANUPHOL AURAIRAT

MR. ALONGKORN SANGAPETCH

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENT FOR THE DEGREE OF

BACHELOR IN DEPARTMENT OF INDUSTRIAL TECHNOLOGY

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2002

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	หุ่นยนต์นักสืบ	
	A Spy Robot	
นักศึกษาผู้จัดทำ	นายอนุพล อุไรรัตน์	รหัสประจำตัว 43015795
	นายอลงกรณ์ สง่าเพ็ชร์	รหัสประจำตัว 43015797
อาจารย์ที่ปรึกษา	รศ. ดร. ปิติเขต สู้รักษา	
ระดับการศึกษา	ปริญญาตรี อุตสาหกรรมศาสตรบัณฑิต	
	สาขาเทคโนโลยีอิเล็กทรอนิกส์	
ภาควิชา	วิศวกรรมสารสนเทศ	
ปีการศึกษา	2545	

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้รับปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรอุตสาหกรรมศาสตรบัณฑิต

_____อาจารย์ที่ปรึกษา
(รศ. ดร. ปิติเขต สู้รักษา)

_____กรรมการ
( )

_____กรรมการ
( )

_____กรรมการ
( )

_____กรรมการ
( )

ลิขสิทธิ์ของคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	หุ่นยนต์นักสืบ	
นักศึกษาผู้จัดทำ	นายอนุพล อุไรรัตน์	รหัสประจำตัว 43015795
	นายอลงกรณ์ สง่าเพ็ชร์	รหัสประจำตัว 43015797
อาจารย์ที่ปรึกษา	รศ. ดร. ปิติเขต สู้รักษา	
ระดับการศึกษา	ปริญญาตรี อุตสาหกรรมศาสตรบัณฑิต	
	สาขาเทคโนโลยีอิเล็กทรอนิกส์	
ภาควิชา	วิศวกรรมสารสนเทศ	
ปีการศึกษา	2545	

บทคัดย่อ

ในปัจจุบันหุ่นยนต์ได้ถูกนำมาประยุกต์ใช้กันอย่างกว้างขวางทั้งในวงการอุตสาหกรรม งานวิจัยทางวิทยาศาสตร์ หรือแม้กระทั่งเป็นสิ่งอำนวยความสะดวกมนุษย์ และสำหรับงานที่มักจะเห็นกันบ่อยๆ ก็คือ การนำหุ่นมาใช้ในงานเสี่ยงภัย แทนมนุษย์ ดังนั้น ปริญญานิพนธ์เล่มนี้ จึงขอนำเสนอในเรื่องแนวความคิดในการศึกษา ออกแบบ และทำการสร้างหุ่นยนต์นักสืบที่ควบคุมด้วยคอมพิวเตอร์แบบไร้สาย โดยอาศัยภาพจากกล้อง CCD เป็นตัวช่วยในการควบคุมการเคลื่อนที่ และการสอดแนม อันเป็นการลดการเสี่ยงภัยของมนุษย์ที่จะเข้าไปสำรวจ สอดแนมในพื้นที่ที่อันตราย อันจะนำไปสู่การพัฒนาหุ่นยนต์ชาญฉลาดให้มีประสิทธิภาพมากขึ้นในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	A Spy Robot	
Student	Mr. Anuphol Aurairat	ID. 43015795
	Mr. Alongkorn Sangapetch	ID. 43015797
Advisor	Assoc.Prof.Dr. Pitikhate Sooraksa	
Graduate Level	Bachelor of Industrial Technology	
Department	Information Engineering	
Year	2002	

ABSTRACT

At present, the robot is extensively applied in industry, scientific research as well as facility for human. In case of in risk, the robot will always be replaced.

The objective of this thesis is to present ideas, to design and build a spy wireless robot. It is controlled by a computer. It uses picture from CCD Camera that can control the movement and detection in order that the operator can avoid the risk while he is spying in dangerous areas. These will be help leading to development an effectively intelligent robot in the future.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ จะไม่สามารถสำเร็จลุล่วงไปได้ด้วยดี ถ้าขาดคุณพ่อ คุณแม่ ที่คอยช่วยเหลือ เป็นกำลังใจ และสนับสนุนกระผมทางด้านการเรียนมาโดยตลอด อีกทั้งคณาอาจารย์ที่คอยประสิทธิ์ประศาสน์ความรู้ โดยเฉพาะ รศ.ดร.ปิติเขต สู้รักษา อาจารย์ที่ปรึกษาปริญญานิพนธ์ ซึ่งท่านคอยให้คำแนะนำและข้อคิดเห็นต่างๆ ของการทำปริญญานิพนธ์มาด้วยดีตลอด ขอขอบคุณเพื่อนๆ ห้อง 3L ที่คอยช่วยเหลือและเป็นกำลังใจกันมาเป็นอย่างดี

ข้าพเจ้าหวังเป็นอย่างยิ่งว่าปริญญานิพนธ์ฉบับนี้จะเป็นประโยชน์สำหรับผู้ที่สนใจที่จะศึกษาค้นคว้าหาความรู้ต่อไปในภายภาคหน้า

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญรูป	ฉ
สารบัญตาราง	ซ
บทที่ 1 บทนำ	1
1.1 ความเป็นมา	1
1.2 วัตถุประสงค์ของปริญญานิพนธ์	1
1.3 ขอบเขตของปริญญานิพนธ์	1
1.4 องค์ประกอบหลักของโครงงาน	2
1.5 ขั้นตอนการดำเนินโครงงาน	2
บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง	3
2.1 ไมโครคอนโทรลเลอร์	3
2.1.1 คุณสมบัติของไมโครคอนโทรลเลอร์ตระกูล MCS-51	3
2.1.2 การจัดขาของไมโครคอนโทรลเลอร์ตระกูล MCS-51	4
2.1.3 โครงสร้างและการทำงานของพอร์ต	6
2.1.4 การใช้งานเป็นพอร์ตอินพุต	8
2.1.5 การใช้งานเป็นพอร์ตเอาต์พุต	9
2.1.6 การอ่านค่าลอจิกพอร์ต	10
2.1.7 จังหวะการทำงานของไมโครคอนโทรลเลอร์	10
2.2 การสื่อสารอนุกรมด้วยไมโครคอนโทรลเลอร์ตระกูล MCS-51	12
2.2.1 การสื่อสารข้อมูลแบบอะซิงโคนัส (Asynchronous)	12
2.2.2 รีจิสเตอร์ที่เกี่ยวข้องกับการทำงานของพอร์ตอนุกรมใน MCS-51	14
2.2.3 โหมดการทำงานของของพอร์ตอนุกรมใน MCS-51	15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.4 อัตราบอดของพอร์ตอนุกรมในไมโครคอนโทรลเลอร์ตระกูล MCS-51 22
2.2.5 การกำหนดค่าของไทเมอร์เพื่อเลือกอัตราบอด 23
2.2.6 การเขียนหรือส่งข้อมูลออกจากพอร์ตอนุกรม 25
2.2.7 การอ่านหรือรับข้อมูลจากพอร์ตอนุกรม 25
2.3 สเต็ปปิ้งมอเตอร์ 26
2.3.1 การกระตุ้นและควบคุมการหมุนของสเต็ปปิ้งมอเตอร์ 27
2.4 กล้อง CCD 29
2.4.1 การนำภาพ CCD ไปใช้งานถ่ายภาพ 31
2.5 การติดต่อกับพอร์ตอนุกรมด้วยโปแกรมวิชวลเบสิก 32
2.5.1 การเพิ่ม Componemt ที่ใช้ติดต่อกับพอร์ตอนุกรมด้วยโปรแกรมวิชวลเบสิก 32
2.5.2 การเขียนโปรแกรมติดต่อกับพอร์ตอนุกรมด้วยโปรแกรมวิชวลเบสิก 34
บทที่ 3 การออกแบบและการสร้าง 38
3.1 วงจรเปลี่ยนสัญญาณแบบอนุกรมเป็นสัญญาณแบบขนาน 38
3.2 ไมโครคอนโทรลเลอร์ 40
3.3 วงจรขับดีซีมอเตอร์ 41
3.4 วงจรขับสเต็ปปิ้งมอเตอร์ 42
3.5 ชุดเครื่องส่งสัญญาณวีดีโอ(Video Sender) 43
3.6 วงจร Inverting 44
3.7 โปรแกรมควบคุม 45
บทที่ 4 ผลการทดลอง 46
4.1 การติดต่อสื่อสารระหว่างคอมพิวเตอร์กับหุ่นยนต์นักสืบ 46
4.2 การแสดงภาพและเสียง 48
บทที่ 5 สรุปผล ปัญหาและข้อเสนอแนะ 50
5.1 สรุปผล 50
5.2 ปัญหา 50
5.3 ข้อเสนอแนะ 50
บรรณานุกรม 51
ภาคผนวก 52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
1.1 แสดงองค์ประกอบหลักโครงงาน	2
2.1 การจัดขามาตรฐานของไมโครคอนโทรลเลอร์ตระกูล MCS-51	4
2.2 การจัดขามาตรฐานของไมโครคอนโทรลเลอร์ตระกูล MCS-51	6
2.3 รายละเอียดโครงสร้างหลักการทำงานของไมโครคอนโทรลเลอร์ตระกูล MCS-51 แบบแฟลช	7
2.4 วงจรภายในของพอร์ตทุกพอร์ตในไมโครคอนโทรลเลอร์	9
2.5 วงจรพูลอัปภายในพอร์ตของไมโครคอนโทรลเลอร์ตระกูล MCS-51 แบบแฟลช	10
2.6 ไซเกิลการทำงานของไมโครคอนโทรลเลอร์ตระกูล MCS -51 แบบแฟลช	11
2.7 รูปแบบข้อมูลอนุกรมแบบอะซิงโครนัส	13
2.8 แสดงรายละเอียดของรีจิสเตอร์ SCON	14
2.9 ไดอะแกรมการทำงานในโหมด 0 ของพอร์ตอนุกรมภายในไมโครคอนโทรลเลอร์ตระกูล MCS-51	17
2.10 ไดอะแกรมการทำงานในโหมด 1 ของพอร์ตอนุกรมภายในไมโครคอนโทรลเลอร์ตระกูล MCS-51	19
2.11 ไดอะแกรมการทำงานในโหมด 2 ของพอร์ตอนุกรมภายในไมโครคอนโทรลเลอร์ตระกูล MCS-51	20
2.12 ไดอะแกรมการทำงานในโหมด 3 ของพอร์ตอนุกรมภายในไมโครคอนโทรลเลอร์ตระกูล MCS-51	21
2.13 แสดงโครงสร้างพื้นฐานของสเต็ปปิ้งมอเตอร์ยูนิโพล่าร์แบบ 5 และ 6 สาย	26
2.14 แสดงโครงสร้างทั่วไปของกล้อง CCD	29
2.15 แสดงถึงลักษณะรูปคลื่นของแต่ละเฟสที่มีความสัมพันธ์กัน และลักษณะการเคลื่อนที่ของประจุเมื่อมีการถ่ายเท	30
2.16 แสดงลักษณะและขนาดของอุปกรณ์ CCD ในขนาดต่าง ๆ	30
2.17 แสดงส่วนประกอบหลักของระบบกล้อง CCD	31
2.18 การเพิ่ม Component ในวิชวลเบสิก	32
2.19 การเลือก Component ที่ใช้ติดต่อกับพอร์ตอนุกรม	33
2.20 การใส่ Control ที่ใช้ติดต่อกับพอร์ตอนุกรมลงในฟอร์ม	33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่		หน้า
3.1	แสดงองค์ประกอบของบล็อกการทำงานทั้งหมด	38
3.2	รายละเอียดเบื้องต้นของไอซีแปลงสัญญาณเพื่อเชื่อมต่อกับพอร์ตอนุกรมของคอมพิวเตอร์	39
3.3	วงจรเชื่อมต่อ MAX232 เข้ากับพอร์ตอนุกรมของคอมพิวเตอร์และไมโครคอนโทรลเลอร์ตระกูล MCS-51	40
3.4	วงจรใช้งานของ IC เบอร์ TA7279P	41
3.5	แสดงตำแหน่งขาของไอซีเบอร์ ULN2003 สำหรับขับสเต็ปปิ้งมอเตอร์	42
3.6	แสดงวงจรเครื่องส่งสัญญาณภาพ (Video Sender)	44
3.7	แสดงวงจร Inverting	44
3.8	แสดงหน้าจอของโปรแกรม Spy Robot Control	45
4.1	แสดงวิธีการเลือกโหมดการควบคุม	47
4.2	แสดงวิธีการเลือกพอร์ต	47
4.3	แสดงภาพจากหุ่นยนต์นักสืบ	48
4.4	แสดงการตรวจสอบการหมุนของกล้อง	49
4.5	แสดงภาพหุ่นยนต์นักสืบและเครื่องส่ง	49

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
2.1 หน้าที่พิเศษของพอร์ต 1 ในไมโครคอนโทรลเลอร์ตระกูล MCS-51 แบบแฟลช	8
2.2 การเลือกอัตราบอดของวงจรพอร์ตอนุกรมภายในไมโครคอนโทรลเลอร์ตระกูล MCS – 51	24
2.3 แสดงรูปแบบการขับสเต็ปปิ้งมอเตอร์แบบฟูลสเต็ป 1 เฟส	27
2.4 แสดงรูปแบบการขับสเต็ปปิ้งมอเตอร์แบบฟูลสเต็ป 2 เฟส	28
2.5 แสดงรูปแบบการขับสเต็ปปิ้งมอเตอร์แบบฮาล์ฟสเต็ป	28
3.1 ตารางการเปลี่ยนข้อมูลแบบอนุกรมเป็นแบบขนาน	38
3.2 ตารางแสดงการควบคุมดีซีมอเตอร์ด้วย IC TA7279P	42
4.1 แสดงค่าข้อมูลและความหมายข้อมูล	46

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมา

ในปัจจุบันเทคโนโลยีได้มีการพัฒนาการไปอย่างรวดเร็ว ทั้งในส่วนของประสิทธิภาพการทำงานและยังช่วยเพิ่มความสะดวกสบายให้กับผู้ใช้ ทำให้มีการพัฒนาต่อไปอย่างไม่หยุดยั้งและหุ่นยนต์ก็เป็นเทคโนโลยีใหม่ที่ผู้คนต่างให้ความสนใจกันมากและนับเป็นการใช้เทคโนโลยีที่ใหม่อีกขั้นหนึ่ง ในโครงงานนี้ก็จะเป็นการสร้างหุ่นยนต์ซึ่งมีการติดกล้องและไมโครโฟน(Microphone)ที่ตัวหุ่นยนต์ ซึ่งสามารถนำภาพและเสียงส่งกลับมายังผู้ควบคุมได้ ทำให้สามารถรับรู้เหตุการณ์ต่างๆ ไปพร้อมกับการบังคับหุ่นยนต์ซึ่งเป็นสิ่งที่มีประโยชน์สามารถช่วยลดอัตราการเสี่ยงของผู้เข้าไปสอดแนมในสถานที่ต่างๆ นั้น โดยใช้หุ่นยนต์ในการสอดแนมแทน

1.2 วัตถุประสงค์ของปริญญานิพนธ์

ปริญญานิพนธ์นี้จะเป็นการศึกษา ออกแบบ และทำการสร้างหุ่นยนต์สำรวจที่ควบคุมผ่านเครื่องคอมพิวเตอร์แบบไร้สาย และสามารถส่งสัญญาณภาพและสัญญาณเสียงกลับมายังเครื่องคอมพิวเตอร์หรือเครื่องรับโทรทัศน์ ทำให้สามารถมองเห็นเหตุการณ์ต่างๆ โดยที่บุคคลอื่นไม่รู้ตัวเลย เราต้องป้อนข้อมูลโปรแกรมสั่งงานให้กับหุ่นยนต์นั้น ปฏิบัติตามคำสั่ง จะเห็นได้ว่าหุ่นยนต์มีบทบาทในปัจจุบันเป็นอย่างมาก และการสร้างเทคโนโลยีที่ไม่มีวันหยุดยั้งเปลี่ยนแปลงได้เสมอ จึงได้มีการพัฒนารูปแบบหุ่นยนต์ให้ใช้งานได้หลากหลายตั้งแต่เป็นเครื่องทุ่นแรงหรือแม้แต่การสอดแนมข้อมูลในเขตพื้นที่อันตราย

ในการสอดแนมในเขตพื้นที่เสี่ยงอันตราย และมนุษย์ไม่สามารถเข้าไปสอดแนมและเก็บข้อมูลได้โดยตรง จึงจำเป็นอย่างยิ่งที่จะต้องมีเครื่องมือทำหน้าที่แทนมนุษย์ดังนั้นหุ่นยนต์นักสืบจึงมีบทบาททางด้านสืบข้อมูล เช่น สอดแนมข้อมูลลับทางทหาร พื้นที่ที่มีวัตถุระเบิด หรือสารเคมีเป็นต้น โดยการควบคุมจากระยะทางไกล ซึ่งโครงงานนี้เป็นการสร้างหุ่นยนต์นักสืบควบคุมด้วยคอมพิวเตอร์แบบไร้สาย โดยมีกล้อง CCD และไมโครโฟน ขนาดเล็กติดที่ตัวถังและส่งข้อมูลกลับมายังคอมพิวเตอร์

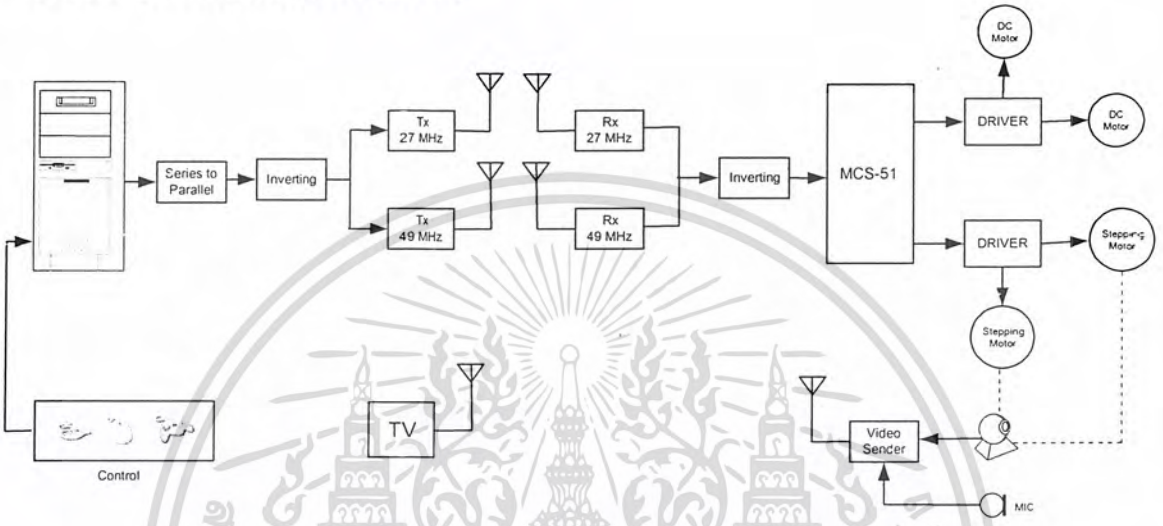
1.3 ขอบเขตของปริญญานิพนธ์

ปริญญานิพนธ์นี้จะกล่าวถึง การควบคุมหุ่นยนต์นักสืบจากเครื่องคอมพิวเตอร์แบบไร้สาย สามารถแสดงภาพและเสียงด้วยโทรทัศน์โดยอาศัยภาพจากกล้อง CCD และไมโครโฟนขนาดเล็กที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ติดอยู่บนตัวหุ่นยนต์ส่งสัญญาณมายังโทรทัศน์ จะมีไมโครคอนโทรลเลอร์ (Microcontroller) ทำหน้าที่ประมวลผลคำสั่งเพื่อควบคุมการเคลื่อนที่ของหุ่นยนต์และกล้อง CCD

1.4 องค์ประกอบหลักของโครงงาน



รูปที่ 1.1 แสดงองค์ประกอบหลักโครงงาน

1.5 ขั้นตอนการดำเนินโครงงาน

เริ่มจากค้นคว้าหาข้อมูล จากหลายๆ แหล่ง เช่น อินเตอร์เนท วารสารอิเล็กทรอนิกส์ หนังสือ ทฤษฎีต่างๆ ซึ่งข้อมูลที่ต้องค้นหาและศึกษามีดังนี้

- ศึกษาข้อมูลการใช้พอร์ตอนุกรมของไมโครคอนโทรลเลอร์ตระกูล MCS-51
- ศึกษาข้อมูลของไอซี (IC) TA7279P
- ศึกษาข้อมูลของเครื่องรับ-ส่งวิทยุเพื่อนำมาประยุกต์ใช้
- ศึกษาการทำงานของเครื่องส่งสัญญาณวีดีโอ
- ศึกษาโปรแกรมวิชวลเบสิก(Visual Basic)

หลังจากศึกษาข้อมูลต่างๆ แล้วก็จะเป็นการออกแบบส่วนของฮาร์ดแวร์ ประกอบไปด้วย ส่วนควบคุมหลัก โดยใช้ไมโครคอนโทรลเลอร์ตระกูล MCS-51 และใช้ไอซี TA7279P เป็นตัวขับดีซีมอเตอร์ (DC Motor) ซึ่งใช้ในการเคลื่อนที่ ไอซี ULN2003 เป็นตัวขับสเต็ปปิ้งมอเตอร์ (Stepping Motor) ซึ่งใช้ในการควบคุมกล้อง วงจร Inverting โดยใช้ทรานซิสเตอร์ วงจรเปลี่ยนสัญญาณแบบอนุกรมเป็นแบบขนาน โดยใช้ไมโครคอนโทรลเลอร์ตระกูล MCS-51

จากนั้นจะเป็นขั้นตอนการออกแบบโปรแกรมเพื่อใช้ในการควบคุม

สุดท้ายจะเป็นการทดลอง และสรุปผลของโครงงานนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
หลักการและทฤษฎีที่เกี่ยวข้อง

2.1 ไมโครคอนโทรลเลอร์

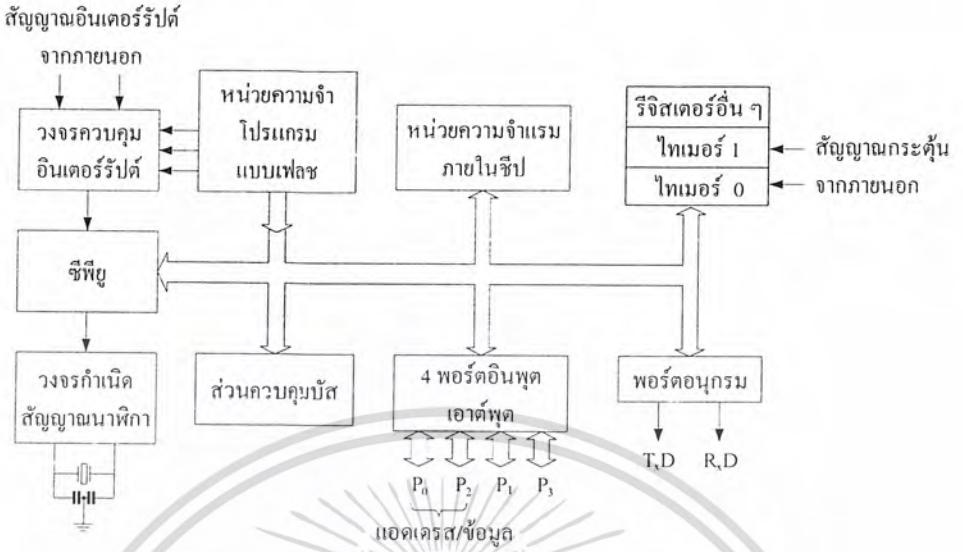
ไมโครคอนโทรลเลอร์เป็นอุปกรณ์อิเล็กทรอนิกส์แบบหนึ่งที่รวมเอาหน่วยประมวลผล (CPU: Central Processing Unit) หน่วยคำนวณทางคณิตศาสตร์และลอจิก (ALU: Arithmetic Logic Unit) วงจรรับสัญญาณอินพุต (Receiver Signal Circuit) วงจรขับสัญญาณเอาต์พุต (Output Driver Circuit) หน่วยความจำ (Memory Unit) และวงจรกำเนิดสัญญาณนาฬิกาไว้ด้วยกัน (Oscillator Unit) ทำให้สามารถนำไปใช้งานแทนวงจรอิเล็กทรอนิกส์ที่ซับซ้อนได้เป็นอย่างดี ช่วยลดจำนวนอุปกรณ์และขนาดของระบบ ในขณะที่มีขีดความสามารถสูงขึ้น ภายใต้งบประมาณที่เหมาะสม

ไมโครคอนโทรลเลอร์มาจากคำ 2 คำรวมกันคือ ไมโคร(Micro) ไมโครโปรเซสเซอร์ (Microprocessor) ซึ่งเป็นอุปกรณ์ประมวลผลข้อมูลขนาดเล็ก ภายในประกอบด้วย หน่วยประมวลผลกลาง หรือ ซีพียู หน่วยคำนวณทางคณิตศาสตร์และลอจิก วงจรเชื่อมต่อหน่วยความจำและวงจรสัญญาณนาฬิกา อีกคำหนึ่งคือคำว่า “คอนโทรลเลอร์” (Controller) หมายถึง อุปกรณ์ควบคุม ดังนั้น ไมโครคอนโทรลเลอร์จึงเป็นอุปกรณ์ที่ใช้ในการควบคุม โดยที่สามารถเขียนโปรแกรมเพื่อกำหนดรูปแบบการควบคุมได้อย่างอิสระ

2.1.1 คุณสมบัติของไมโครคอนโทรลเลอร์ตระกูล MCS-51

- เป็นไมโครคอนโทรลเลอร์ที่ใช้ซีพียูขนาด 8 บิต
- ภายในมีหน่วยความจำโปรแกรมเป็นแบบแฟลช (Flash) สามารถลบและเขียนใหม่ได้พันครั้ง
- หน่วยความจำข้อมูลพื้นฐานเป็นหน่วยความจำแบบแรม (RAM) ในบางเบอร์จะมีหน่วยความจำแบบอีพรอม (EPROM) เพิ่มเติม
- ขาพอร์ตเป็นแบบสองทิศทาง สามารถใช้งานได้ทั้งอินพุตและเอาต์พุต
- มีวงจรสื่อสารอนุกรมแบบฟูลดูเพล็กซ์ (Full Duplex)
- ไทเมอร์หรือเคาน์เตอร์ (Timer/Counter) ขนาด 16 บิตอย่างน้อย 2 ตัว
- สามารถรองรับแหล่งกำเนิดอินเตอร์รัปต์ (Interrupt) ได้ 6 ประเภท
- สามารถขยายหน่วยความจำภายนอกเพิ่มเติมได้สูงสุด 64 กิโลไบต์
- มีวงจรกำเนิดสัญญาณนาฬิกาอยู่ภายในชิป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 การจัดขามาตรฐานของไมโครคอนโทรลเลอร์ตระกูล MCS-51

2.1.2 การจัดขาของไมโครคอนโทรลเลอร์ตระกูล MCS-51

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ทุกเบอร์จะมีสถาปัตยกรรมพื้นฐานเหมือนกัน ดังแสดงในรูปที่ 2.2 และ 2.3 โดยมีรายละเอียดขั้นต้นดังนี้

ขา V_{CC} ใช้สำหรับต่อไฟเลี้ยง +5 V

ขา GND เป็นขากราวด์ สำหรับต่อกับกราวด์ของระบบ

ขาพอร์ต 0 ($P_{0.0}$-$P_{0.7}$) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุต (Input/Output Port) สำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ต 0 (P_0) ขาใดขาหนึ่งเป็นอินพุตสามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (Float) จึงมีอินพุตอิมพีแดนซ์สูง (Hi Input Impedance) สามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ตนี้ยังถูกใช้งานในการติดต่อกับขาแอด-เดรสไบต์ต่ำ ของหน่วยความจำภายนอก (A_0-A_7) และขาข้อมูล (D_0-D_7) โดยใช้กระบวนการมัลติเพล็กซ์ (Multiplex) เข้าช่วย เพื่อสลับการทำงานเป็นได้ทั้งขาติดต่อแอดเดรสและขาข้อมูล

ขาพอร์ต 1 ($P_{1.0}$-$P_{1.7}$) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุต สำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุต สามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย นอกจากนั้นในอนุกรม AT89Sxx จะใช้ขาพอร์ต 1.0 สำหรับนับค่าของไทเมอร์ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขาพอร์ต 2 ($P_{3.0}$-$P_{3.7}$) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุต สำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นพอร์ตอินพุต สามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (Float) จึงมีอินพุตอิมพีแดนซ์สูง สามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้น ขาพอร์ตนี้ยังถูกใช้งานในการติดต่อกับขาแอดเดรสไบต์สูง ของหน่วยความจำภายนอก (A_8-A_{15})

ขาพอร์ต 3 ($P_{3.0}$- $P_{3.7}$) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุต สำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุต สามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (Float) จึงมีอินพุตอิมพีแดนซ์สูง สามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ต 3 (P_3) ยังเป็นขาที่มีหน้าที่การใช้งานพิเศษ ดังมีรายละเอียดขั้นต้นต่อไปนี้

$P_{3.0}$ ใช้เป็นขาอินพุตสำหรับรับข้อมูลจากการสื่อสารแบบอนุกรม หรือขา RxD

$P_{3.1}$ ใช้เป็นขาอินพุตสำหรับส่งข้อมูลจากการสื่อสารแบบอนุกรม หรือขา TxD

$P_{3.2}$ ใช้เป็นขาอินพุตรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 0 หรือขา INT_0

$P_{3.3}$ ใช้เป็นขาอินพุตรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 1 หรือขา INT_1

$P_{3.4}$ ใช้เป็นขาอินพุตสำหรับรับสัญญาณไทเมอร์จากภายนอกช่อง 0 หรือขา T_0

$P_{3.5}$ ใช้เป็นขาอินพุตสำหรับรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 1 หรือขา T_1

$P_{3.6}$ ใช้เป็นขาสัญญาณ WR ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก

$P_{3.7}$ ใช้เป็นขาสัญญาณ RD ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก

ขารีเซต (Reset Pin) ใช้ในการรีเซตการทำงานของไมโครคอนโทรลเลอร์ โดยในการป้อนสัญญาณเพื่อรีเซตสถานะที่ขานี้ ต้องอยู่ในระดับรีเซตอย่างน้อย 2 แมชีนไซเกิล (Machine Cycle) โดยที่วงจรกำเนิดสัญญาณนาฬิกายังคงทำงานต่อเนื่องไปอย่างปกติ

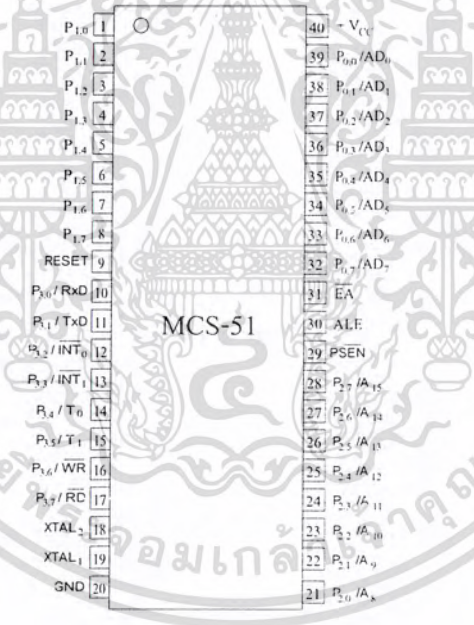
ขา $\overline{\text{ALE/PROG}}$ (Address Latch Enable/Program pulse input) เป็นขาที่ใช้ในการควบคุมการแลตช์(Latch) ของขาพอร์ต 0(P_0) เมื่อมีการใช้งานหน่วยความจำภายนอก นอกจากนั้นขานี้ยังใช้เป็นขาสำหรับรับพัลส์(Pulse) ของการโปรแกรม สำหรับโปรแกรมข้อมูลลงในไมโครคอนโทรลเลอร์ตระกูล MCS-51 ในรุ่นที่มีหน่วยความจำแบบอีพรอม

ขา $\overline{\text{PSEN}}$ (Program Store Enable) ขานี้ใช้ในการส่งสัญญาณ เพื่อร้องขอติดต่อกับหน่วยความจำโปรแกรมภายนอก เมื่อไมโครคอนโทรลเลอร์ต้องการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอก ตัวไมโครคอนโทรเลอร์จะส่งสัญญาณออกมาที่ขานี้ 2 ครั้ง ในแต่ละแมชีนไซเกิล แต่ถ้าหากติดต่อกับหน่วยความจำโปรแกรมข้อมูลภายนอก ขานี้จะไม่มีการส่งสัญญาณใดๆ ออกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขา $\overline{E_a}/V_{pp}$ (External Access Enable/Programming Voltage Input) ใช้สำหรับเลือกการติดต่อหน่วยความจำโปรแกรมจากภายนอก หรือภายในตัวไมโครคอนโทรลเลอร์ ถ้าหากขานี้เป็น "0" เป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำภายนอก แต่ถ้าหากขานี้เป็น "1" เป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำภายในตัวไมโครคอนโทรลเลอร์ นอกจากนี้ ที่ขานี้ยังใช้เป็นขาอินพุตสำหรับรับแรงดันไฟสูง สำหรับการโปรแกรมหน่วยความจำภายในไมโครคอนโทรลเลอร์ สำหรับไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช ต้องการแรงดันสำหรับการโปรแกรม +12 V

ขา $XTAL_1$ และ $XTAL_2$ เป็นขาสำหรับต่อคริสตอลเพื่อสร้างสัญญาณนาฬิกาในการกำหนดจังหวะการทำงานของไมโครคอนโทรลเลอร์



รูปที่ 2.2 การจัดขามาตรฐานของไมโครคอนโทรลเลอร์ตระกูล MCS-51

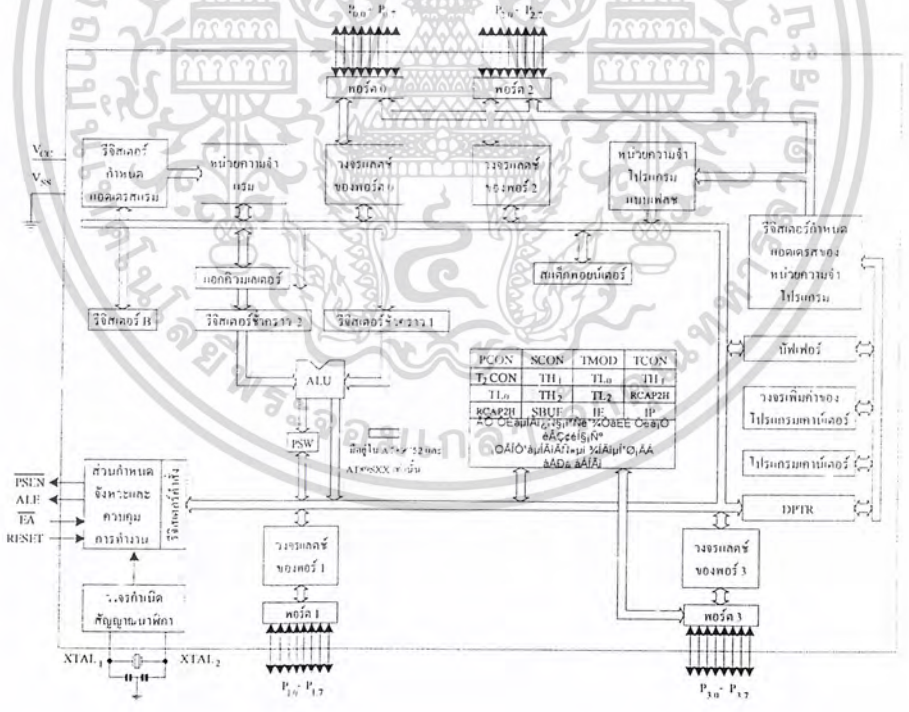
2.1.3 โครงสร้างและการทำงานของพอร์ต

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 แบบแฟลช มีพอร์ตให้ใช้งานทั้งสิ้น 4 พอร์ตคือ พอร์ต 0 ถึงพอร์ต 3 แต่ละพอร์ตมีขนาด 8 บิต เป็นพอร์ตแบบ 2 ทิศทาง กล่าวคือ สามารถเป็นได้ทั้งอินพุตสำหรับรับสัญญาณข้อมูลเข้า และเอาต์พุตสำหรับส่งสัญญาณข้อมูลออก ทุกพอร์ตของไมโครคอนโทรลเลอร์ตระกูล MCS-51 แบบแฟลชมีวงจรแลตช์ (Latch) และวงจรขับตลอดจน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บัฟเฟอร์อินพุต (Buffer Input) ดังแสดงให้เห็นในสถาปัตยกรรมรูปที่ 2.3 ที่พอร์ต 0 และ พอร์ต 2 จะใช้งานเป็นพอร์ตอินพุต และเอาต์พุต สำหรับงานทั่วไป และใช้ในการติดต่อกับหน่วยความจำภายนอก สำหรับพอร์ต 3 และพอร์ต 1 บางขา นอกจากจะใช้เป็นขาพอร์ตอินพุตเอาต์พุตตามปกติแล้ว ยังสามารถใช้งานในหน้าที่พิเศษได้อีก ขึ้นอยู่กับว่าเป็นไมโครคอนโทรลเลอร์ตระกูล MCS-51 แบบแฟลชเบอร์ใด ดังสรุปได้ในตารางที่ 2.1

ในรูปที่ 2.4 แสดงวงจรภายในแต่ละพอร์ต ของไมโครคอนโทรลเลอร์ตระกูล MCS-51 แบบแฟลช โดยในรูป(ก) เป็นวงจรของพอร์ต 0 วงจรแลตช์สามารถกระทำได้อย่างอิสระด้วยสัญญาณที่แยกจากกัน นั้นคือสัญญาณอ่านข้อมูลจากขาพอร์ต และสัญญาณอ่านข้อมูลจากวงจรแลตช์ ส่วนการเขียนข้อมูลมายังพอร์ต ต้องส่งสัญญาณมายังขา CLK ของดีฟลิปฟลอป (D– FLIP FLOP) ในขณะที่ข้อมูลจะผ่านมาทาง ขาบัสข้อมูลภายใน เข้าสู่ขา D ของดีฟลิปฟลอป ที่พอร์ตนี้มีวงจรมัลติเพล็กซ์ สำหรับกำหนดลักษณะการทำงานของพอร์ต ว่าต้องการใช้งานเป็นขาพอร์ตอินพุตเอาร์พุตปกติหรือใช้ในการติดต่อกับหน่วยความจำภายนอกไมโครคอนโทรลเลอร์



รูปที่ 2.3 รายละเอียดโครงสร้างหลักการทำงานของไมโครคอนโทรลเลอร์ตระกูล MCS-51 แบบแฟลช

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากที่ขาพอร์ต 0 (P_0) ไม่มีวงจรพูลอัปภายใน (Internal Pullups) หากมีการนำพอร์ต 0 ไปใช้งานพอร์ตอินพุตจะต้องต่อตัวต้านทานพูลอัปภายนอก เข้าที่ขาพอร์ต 0 ทุกขาด้วย

ในรูปที่ 2.4 (ข) เป็นวงจรของพอร์ต 1 ซึ่งมีลักษณะโดยทั่วไปคล้ายกับพอร์ต 0 หากแต่ไม่มีวงจรมัลติเพล็กซ์ เนื่องจากพอร์ตนี้จะไม่ใช้ในการติดต่อกับหน่วยความจำภายนอก แต่จะมีวงจรพูลอัปภายในที่แต่ละบิตของพอร์ตนี้แทน สำหรับรายละเอียดของวงจรพูลอัปแสดงในรูปที่ 2.4

ในรูปที่ 2.4 (ค) เป็นวงจรภายในของพอร์ต 2 จะคล้ายกับพอร์ต 0 มาก ต่างกันเพียงมีวงจรพูลอัปเพิ่มเติมเข้ามา ส่วนในรูปที่ 2.4 (ง) เป็นวงจรภายในของพอร์ต 3 จะเห็นได้ว่าคล้ายกับพอร์ต 1 มีการเพิ่มเติมวงจรบัฟเฟอร์และวงจรอินพุตเอาต์พุต เมื่อทำงานในฟังก์ชั่นพิเศษเข้ามา เนื่องจากพอร์ต 3 สามารถนำไปใช้งานในหน้าที่พิเศษได้ทุกขา

ตารางที่ 2.1 หน้าที่พิเศษของพอร์ต 1 ในไมโครคอนโทรลเลอร์ตระกูล MCS-51 แบบแฟลช

ขา	เบอร์ของไมโครคอนโทรลเลอร์	หน้าที่พิเศษ
$P_{1.0}$	AT89C52/AT89Sxx	ขา T2 เป็นขาอินพุตนับค่าของไทเมอร์/เคาน์เตอร์ 2
$P_{1.1}$	AT89C52/AT89Sxx	เป็นขาควบคุมทิศทางของสัญญาณ
$P_{1.4}$	AT89Sxx	ขา SS (Slave select) เป็นขาเลือกการติดต่อในกรณีที่ไมโครคอนโทรลเลอร์เป็นอุปกรณ์สเลฟ ในระบบการติดต่อแบบ SPI
$P_{1.5}$	AT89Sxx	ขา MISO (Master data output, Slave data output) ใช้ในการติดต่อกับพอร์ต SPI
$P_{1.6}$	AT89Sxx	ขา MISO (Master data output, Slave data output) ใช้ในการติดต่อกับพอร์ต SPI
$P_{1.7}$	AT89Sxx	ขา SCK (Master clock output) เป็นขาสัญญาณนาฬิกาของการติดต่อกับพอร์ต SPI

2.1.4 การใช้งานเป็นพอร์ตอินพุต

เนื่องจากพอร์ตทั้งหมดของไมโครคอนโทรลเลอร์ตระกูล MCS-51 แบบแฟลชสามารถเป็นได้ทั้งอินพุตและเอาต์พุต ดังนั้นจึงมีความจำเป็นอย่างยิ่งที่ต้องทำความเข้าใจถึงการกำหนดลักษณะการทำงาน ให้แก่พอร์ตไมโครคอนโทรลเลอร์ตระกูล MCS-51 แบบแฟลช

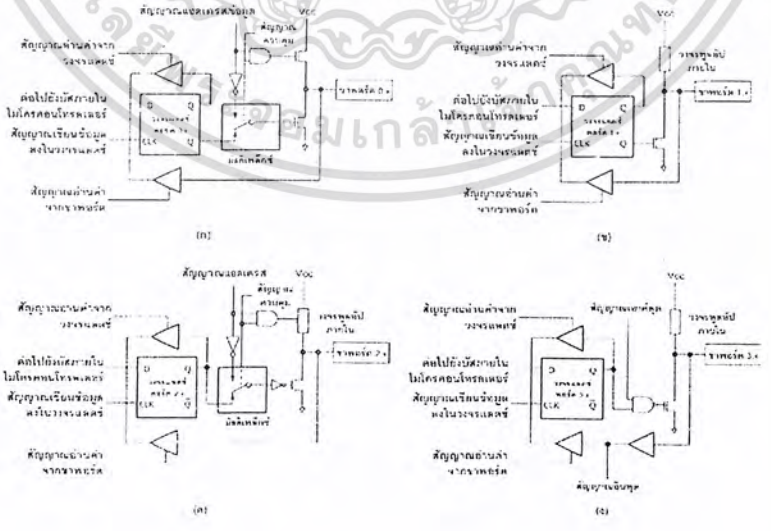
ในการกำหนดให้เป็นพอร์ตอินพุต ต้องเริ่มต้นด้วยการเขียนข้อมูล "1" มาที่แต่ละบิตของพอร์ตที่ต้องการใช้งานเป็นอินพุต เพื่อหยุดการทำงานของเฟต ที่ใช้ในการขับสัญญาณเอาต์พุตของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิตนั้นๆ ทำให้ขาสัญญาณของพอร์ต เชื่อมต่อเข้ากับวงจรพูลอัปภายในโดยตรง ส่งผลให้ขาพอร์ตนั้นมีลอจิกเป็น "1" สามารถรับข้อมูลสัญญาณลอจิก "0" จากอุปกรณ์ภายนอกได้ง่าย สัญญาณข้อมูลจากอุปกรณ์ภายนอกถูกส่งเข้ามาแล้วเก็บไว้ในวงจรบัฟเฟอร์ภายในพอร์ต แล้วรอให้ซีพียูมาอ่านค่าเข้าไป เมื่อเป็นเช่นนี้อุปกรณ์ภายนอกที่เชื่อมต่อกับพอร์ตอินพุตของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชควรกำหนดให้ทำงานในสภาวะลอจิก "0" จะดี และสะดวกที่สุด (ซึ่งในปัจจุบันอุปกรณ์อินพุตที่เชื่อมต่อไมโครคอนโทรลเลอร์แทบทั้งหมดทำงานที่ลอจิก "0" แล้ว)

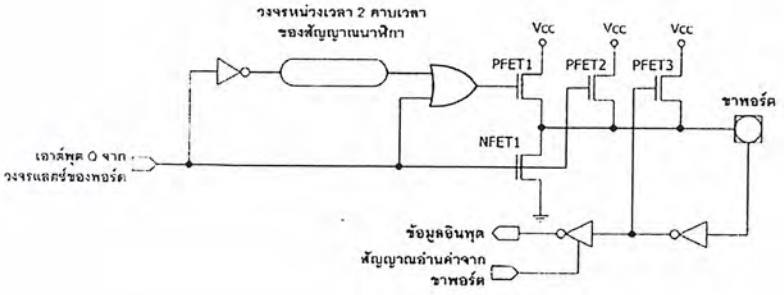
2.1.5 การใช้งานเป็นพอร์ตเอาต์พุต

โดยปกติแล้ว ขาพอร์ตจะกำหนดให้มีลักษณะเป็นเอาต์อยู่แล้ว ดังนั้นจึงสามารถส่งข้อมูลออกไปได้อย่างง่ายดาย และตรงไปตรงมา กล่าวคือ เมื่อต้องการส่งข้อมูล "0" ไปยังวงจรแลตช์ ซึ่งก็จะส่งต่อไปขับเฟต ทำให้เฟตทำงานที่ขาพอร์ตที่กำหนดให้ทำงานก็จะเกิดลอจิก "0" ขึ้น ในทางตรงข้าม หากต้องการส่งข้อมูล "1" ออกไป ก็ให้เขียนข้อมูล "1" ไปยังวงจรแลตช์ วงจรขับก็จะหยุดทำงาน ทำให้ที่ขาพอร์ตเชื่อมต่อกับวงจรพูลอัปภายในเกิดเป็นลอจิก "1" ที่ขาพอร์ตนั้น ซึ่งจะคล้ายกับการกำหนดให้เป็นขาอินพุตมาก เพียงแต่แตกต่างกันที่กระบวนการในการเคลื่อนย้ายข้อมูล โดยถ้าเป็นอินพุตจะมีสัญญาณมาอ่านข้อมูลที่บัฟเฟอร์ แต่ถ้าเป็นเอาต์พุตจะไม่มีการอ่านข้อมูลที่บัฟเฟอร์แต่อย่างใด เว้นแต่ในกรณีที่ต้องการตรวจสอบข้อมูลที่ส่งออกมาทางเอาต์พุต



รูปที่ 2.4 วงจรภายในของพอร์ตทุกพอร์ตในไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



วงจรพูลอัปประกอบด้วยเท่ตรนิตพีแชนเนล 3 ตัวคือ PFET1-PFET3 โดย NFET1 จะทำงานเมื่อได้รับลอจิก "1" จากขา Q และหยุดทำงานเมื่อได้รับลอจิก "0" วงจรพูลอัปจะเริ่มต้นทำงานเมื่อ NFET1 ได้รับลอจิก "1" PFET1 จะทำงานนานประมาณ 2 คาบเวลาของสัญญาณนาฬิกาภายใน หลังจากที่เกิดการเปลี่ยนแปลงจากลอจิก "0" เป็นลอจิก "1" ในขณะที่ PFET1 ทำงาน จะทำให้ PFET3 ทำงานตามไปด้วย ทำให้เกิดการพูลอัปขาพอร์ต

รูปที่ 2.5 วงจรพูลอัปภายในพอร์ตของไมโครคอนโทรลเลอร์ตระกูล MCS-51 แบบแฟลช

เมื่อใช้งานเป็นพอร์ตเอาต์พุตแต่ละขา (หรือแต่ละบิต) ของแต่ละพอร์ต มีความสามารถในการจ่ายกระแสหรือที่เรียกว่า กระแสซอร์ส (Source Current) ได้สูงสุด 10 mA และทุกขารวมกันในแต่ละพอร์ต (ทั้ง 8 บิต) สูงสุด 26 mA สำหรับพอร์ต 0 และ 15 mA สำหรับพอร์ต 1- 3 ในกรณีที่ใช้งานทุกพอร์ตเอาต์พุต จะสามารถจ่ายกระแสได้รวมกันสูงสุด 71 mA ดังนั้นในการใช้งานเป็นพอร์ตเอาต์พุตเพื่อไม่ให้เกิดปัญหาเกี่ยวกับความสามารถในการจ่ายกระแส จึงควรต่อวงจรบัฟเฟอร์ทางเอาต์พุตเพื่อช่วยในการขับกระแสอีกทางหนึ่ง

2.1.6 การอ่านค่าลอจิกพอร์ต

ในไมโครคอนโทรลเลอร์ตระกูล MCS-51 แบบแฟลช สามารถอ่านค่าลอจิกพอร์ตได้ 2 ลักษณะคือ อ่านจากขาพอร์ตโดยตรง และอ่านจากวงจรแลตช์ของแต่ละพอร์ต

ในกรณีที่พอร์ตต่อกับขาเบสของทรานซิสเตอร์ชนิด NPN และขาอิมิตเตอร์ของทรานซิสเตอร์ตัวนั้นต่อลงกราวด์ หากมีการส่งข้อมูล "1" ไปยังทรานซิสเตอร์ จะทำให้ทรานซิสเตอร์ทำงาน สถานะลอจิกที่ขาพอร์ตจะเป็น "0" เนื่องจากเมื่อทรานซิสเตอร์ทำงานจะเสมือนว่าขาพอร์ตนั้นถูกต่อลงกราวด์ ทำให้อ่านค่าลอจิกที่ขาพอร์ต จะได้ผลตรงข้ามกับที่ส่งออกมา แต่ถ้าหากทำการอ่านค่าลอจิกที่วงจรแลตช์ จะได้ค่าที่ตรงกับค่าที่ต้องการส่งจริง ดังนั้น ในการอ่านค่าลอจิกจากพอร์ตจึงต้องเลือกวิธีการให้เหมาะสมกับอุปกรณ์ที่นำมาต่อด้วย

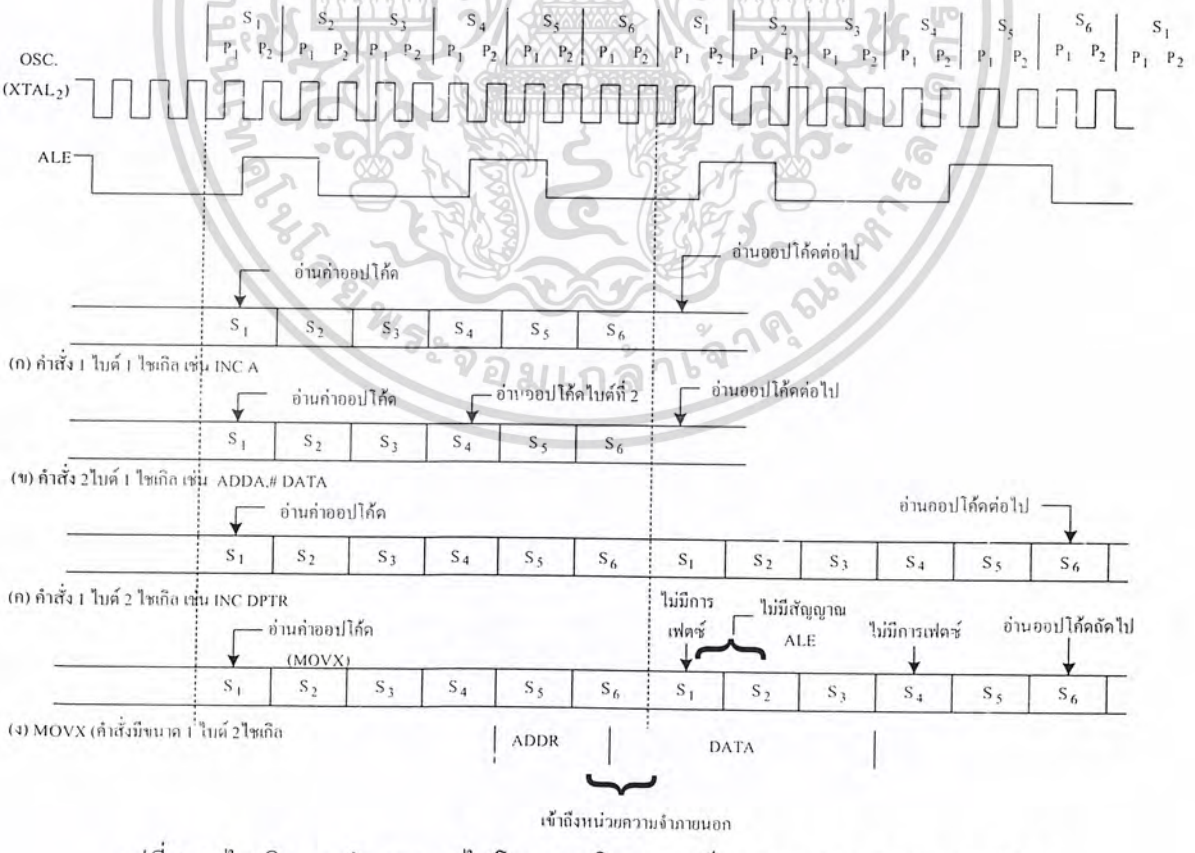
2.1.7 จังหวะการทำงานของไมโครคอนโทรลเลอร์ตระกูล MCS-51

ในการใช้งานไมโครคอนโทรลเลอร์ตระกูล MCS-51 จะต้องทำความเข้าใจถึงจังหวะการทำงานของซีพียู และลำดับขั้นตอนการประมวลผลคำสั่ง ในการประมวลผลคำสั่งของซีพียูจะมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนหลักๆ 2 ขั้นตอน คือ กระบวนการเฟตช์ (fecth) เป็นการเรียกคำสั่งออกจากหน่วยความจำโปรแกรมแล้วทำการแปลงรหัสคำสั่งนี้เป็นภาษาเครื่อง เพื่อเตรียมการประมวลผล ขั้นตอนต่อมาคือกระบวนการเอ็กซิคิวต์ (execute) เป็นการกระทำตามคำสั่งที่กำหนดหรือตามที่เฟตช์ขึ้นมา โดยกระบวนการก่อนหน้านี้ เมื่อทำการเอ็กซิคิวต์คำสั่งเรียบร้อยแล้ว ก็จะไปเริ่มกระบวนการเฟตช์คำสั่งใหม่ต่อไป

เมื่อเริ่มจ่ายไฟให้กับไมโครคอนโทรลเลอร์จะเกิดการรีเซต ในลักษณะที่เรียกว่า เพาเวอร์ออนรีเซต(Power-on reset) ซีพียูเริ่มต้นการทำงานที่แอดเดรส 0000H ของหน่วยความจำโปรแกรม จังหวะการทำงานของซีพียูจะเป็นไปตามรูปแบบ โดยได้รับการกำหนดมาจาก รอบการทำงานหรือแมชีนไซเกิล (Machine Cycle) ในรูปที่ 2.6 เป็นไดอะแกรมเวลาแสดงจังหวะการทำงานของไมโครคอนโทรลเลอร์ตระกูล MCS-51 โดยใน 1 รอบการทำงานหรือ 1 แมชีนไซเกิลจะแบ่งย่อยออกเป็น 6 สเตต (State) กำหนดชื่อเป็น S1 – S6 ในแต่ละสเตตมีค่าเวลาเท่ากับ 2 คาบเวลาของสัญญาณนาฬิกา สัญญาณนาฬิกามีความถี่ 12 MHz จะมีคาบเวลาเท่ากับ 1 ms คาบเวลาทั้งสองภายในหนึ่งสเตตจะเรียกว่า เฟส 1 (Phase 1) และ เฟส 2 (Phase 2)



รูปที่ 2.6 ไซเกิลการทำงานของไมโครคอนโทรลเลอร์ตระกูล MCS -51 แบบแฟลช

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 2.6 (ก) และ (ข) จะเป็นการเอ็กซิคิวต์คำสั่งที่ใช้เวลา 1 ไซเกิล (Cycle) เริ่มต้นที่สเตต 1 ซึ่งจะเป็นการอ่านค่าออปโค้ด อันเป็นกระบวนการแลตช์ค่าของออปโค้ด แล้วส่งไปให้รีจิสเตอร์คำสั่ง (Instruction Register: IR) การเฟตช์ครั้งที่สองจะเกิดขึ้นที่สเตต 4 ภายในแมชีนไซเกิลเดียวกัน ในกรณีที่เป็นคำสั่งไบต์เดียวการเฟตช์ครั้งที่ 2 ภายในแมชีนไซเกิลเดียวกันจะถูกตัดทิ้งออกไปในคำสั่งที่มีการใช้เวลา 1 ไซเกิล จะสิ้นสุดการทำงานลงในสเตต 6 ของแมชีนไซเกิลเดียวกัน

ในกรณีที่คำสั่งใช้เวลา 2 ไซเกิล การทำงานของคำสั่งนั้นจะสิ้นสุดลงในสเตต 6 ของแมชีนไซเกิลที่สอง ดังในไดอะแกรมรูปที่ 2.6 (ค) สำหรับในการกระทำคำสั่ง MOVX ซึ่งเป็นคำสั่งขนาด 1 ไบต์ 2 ไซเกิล จะไม่มีการเฟตช์เกิดขึ้นในไซเกิลที่สองของคำสั่ง MOVX นี้ เนื่องจากซีพียูจะไปทำการติดต่อกับหน่วยความจำภายนอก ดังแสดงในไดอะแกรมรูปที่ 2.6 (ง) จะเห็นได้ว่าเวลาในการเอ็กซิคิวต์ จะไม่ได้ขึ้นอยู่กับว่า ทำการติดต่อหน่วยความจำโปรแกรมภายในหรือ ภายนอก

2.2 การสื่อสารอนุกรมด้วยไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 มีวงจรสื่อสารแบบฟลูดูแพล็กซ์ 1 ชุด (วงจรสื่อสารแบบฟลูดูแพล็กซ์ หมายถึง วงจรสื่อสารที่สามารถทำการรับและส่งข้อมูลในลักษณะ 2 ทิศทางได้ในเวลาเดียวกัน) โดยใช้ขาสัญญาณของพอร์ต 3 คือ ขา P3.0 เป็นขารับข้อมูลเข้าหรือ RxD และขา P3.1 เป็นขาส่งข้อมูลออกหรือ TxDโดยวงจรสื่อสารข้อมูลแบบอนุกรมของไมโครคอนโทรลเลอร์ตระกูล MCS-51 แบบแฟลชเป็นแบบอะซิงโครนัส ปกติแล้วพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51 จะใช้ในการติดต่อสื่อสารกับพอร์ตอนุกรมของคอมพิวเตอร์ โดยใช้มาตรฐาน RS-232 โดยใช้ไอซีพิเศษทำหน้าที่ในการแปลงสัญญาณการสื่อสารดังกล่าว

2.2.1 การสื่อสารข้อมูลแบบอะซิงโครนัส (Asynchronous)

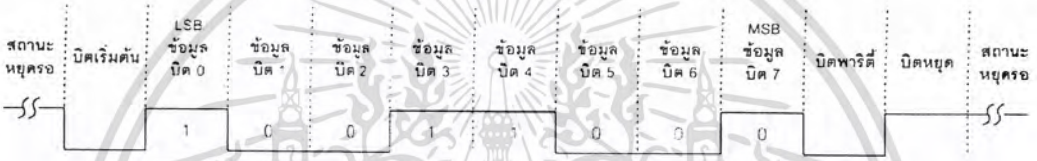
การสื่อสารข้อมูลแบบอะซิงโครนัส คือ การรับและส่งข้อมูลโดยไม่จำเป็นต้องมีสัญญาณนาฬิการ่วมด้วยแต่จะใช้การกำหนดค่าอัตราเร็วในการรับและส่งข้อมูลให้มีค่าเท่ากัน ซึ่งเรียกอัตราเร็วนี้ว่า อัตราบอด หรือ บอดเรต (Baud rate) มีหน่วยเป็น บิตต่อวินาที (bit per second: bps)

รูปแบบของข้อมูลที่ใช้ในการรับส่งแบบอะซิงโครนัสประกอบด้วย 4 ส่วนด้วยกันคือ

1. บิตเริ่มต้น (start bit) มีขนาด 1 บิต
2. บิตข้อมูลแบบอนุกรม มีนาด 8 บิต
3. บิตตรวจสอบพาริตี้ (parity bit) มีขนาด 1 บิตหรือไม่มี
4. บิตปิดท้ายหรือบิตหยุด (stop bit) มีขนาด 1 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.7 แสดงรูปแบบของข้อมูลอนุกรมแบบอะซิงโครนัสเมื่อไม่มีการส่งข้อมูล ขา DATA จะมีสถานะลอจิก "1" เรียกสถานะนี้ว่า สถานะหยุดรอ (waiting stage) การเริ่มต้นส่งข้อมูลจะเริ่มจากการให้ขา DATA มีลอจิก "0" ด้วยช่วงระยะเวลา 1 บิต เรียกบิตนี้ว่า บิตเริ่มต้น (start bit) จากนั้นบิตข้อมูลจะถูกส่งออกไป โดยเริ่มจากบิตที่มีนัยสำคัญต่ำสุดหรือบิต LSB ก่อน ซึ่งข้อมูลที่ต้องการส่งมีจำนวน 8 บิต จากนั้นตามด้วย บิตพาริตี้ (parity bit) ซึ่งใช้ในการตรวจสอบความผิดพลาดที่เกิดขึ้นจากการส่งข้อมูล บิตสุดท้ายที่จะส่งคือ บิตปิดท้าย หรือ บิตหยุด (stop bit) โดยจะเป็นการทำให้ขา DATA มีสถานะลอจิก "1" อีกครั้งด้วยระยะเวลาอย่างน้อย 1 บิต, 1.5 หรือ 2 บิต เพื่อเป็นการแสดงว่าสิ้นสุดข้อมูลแล้ว



รูปที่ 2.7 รูปแบบข้อมูลอนุกรมแบบอะซิงโครนัส

อัตราความเร็วในการรับและส่งข้อมูลของการรับส่งข้อมูลแบบอะซิงโครนัสหรืออัตราบอดหรือบอดเรตที่ใช้สำหรับพอร์ตอนุกรม RS-232 มีด้วยกันหลายค่า ตั้งแต่ 110 ถึง 19,200 บิตต่อวินาที โดยมีค่ามากขึ้นตามเทคโนโลยีของคอมพิวเตอร์ เนื่องจากอัตราบอดคือค่าของจำนวนบิตที่สามารถส่งได้ใน 1 วินาที สมมติว่าข้อมูลอนุกรมมีขนาด 8 บิต ไม่มีการตรวจสอบพาริตี้ มีบิตเริ่มต้น 1 บิตและปิดท้าย 1 บิต ความยาวของข้อมูล 1 ไบต์จะมีความยาวเท่ากับ 10 บิต ถ้าใช้บอดเรตในการส่งข้อมูลเท่ากับ 9,600 บิตต่อวินาทีก็จะสามารถรับส่งข้อมูลได้ด้วยความเร็ว 960 ไบต์ต่อวินาที

การตรวจสอบพาริตี้สามารถกำหนดให้เป็นแบบคี่ (odd), แบบคู่ (even) หรือไม่มีการตรวจสอบพาริตี้ก็ได้พาริตี้คี่หรือพาริตี้คู่แสดงถึงจำนวนลอจิก "1" ทั้งหมดภายในข้อมูลที่ส่งไป 1 ไบต์รวมบิตพาริตี้ว่ามีจำนวนเป็นเลขคู่หรือเลขคี่ ยกตัวอย่าง ข้อมูลที่จะทำการส่งมีขนาด 8 บิต มีค่าเท่ากับ 99H หรือ 100 1001B จะเห็นว่าข้อมูลในไบต์นี้มีจำนวนลอจิก "1" จำนวน 4 ตัวซึ่งเป็นเลขคู่ ดังนั้นถ้ากำหนดค่าพาริตี้เป็นคู่ ค่าของบิตพาริตี้จะต้องมีลอจิกเป็น "0" แต่ถ้ากำหนดพาริตี้เป็นคี่ ค่าของบิตพาริตี้จะต้องเป็น "1" เพื่อให้ข้อมูล 1 ไบต์รวมทั้งบิตพาริตี้เป็นคี่

บิตพาริตี้ถูกสร้างจากภาคส่งข้อมูลของ UART (Universal Asynchronous Receiver Transmitter : เป็นอุปกรณ์ที่ใช้ในการรับและส่งข้อมูลอนุกรม) ซึ่งทางภาครับต้องกำหนดการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตรวจสอบพาริตี้ที่ตรงกันเอาไว้ว่าจะตรวจสอบพาริตี้คี่หรือคู่ จากนั้นภาครับของ UART จะทำการตรวจสอบค่าพาริตี้ที่เกิดขึ้นว่าเป็นคู่หรือเป็นคี่โดยการนับจำนวนลอจิก "1" ทั้งหมดรวมทั้งบิตพาริตี้ด้วย ถ้ากำหนดพาริตี้ไว้เป็นคู่แต่อ่านค่าตัวเลขในการนับออกมาได้ตัวเลขเป็นคี่ ทางภาครับจะแสดงข้อผิดพลาดออกมาให้ผู้ใช้ทราบ กระบวนการดังกล่าวเป็นวิธีการตรวจสอบความผิดพลาดที่เกิดขึ้นในการรับส่งข้อมูลที่ง่ายที่สุด แต่มันสามารถตรวจสอบได้เมื่อมีบิตข้อมูลที่ทำการรับส่งผิดพลาดเพียงบิตเดียวเท่านั้น ถ้าข้อมูลที่ทำการส่งบิตมีบิตที่ผิดพลาดมากกว่า 1 บิต การตรวจสอบด้วยวิธีนี้จะไม่ได้ผลสำหรับการตั้งพาริตี้เป็น none นั้นทั้งภาครับและส่งจะไม่มีการตรวจสอบพาริตี้

2.2.2 รีจิสเตอร์ที่เกี่ยวข้องกับการทำงานของพอร์ตอนุกรมใน MCS-51

ในการทำงานของพอร์ตอนุกรมในไมโครคอนโทรลเลอร์ตระกูล MCS-51 มีรีจิสเตอร์ที่เกี่ยวข้องคยู่ 2 ตัว ดังนี้

- รีจิสเตอร์บัฟเฟอร์ของพอร์ตอนุกรมหรือ SBUF (Serial data buffer register)

มีแอดเดรสอยู่ที่ 99H ในพื้นที่ของรีจิสเตอร์ฟังก์ชั่นพิเศษหรือ SFR มีขนาด 8 บิต แบ่งเป็น 2 ส่วนคือ รีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูล (transmit buffer register) และรับข้อมูล (receive buffer register) เมื่อมีการเขียนข้อมูลมายังรีจิสเตอร์ SBUF ข้อมูลนั้นจะถูกส่งต่อไปยังบัฟเฟอร์สำหรับส่งข้อมูล เพื่อส่งออกจากไมโครคอนโทรลเลอร์ผ่านขาทาง TxD หรือขา P3.1 ในกรณีที่มีการอ่านข้อมูลจากรีจิสเตอร์ SBUF ข้อมูลจะถูกส่งผ่านไปยังรีจิสเตอร์บัฟเฟอร์สำหรับรับข้อมูลเพื่อส่งต่อไปยังไมโครคอนโทรลเลอร์ต่อไป สำหรับการรับข้อมูลอนุกรมจากภายนอกนั้นจะผ่านมาทางขา RxD หรือ P3.0 ของไมโครคอนโทรลเลอร์ตระกูล MCS-51 แบบแฟลช

- รีจิสเตอร์ควบคุมการทำงานของพอร์ตอนุกรมหรือ SCON (Serial port Control Register)

เป็นรีจิสเตอร์ขนาด 8 บิต มีแอดเดรสอยู่ที่ 98H ในพื้นที่ของรีจิสเตอร์ SFR สามารถเข้าถึงได้ในระดับบิต มีรายละเอียดการทำงานดัง นี้

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

รูปที่ 2.8 แสดงรายละเอียดของรีจิสเตอร์ SCON

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SM0-SM1 (Serial port mode bit 0-1): ใช้ในการเลือกโหมดการทำงานของพอร์ตอนุกรมภายในไมโครคอนโทรลเลอร์ตระกูล MCS-51 ดังมีรายละเอียดต่อไปนี้

SM2: ใช้ในการเอ็นเอเบิลการสื่อสารในแบบมัลติโปรเซสเซอร์ (multiprocessor) ในการทำงานของโหมด 2 และ3 ของพอร์ตอนุกรมในไมโครคอนโทรลเลอร์ตระกูล MCS-51 ถ้าบิตนี้เป็น "1" บิต RI จะไม่แอกตีฟถ้าบิตที่ 9 ที่รับเข้ามาเป็น "0" (ข้อมูลบิตที่ 9 เก็บไว้ที่บิต RB8) ในการทำงานโหมด 1 ถ้าบิตนี้เซต บิต RI จะไม่แอกตีฟถ้ายังไม่ได้รับบิตหยุด ส่วนในโหมด 0 บิตนี้ไม่มีการใช้งาน

REN (Enable serial reception): ใช้ในการเอ็นเอเบิลการรับข้อมูลของพอร์ตอนุกรม ทำการเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์ถ้าต้องการให้มีการรับข้อมูลต้องเซตบิตนี้ให้เป็น "1"

TB8: ใช้สำหรับเก็บข้อมูลบิตที่ 9 ที่ต้องการส่งออกไปในการทำงานโหมด 2 และ 3 ของพอร์ตอนุกรมในไมโครคอนโทรลเลอร์ตระกูล MCS-51 เซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

RB8: ใช้สำหรับรับข้อมูลบิตที่ 9 ที่เข้ามาในการทำงานโหมด 2 และ 3 ของพอร์ตอนุกรมในไมโครคอนโทรลเลอร์ตระกูล MCS-51 แต่ถ้าหากพอร์ตอนุกรมทำงานอยู่ในโหมด 1 และบิต SM2 เป็น "0" ข้อมูลที่บิต RB8 คือข้อมูลของบิตหยุด (stop bit) สำหรับในการทำงานโหมด 0 บิตนี้จะไม่ใช้งาน บิต RB8 นี้สามารถเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

TI (Transmit Interrupt flag): ใช้ในการแสดงการเกิดอินเตอร์รัปต์เมื่อมีการส่งข้อมูลออกจากพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ตระกูล MCS-51 สามารถเซตได้ด้วยกระบวนการทางฮาร์ดแวร์ เมื่อทำการส่งข้อมูลบิตที่ 8 ไปเรียบร้อยแล้วในการทำงานโหมด 0 ส่วนในการทำงานโหมดอื่น บิตนี้จะเซตเมื่อมีการเริ่มต้นส่งบิตหยุดออกไป การเคลียร์บิตนี้ต้องใช้กระบวนการทางซอฟต์แวร์เท่านั้น

RI (Receive Interrupt flag): ใช้แสดงการเกิดอินเตอร์รัปต์เมื่อมีการรับข้อมูลสู่พอร์ตอนุกรม สามารถเซตได้ด้วยกระบวนการทางฮาร์ดแวร์ เมื่อทำการรับข้อมูลที่บิต 8 เรียบร้อยแล้วในการทำงานโหมด 0 ส่วนในการทำงานโหมดอื่น บิตนี้จะเซตเมื่อสามารถรับบิตหยุดของข้อมูลอนุกรมได้ครึ่งทางแล้ว ยกเว้นในกรณีที่บิต SM2 มีการเซต บิตนี้จะเซตได้ก็ต่อเมื่อการรับบิตหยุดหรือบิตที่ 9 เกิดขึ้นอย่างสมบูรณ์แล้ว การเคลียร์บิตนี้ต้องใช้กระบวนการทางซอฟต์แวร์เท่านั้น

2.2.3 โหมดการทำงานของพอร์ตอนุกรมใน MCS-51

พอร์ตอนุกรมในไมโครคอนโทรลเลอร์ตระกูล MSC-51 สามารถเลือกการทำงานได้ถึง 4 โหมด คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. โหมด 0 เป็นการกำหนดให้พอร์ตอนุกรมทำงานในลักษณะชิฟต์รีจิสเตอร์
2. โหมด 1 เป็นการกำหนดให้เป็น UART ขนาด 8 บิต สามารถเลือกอัตราบอดได้
3. โหมด 2 เป็นการกำหนดให้เป็น UART ขนาด 9 บิต โดยมีอัตราบอดคงที่
4. โหมด 3 เป็นการกำหนดให้เป็น UART ขนาด 9 บิต สามารถเลือกอัตราบอดได้

การเลือกโหมดทำได้ด้วยการกำหนดข้อมูลให้แก่บิต SMO และ SMI ในรีจิสเตอร์ SCON

- การทำงานในโหมด 0 ของวงจรพอร์ตอนุกรม

มีไดอะแกรมการทำงานและไดอะแกรมเวลาแสดงในรูปที่ 2.9 ข้อมูลอนุกรมจะผ่านเข้าและออกทางขา RxD ส่วนขา TxD ทำหน้าที่เป็นสัญญาณนาฬิกาของการเลื่อนข้อมูล (shift clock) ในโหมดนี้มีจำนวนข้อมูล 8 บิต โดยทำการรับและส่งข้อมูลในบิต LSB ก่อน อัตราในการรับส่งข้อมูลหรืออัตราบอดถูกกำหนดไว้คงที่ที่ 1/12 ของความถี่สัญญาณนาฬิกา

เริ่มต้นการส่งข้อมูลด้วยการเขียนข้อมูลที่ต้องการส่งมายังรีจิสเตอร์ SBUF สัญญาณเขียนข้อมูล SBUF แอกตีฟเป็น "1" ที่สเตต 6 เฟส 2 (S62P2) ของแมชีนไซเกิล ส่งมายังวงจรควบคุมการส่ง (TX control) ทำให้วงจรควบคุมเริ่มต้นส่งข้อมูล สัญญาณส่ง จะแอกตีฟเป็น "1" ตลอดการส่งข้อมูล

ข้อมูลจากรีจิสเตอร์ SBUF จะถูกเลื่อนออกที่ขา P3.0 หรือขา RxD ครั้งละบิต ตามจังหวะของสัญญาณนาฬิกาที่ส่งออกมาทางขา P3.1 หรือ TxD โดยสัญญาณนาฬิกาของการเลื่อนข้อมูลจะมีขอบขาลงของสัญญาณที่สเตต 3 เฟส และมีขอบขาขึ้นที่สัญญาณสเตต 6 เฟส 1 ของแต่ละแมชชีนไซเกิลในกระบวนการส่งข้อมูล จนกระทั่งเมื่อส่งข้อมูลครบ 8 บิตแล้ว บิต IT ในรีจิสเตอร์ SCON จะเกิดการเซตเป็นการแจ้งให้ทราบว่า ส่งข้อมูลครบแล้ว หากการอินเตอร์รัปต์จากพอร์ตอนุกรมได้รับการเอ็นเอเบิลไว้ ก็จะเกิดการอินเตอร์รัปต์ขึ้นในระบบ เมื่อเสร็จสิ้นกระบวนการรับข้อมูล สัญญาณส่ง จะกลายเป็น "0" จนกว่าจะเริ่มต้นกระบวนการรับข้อมูลใหม่

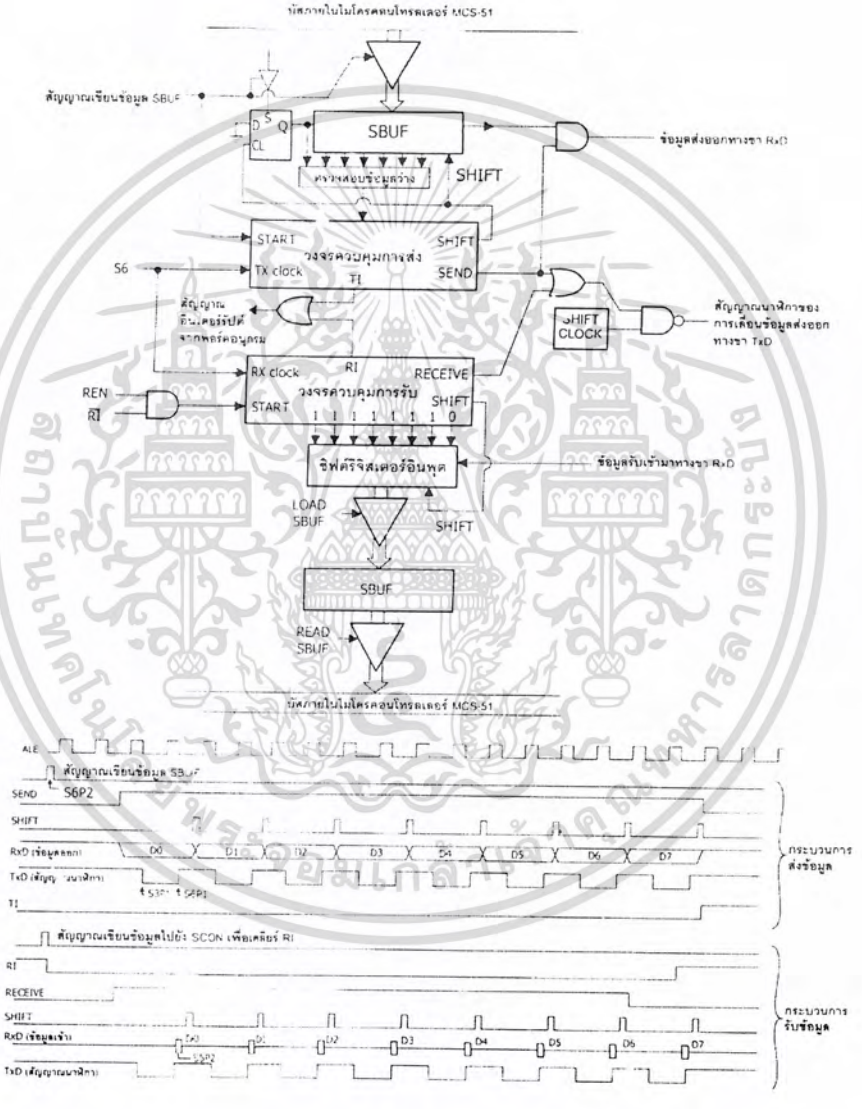
ในกระบวนการรับข้อมูล เริ่มต้นด้วยการเซต REN ให้เป็น "1" และเคลียร์บิต RI ในรีจิสเตอร์ SCON ก่อนที่สเตต 6 เฟส 2 ของแมชีนไซเกิลถัดไป วงจรควบคุมการรับ (RX control) จะทำการเขียนข้อมูล 11111110 ไปยังชิฟต์รีจิสเตอร์สำหรับรับข้อมูลและการทำแอกตีฟสัญญาณรับให้เป็น "1" ในสัญญาณนาฬิกาลูกถัดไป

เมื่อสัญญาณรับแอกตีฟ ก็จะเกิดการส่งสัญญาณนาฬิกาของการเลื่อนข้อมูล (Shift clock) ขึ้นผ่านทางขา P3.1 หรือ TxD เพื่อทำการกำหนดจังหวะการรับข้อมูลครั้งละบิต โดยสัญญาณนาฬิกานี้จะเกิดขึ้นในช่วงสเตต 3 เฟส 1 ถึงสเตต 6 เฟส 1 ของแต่ละแมชีนไซเกิล การรับข้อมูลเข้ามาทางขา P3.0 หรือ RxD จะเกิดขึ้นที่ สเตต 5 เฟส 2 ในแมชีนไซเกิลเดียวกับสัญญาณนาฬิกาของการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เลื่อนข้อมูล จนกระทั่งรับข้อมูลครบทั้ง 8 บิต บิต RI จะได้รับการเซตเพื่อแจ้งการเสร็จสิ้นกระบวนการรับข้อมูล หากการอินเตอร์รัปต์จากพอร์ตอนุกรมได้รับการเอ็นเอเบิลไว้ ก็จะเกิดการอินเตอร์รัปต์ขึ้นในระบบ เมื่อเสร็จสิ้นกระบวนการรับข้อมูล สัญญาณรับจะกลายเป็น “0” จนกว่าจะเริ่มต้นกระบวนการรับข้อมูลใหม่



รูปที่ 2.9 ไดอะแกรมการทำงานในโหมด 0 ของพอร์ตอนุกรมภายในไมโครคอนโทรลเลอร์ตระกูล MCS-51

การทำงานในโหมดนี้ของพอร์ตอนุกรมในไมโครคอนโทรลเลอร์ตระกูล MCS-51 จะใช้ในการเชื่อมต่อกับไอซีรีจิสเตอร์ภายนอกเพื่อทำการขยายจำนวนพอร์ตอินพุตหรือเอาต์พุต แต่ไม่เป็นที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

50377

นิยมใช้งานมากนัก เนื่องจากในไมโครคอนโทรลเลอร์ตระกูล MCS-51 เองมีพอร์ตอยู่ค่อนข้างมาก และติดต่อกับพอร์ตเหล่านั้นได้ง่ายและเร็วกว่ามาก

- การทำงานในโหมด 1 ของวงจรพอร์ตอนุกรม

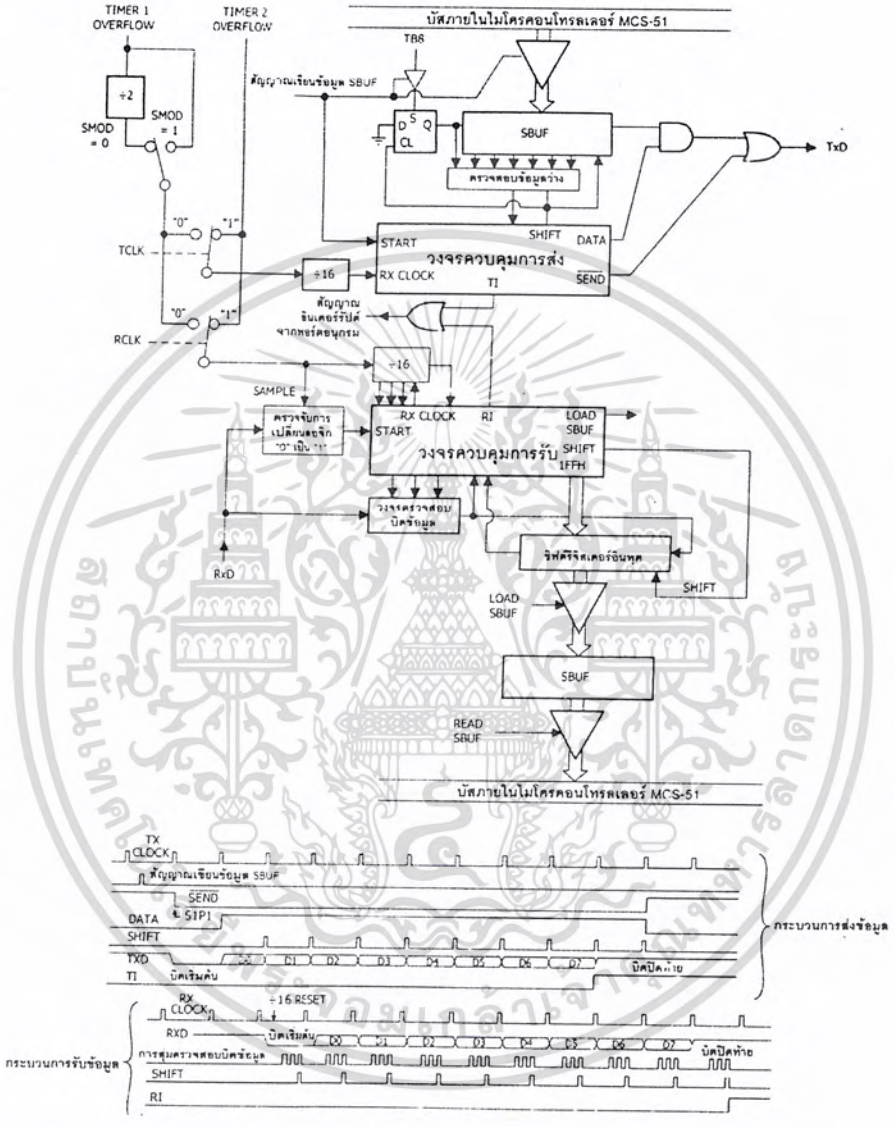
มีไดอะแกรมแสดงในรูปที่ 2.10 ในโหมดนี้ใช้ในการรับส่งข้อมูลรวม 10 บิต โดยส่งข้อมูลออกทางขา P3.1 หรือ TxD และรับข้อมูลเข้าทางขา P3.0 หรือ RxD ข้อมูลทั้ง 10 บิตประกอบด้วย บิตเริ่มต้น (มีค่าเป็น "0") 1 บิต บิตข้อมูล 8 บิต โดยรับหรือส่งข้อมูลในบิต LSB ก่อน และบิตหยุดหรือบิตปิดท้าย (มีค่าเป็น "1") ในการรับข้อมูลบิตหยุดจะถูกเก็บไว้ในบิต RB8 ในรีจิสเตอร์ SCON อัตราบอดในโหมดนี้ได้รับการกำหนดโดยอัตราการเกิดโอเวอร์โฟลวของไทเมอร์ 1 ใน AT89C51 ส่วนในไมโครคอนโทรลเลอร์เบอร์ AT89C52 และในอนุกรม AT89Sxx สามารถเลือกใช้อัตราการเกิดโอเวอร์โฟลวของไทเมอร์ 1 หรือไทเมอร์ 2 ในการกำหนดอัตราบอดได้

กระบวนการส่งข้อมูลเริ่มต้นด้วยการแอกตีฟสัญญาณเขียนข้อมูลมายังรีจิสเตอร์ SBUF ส่งมายังวงจรควบคุมการส่ง (TX control) จากนั้นวงจรควบคุมจะทำการแอกตีฟสัญญาณส่ง ที่สเตต 1 เฟส 1 ของแมชีนไซเกิลต่อมา โดยสัญญาณส่ง จะเป็น "0" ตลอดการส่งข้อมูล เมื่อสัญญาณส่งแอกตีฟ จะทำการส่งบิตเริ่มต้นก่อนเป็นบิตแรก โดยมีคาบเวลาของบิตเริ่มต้นเท่ากับ 1 แมชีนไซเกิล จากนั้นตามด้วยการส่งบิตข้อมูล 8 บิต เรียงลำดับจากบิต LSB โดยข้อมูลที่ทำการส่งถูกเรียบเรียงออกมาจากรีจิสเตอร์บัฟเฟอร์สำหรับการส่งข้อมูลในทุกๆ บิตข้อมูลที่ทำการส่งออกไป จะเกิดสัญญาณพัลส์ SHIFT ขึ้น เพื่อให้เรียกข้อมูลในแต่ละบิตจากรีจิสเตอร์บัฟเฟอร์การกำหนดจังหวะการส่งข้อมูลใช้สัญญาณนาฬิกาส่ง (TX clock) เป็นตัวกำหนด โดยสัญญาณนาฬิกานี้ได้มาจากการหารสัญญาณ TCLK จากไทเมอร์ 1 ด้วย 16 หลังจากการส่งบิตข้อมูลก็จะทำการส่งบิตหยุดหรือบิตปิดท้าย 1 บิต ดังนั้นการส่งข้อมูลจะใช้สัญญาณนาฬิกาทั้งหมด 10 ลูก เมื่อทำการส่งข้อมูลครบเรียบร้อยแล้ว จะทำการเซตบิต TI ในรีจิสเตอร์ SCON หากการอินเตอร์รัปต์จากพอร์ตอนุกรมได้รับการเอ็นเอเบิลไว้ก็จะเกิดการอินเตอร์รัปต์ขึ้นในระบบ หลังจากที่ทำการบริการอินเตอร์รัปต์หรือส่งข้อมูลเรียบร้อยแล้ว ต้องทำการเคลียร์บิต TI ก่อนเป็นอันดับแรก เพื่อให้การรับส่งข้อมูลทางพอร์ตอนุกรมดำเนินต่อไปได้

ด้านการรับข้อมูล จะทำการตรวจจับการเปลี่ยนแปลงระดับลอจิกจาก "1" เป็น "0" ที่ขา RxD โดยใช้อัตราการสุ่มเท่ากับ 1/16 เท่าของอัตราบอด เมื่อตรวจจับพบ ไทเมอร์/เคาน์เตอร์ที่ใช้ในการกำหนดอัตราบอดจะรีเซตและทำการเขียนข้อมูล IFFH ไปยังชิฟต์รีจิสเตอร์ ข้อมูลจะเริ่มเดินทางเข้าสู่พอร์ตอนุกรมของไมโครคอนโทรลเลอร์ผ่านทางขา RxD ในการตีความว่าบิตที่เข้ามาเป็น "0" หรือ "1" จะใช้ผลการสุ่มข้างมาก โดยบิตของข้อมูลที่เข้ามาได้รับการแบ่งออกเป็น 16 สเตต การสุ่มสเตตที่ 7, 8 หาก 2 ใน 3 ของการสุ่มพบว่าข้อมูลเป็นลอจิกใด จะตีความข้อมูลในบิตนั้นเป็นตาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เสียงข้างมาก ยกตัวอย่าง สุ่มพบลอจิก “1” 2 ใน 3 ครั้ง จะตีความว่าบิตของข้อมูลที่ได้รับนั้นเป็น “1”



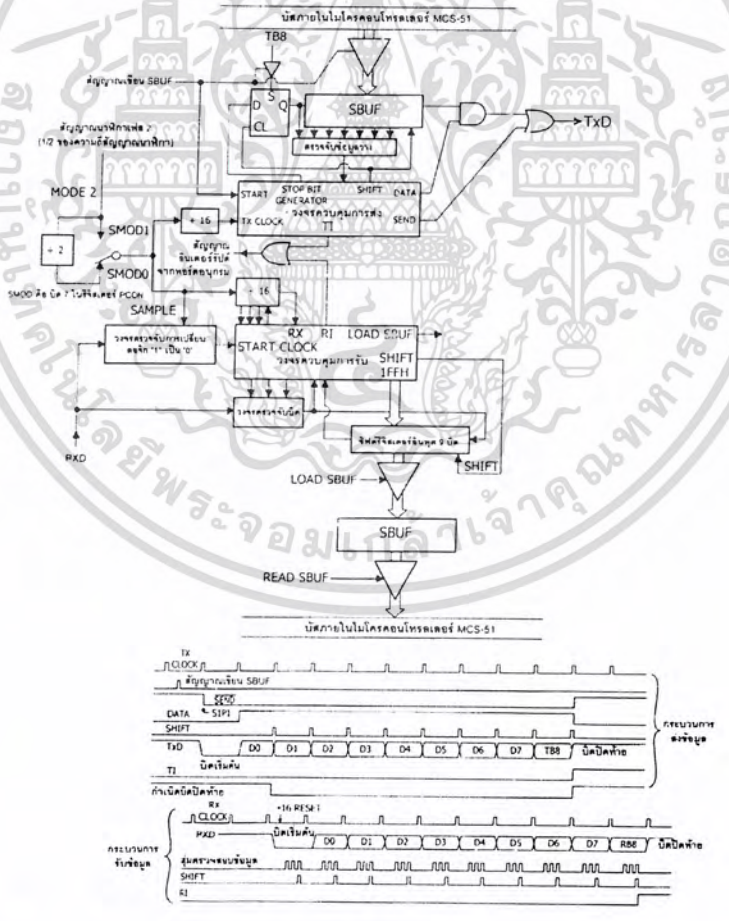
รูปที่ 2.10 ไดอะแกรมการทำงานในโหมด 1 ของพอร์ตอนุกรมภายในไมโครคอนโทรลเลอร์ตระกูล MCS-51

ลำดับของการรับข้อมูลมีลักษณะเดียวกับการส่งข้อมูลคือ เริ่มด้วยบิตที่เริ่มต้นก่อน ตามด้วยบิตข้อมูล และบิตปิดท้ายในทุกๆ การรับข้อมูลได้ 1 บิตจะมีพัลส์ SHIFT เกิดขึ้น เพื่อทำการเลื่อนข้อมูลเข้าสู่รีจิสเตอร์บัฟเฟอร์การรับข้อมูล การกำหนดจังหวะการรับข้อมูลใช้สัญญาณนาฬิกาการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รับข้อมูลใช้สัญญาณนาฬิการับข้อมูล (Rx clock) หลังจากสัญญาณนาฬิกาลูกสุดท้าย อันหมายถึงสามารถรับข้อมูลได้ครบแล้ว วงจรควบคุมการรับข้อมูลจะทำการส่งข้อมูลจากรีจิสเตอร์บัฟเฟอร์ไปยังรีจิสเตอร์ SBUF และบิต RB8 ในรีจิสเตอร์ SCON โดยข้อมูลในบิต RB8 ก็คือ ข้อมูลของบิตหยุดนั่นเอง พร้อมกันนั้นยังทำการเซตบิต RI ในรีจิสเตอร์ SCON ด้วย หากการอินเตอร์รัปต์จากพอร์ตอนุกรมได้รับการเอ็นเอเบิลไว้ ก็จะเกิดการอินเตอร์รัปต์ขึ้นในระบบ หลังจากบริการอินเตอร์รัปต์หรือรับข้อมูลเรียบร้อยแล้ว ต้องทำการเคลียร์บิต RI ก่อน เพื่อให้การรับส่งข้อมูลทางพอร์ตอนุกรมดำเนินต่อไปได้

การทำงานในโหมดนี้ได้รับความนิยมสูงสุดเนื่องจากมีกระบวนการที่ไม่ซับซ้อนและสามารถทำการรับส่งข้อมูลกับคอมพิวเตอร์ได้อย่างมีประสิทธิภาพ

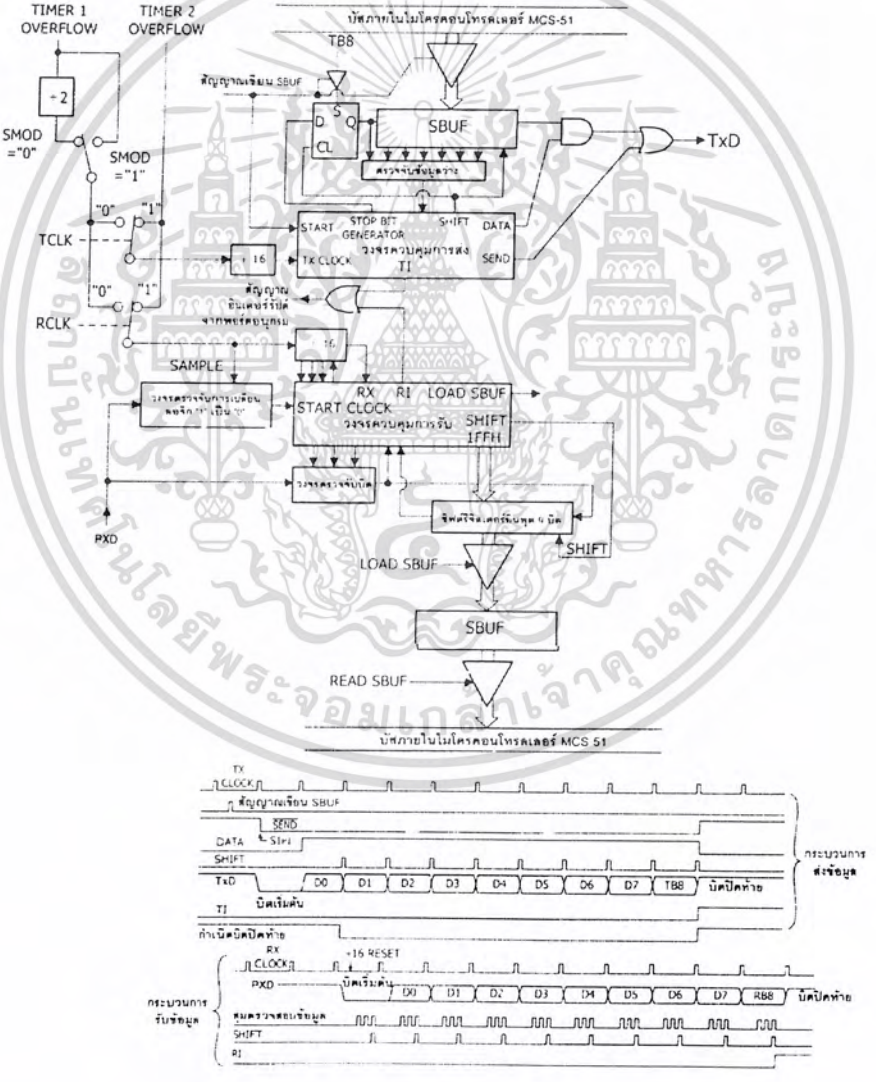


รูปที่ 2.11 ไดอะแกรมการทำงานในโหมด 2 ของพอร์ตอนุกรมภายในไมโครคอนโทรลเลอร์ตระกูล MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การทำงานในโหมด 2 และ 3 ของวงจรพอร์ตอนุกรม

ในทั้งสองโหมดนี้จะใช้รูปแบบข้อมูลรวม 11 บิต ประกอบด้วยบิตเริ่มต้น มีค่าเป็น “0” จำนวน 1 บิต, บิตข้อมูล 8 บิต โดยการทำการรับและส่งบิต LSB ก่อน, บิตข้อมูลบิตที่ 9 และบิตปิดท้ายมีค่าเป็น “1” จำนวน 1 บิตในการรับส่งข้อมูล ข้อมูลบิตที่ 9 จะได้รับการเก็บไว้ในบิต TB8 ในรีจิสเตอร์ SCON และในการรับข้อมูล ข้อมูลบิตที่ 9 จะนำไปเก็บไว้ในบิต RB8 ในรีจิสเตอร์ SCON สำหรับอัตราบอดใน โหมด 2 จะคงที่โดยเลือกได้ 2 ค่า คือ 1/32 หรือ 1/64 ของความถี่สัญญาณนาฬิกา สำหรับในโหมด 3 อัตราบอดสามารถปรับได้เหมือนกับในโหมด 1



รูปที่ 2.12 ไดอะแกรมการทำงานในโหมด 3 ของพอร์ตอนุกรมภายในไมโครคอนโทรลเลอร์ตระกูล MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.11 และ 2.12 เป็นไดอะแกรมการทำงานและไดอะแกรมของการทำงานในโหมด 2 และ 3 ของพอร์ตอนุกรม การทำงานโดยรวมจะคล้ายกับการทำงานในโหมด 1 ส่วนที่แตกต่างกันคือจำนวนบิตของข้อมูลที่ในโหมด 2 และ 3 จะมีเพิ่มมาอีก 1 บิต โดยส่วนใหญ่จะใช้เป็นบิตตรวจสอบพาริตี้

2.2.4 อัตราบอดของพอร์ตอนุกรมในไมโครคอนโทรลเลอร์ตระกูล MCS-51

- โหมด 0

อัตราบอดของโหมดนี้มีค่าคงที่ โดยสามารถคำนวณได้จากสูตร

อัตราบอดในโหมด 0 = ความถี่ของสัญญาณนาฬิกา /12 หน่วยเป็น บิตต่อวินาที

- โหมด 1 และ 3

เนื่องจากทั้งสองโหมดนี้สามารถเลือกแหล่งกำเนิดอัตราบอดได้ 2 แหล่งคือ จากอัตราโอเวอร์โฟลวของไทเมอร์ 1 และ 2 สำหรับอัตราบอดเมื่อใช้การโอเวอร์โฟลวของไทเมอร์ 1 จะต้องใช้ค่าของบิต SMOD ในรีจิสเตอร์ PCON (จะกล่าวถึงรายละเอียดของรีจิสเตอร์ตัวนี้ที่ว่าด้วยการทำงานในโหมดประหยัดพลังงาน) มาพิจารณาประกอบด้วย สามารถคำนวณค่าอัตราบอดได้จาก

อัตราบอด = $(2^{\text{ค่าของบิต SMOD}}/32)$ x อัตราโอเวอร์โฟลวของไทเมอร์ 1

ถ้าหากในไทเมอร์ 1 ไม่ได้เอ็นเอเบิลการอินเตอร์รัปต์ไว้ สามารถคำนวณอัตราบอดได้จาก

อัตราบอด = $(2^{\text{ค่าในรีจิสเตอร์ SMOD}}/32)$ x (ความถี่สัญญาณนาฬิกา/{12 x [256-(TH1)]})

ในตารางที่ 2.2 แสดงการกำหนดอัตราบอดโดยใช้ไทเมอร์ 1

ในกรณีที่ใช้ไทเมอร์ 2 ในการกำหนดอัตราบอด โดยกำหนดให้ไทเมอร์ 2 ทำงานในโหมดกำเนิดอัตราบอด (Baud rate generator) สามารถคำนวณหาค่าอัตราบอดได้จาก

อัตราบอด = อัตราโอเวอร์โฟลวของไทเมอร์ 2/16 หน่วยเป็น บิตต่อวินาที

ถ้าหากกำหนดให้ไทเมอร์ 2 ทำงานในโหมดปกติ สามารถคำนวณหาอัตราบอดได้จาก

อัตราบอด = ความถี่ของสัญญาณนาฬิกา /(32 x (65536 – (RCAP2H,RCAP2L)))

โดยที่ (RCAP2H, RCAP2L) เป็นค่าของรีจิสเตอร์ RCAP2H และ RCAP2L มีขนาด 16 บิต ไม่คิดเครื่องหมาย

- โหมด 2

ในโหมดนี้อัตราบอดจะขึ้นอยู่กับค่าของบิต SMOD ในรีจิสเตอร์ PCON ถ้า SMOD เป็น "0" อัตราบอดจะเท่ากับ 1/64 ของความถี่สัญญาณนาฬิกาในกรณีที่ SMOD เป็น "1" อัตราบอดจะเท่ากับ 1/32 ของความถี่สัญญาณนาฬิกา สามารถแสดงเป็นสูตรคำนวณทางคณิตศาสตร์ได้ดังนี้

อัตราบอด = $(2^{\text{ค่าของบิต SMOD}}/64)$ x ความถี่สัญญาณนาฬิกา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.5 การกำหนดค่าของไทเมอร์เพื่อเลือกอัตราบอด

ในการใช้งานพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ตระกูล MCS-51 สิ่งที่ต้องให้ความสนใจมากที่สุดประการหนึ่ง คือ อัตราการถ่ายทอดข้อมูล หรือ อัตราบอด ซึ่งการกำหนดอัตราบอดนั้นจะขึ้นอยู่กับค่าความถี่ของสัญญาณนาฬิกาเป็นหลัก สำหรับโหมดการทำงานของพอร์ตอนุกรมที่สามารถเลือกอัตราบอดได้อย่างอิสระคือ ในโหมด 1 และ 3 โดยกำหนดได้จากอัตราการเกิดโอเวอร์โฟลวของไทเมอร์ 1 ถ้าหากไทเมอร์ 1 มีการเกิดโอเวอร์โฟลวในอัตราที่สูงมากเท่าใด อัตราบอดก็จะมีค่าสูงมากขึ้นตาม นั่นหมายความว่า อัตราในการถ่ายทอดข้อมูลจะสูงมาก สามารถถ่ายทอดได้อย่างรวดเร็ว

ในการใช้ไทเมอร์ 1 เพื่อกำหนดอัตราบอดในโหมด 1 และ 3 ของพอร์ตอนุกรมจะต้องกำหนดให้ไทเมอร์ 1 ทำงานในโหมด 2 หรือ โหมด 8 บิตแบบตั้งค่าการนับอัตโนมัติ และการกำหนดค่ารีโหลดให้แก่รีจิสเตอร์ TH1 จึงเป็นตัวแปรหลักที่ใช้ในการกำหนดอัตราบอดให้แก่พอร์ตอนุกรมของไมโครคอนโทรลเลอร์ตระกูล MCS-51

เริ่มต้นด้วยการเคลียร์บิต SMOD ซึ่งเป็นบิต 7 ของรีจิสเตอร์ PCON ให้เป็น "0" ค่าของการรีโหลดลดให้แก่ TH1 สามารถคำนวณได้จาก

TH1 = 256-((ค่าความถี่ของคริสตอล /384)/อัตราบอด)

แต่ถ้าบิต SMOD เกิดการเซต จะเป็นการเอ็นเอเบิลการทวีคูณของอัตราบอด ดังนั้นการกำหนดค่าให้แก่ TH1 จึงต้องคำนวณจาก

TH1 = 256 – ((ค่าความถี่ของคริสตอล /192) อัตราบอด)

ยกตัวอย่าง ถ้าหากในไมโครคอนโทรลเลอร์ AT89C51 ใช้คริสตอล 11.0592 MHz ต้องการกำหนดอัตราบอดของพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ไว้ที่ 19,200 บิตต่อวินาที ในกรณีที่ไม่เอ็นเอเบิลการทวีคูณของอัตราบอด ค่ารีโหลดของไมโครคอนโทรลเลอร์จะเท่ากับ

TH1 = 256 – ((ค่าความถี่ของคริสตอล/384)อัตราบอด)

= 256 – ((11059200/384)/19200)

= 256 – (28800/19200)

= 256 – 1.5 = 254.5

เนื่องจากผลลัพธ์ที่ได้เป็นค่าคงที่ไม่ใช่จำนวณเต็ม ถ้าหากกำหนดค่าของ TH1 เป็น 254 เมื่อทำการแทนค่าเพื่อคำนวณหาอัตราบอด จะได้อัตราบอดเท่ากับ 14,400 บิตต่อวินาที และถ้าหากกำหนดค่าของ TH 1 เป็น 255อัตราบอดจะมีค่าเท่ากับ 28,800 บิตต่อวินาที ดังนั้นจะเห็นได้ว่าค่าของ TH1 ที่ไม่เป็นจำนวนเต็มจะไม่สามารถทำให้เกิดอัตราบอดตามที่ต้องการได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทางแก้ไขคือ ให้ทำการเอ็นเอเบิลการทวีคูณอัตราบอด โดยการเซตบิต SMOD ในรีจิสเตอร์ PCON ให้เป็น "1" จากนั้นแทนค่าลงในสมการหาค่า TH1 เมื่อมีการเซตบิต SOMD ได้ผลดังนี้

TH1 = 256 – ((ค่าความถี่ของคริสตอล/192)/อัตราบอด)

= 256 – ((11059200/192)/19200)

= 256 – (57600/19200)

= 256 – 3 = 253

นำค่าของ TH1 ที่ได้จากการแทนค่าคำนวณหาอัตราบอดได้จะเท่ากับ 19,200 บิตต่อวินาที สามารถสรุปขั้นตอนในการเลือกอัตราบอดโดยการกำหนดค่าของไทเมอร์ 1 ได้ดังตารางที่ 2.2

ตารางที่ 2.2 การเลือกอัตราบอดของวงจรพอร์ตอนุกรมภายในไมโครคอนโทรลเลอร์ตระกูล MCS – 51

อัตราบอด (บิตต่อวินาที :bps)	ความถี่สัญญาณนาฬิกา	SMOD	ไทเมอร์ 1		
			C/T	โหมด	ค่ารีโหลด
โหมด 0 : สูงสุด 1 MHz	12 MHz	x	x	x	x
โหมด 2 : สูงสุด 375K	12 MHz	1	x	x	x
โหมด 1,3 : 62.5 K	12 MHz	1	0	2	FFH
19.2K (19,200)	11.0592 MHz	1	0	2	FHD
9.6K (9,600)	11.0592 MHz	0	0	2	FDH
4.8K (4,800)	11.0592 MHz	0	0	2	FAH
2.4K (2,400)	11.0592 MHz	0	0	2	F4H
1.2 K (1,200)	11.0592 MHz	0	0	2	E8H
137.5	11.0592 MHz	0	0	2	1DH
110	6 MHz	0	0	2	72 H
110	12 MHz	0	0	1	FEEBH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. กำหนดให้พอร์ตอนุกรมของไมโครคอนโทรลเลอร์ตระกูล MCS – 51 ทำงานในโหมด 1 หรือ 3
2. กำหนดให้ไทเมอร์ 1 ทำงานในโหมด 2 หรือ โหมด 8 บิต ตั้งค่าอัตโนมัติ
3. กำหนดข้อมูลให้แก่ TH1 เท่ากับ 253 เพื่อให้สามารถกำเนิดอัตราบอดได้ 19,200 บิตต่อวินาทีตามต้องการ
4. ทำการเซตบิต SMOD ซึ่งเป็นบิต 7 ของรีจิสเตอร์ PCON เพื่อเอ็นเอเบิลการทวีคูณอัตราบอด

2.2.6 การเขียนหรือส่งข้อมูลออกจากพอร์ตอนุกรม

ข้อมูลที่ต้องการส่งออกทุกค่าต้องนำไปเก็บไว้ในรีจิสเตอร์บัฟเฟอร์ของพอร์ตอนุกรมซึ่งก็คือรีจิสเตอร์ SBUF ดังตัวอย่าง

```
MOV   SBUF, # A
```

จากคำสั่งข้างต้นเป็นการส่งข้อมูลของตัวอักษร A ออกไปยังพอร์ตอนุกรมของไมโครคอนโทรเลอร์ อย่างไรก็ตามก่อนทำการส่งข้อมูลทุกครั้งต้องแน่ใจว่าบิต TI เคลียร์หรือไม่มีค่าเป็น "0" และเมื่อทำการส่งข้อมูลเรียบร้อยแล้ว ก็จะเกิดการเซตบิต TI เพื่อแจ้งให้ทราบ ดังตัวอย่างโปรแกรมต่อไปนี้

```
CLR   RI          : เคลียร์บิต TI เพื่อเตรียมการส่งข้อมูลออก
MOV   SBUF,#A     : ส่งข้อมูลของตัวอักษร A ไปยังพอร์ตอนุกรม
JNB   TI, $       : รอการเซตของบิต TI เพื่อแจ้งการส่งข้อมูลที่เสร็จสมบูรณ์
```

2.2.7 การอ่านหรือรับข้อมูลจากพอร์ตอนุกรม

การรับข้อมูลจากพอร์ตอนุกรมสามารถจะกระทำได้ง่ายมาก เพียงทำการตรวจสอบว่าบิต RI เกิดการเซตขึ้นหรือไม่ ถ้าพบว่ามีการเซตเกิดขึ้นแล้ว ให้ทำการอ่านค่าจากรีจิสเตอร์ SBUF โดยต้องทำการโอนย้ายข้อมูลผ่านทางแอกคิวมูลเลเตอร์หรือรีจิสเตอร์ A ดังตัวอย่าง

```
CLR   RI          : เคลียร์บิต TI เพื่อเตรียมการส่งข้อมูลออก
JNB   RI, $       : รอคอยการเซตของบิต RI อันเป็นการแจ้งให้ทราบว่า การรับ
                    ข้อมูลเสร็จสมบูรณ์และมีข้อมูลเกิดขึ้นที่รีจิสเตอร์ SBUF
MOV   A, SBUF     : อ่านค่าจากรีจิสเตอร์โดยการโอนย้ายข้อมูลผ่านทางรีจิสเตอร์ A
CLR   RI          : หลังจากทำการอ่านข้อมูลเรียบร้อยแล้ว ต้องทำการเคลียร์บิต RI
                    เสมอ
```

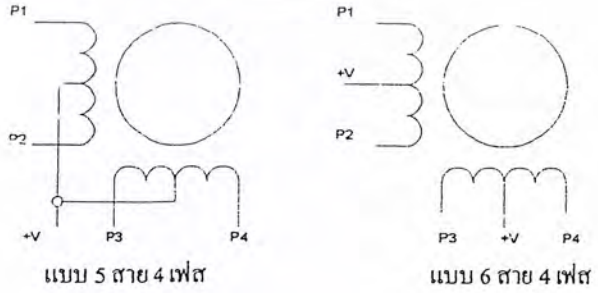
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 สเต็ปปิ้งมอเตอร์

สเต็ปปิ้งมอเตอร์เป็นมอเตอร์ที่มีลักษณะการทำงานแตกต่างจากมอเตอร์ทั่วไปเพราะจะต้องป้อนสัญญาณเป็นพัลส์ให้แก่ขดลวดมอเตอร์เป็นจังหวะอย่างเหมาะสม และการหมุนของมอเตอร์ชนิดนี้จะหมุนเป็นจังหวะตามพัลส์ที่ป้อนเข้ามาไม่หมุนต่อเนื่องเหมือนกับมอเตอร์ธรรมดาทำให้ผู้ควบคุมสามารถเลือกตำแหน่งที่ต้องการให้กับมอเตอร์หมุนได้ จังหวะการหมุนของสเต็ปปิ้งมอเตอร์เรียกว่า สเต็ป (step) ความละเอียดของมอเตอร์กำหนดเป็นองศาที่หมุนไปในหนึ่งสเต็ปหากมอเตอร์มีจำนวนองศาต่อสเต็ปมาก หมายความว่ามอเตอร์ตัวนี้มีความละเอียดของการหมุนต่ำ ยกตัวอย่าง การหมุนครบ 1 รอบเท่ากับ 360 องศา หากมอเตอร์มีการหมุนเท่ากับ 7.5 องศาต่อสเต็ป มอเตอร์นี้มีความละเอียดของการหมุนเท่ากับ 48 ตำแหน่ง แต่ถ้าหากว่าสเต็ปการหมุนเท่ากับ 1.8 องศาต่อสเต็ป ความละเอียดของการหมุนเท่ากับ 200 ตำแหน่ง จะเห็นได้ว่ามอเตอร์มีความละเอียดสูง ทำให้การนำไปใช้กำหนดตำแหน่งได้ดีกว่า แม่นยำกว่า ผนวกเข้ากับวงจรขับแบบฮาล์ฟสเต็ป ความละเอียดของการหมุนจะเพิ่มอีก 2 เท่าซึ่งจะทำให้มีความละเอียดของการหมุนกลายเป็น 400 ตำแหน่ง

ขนาดของสเต็ปปิ้งมอเตอร์ที่ผลิตและจำหน่ายในท้องตลาด มีตั้งแต่ขนาดแรงดัน 3 V ไปถึง 24 V ส่วนขนาดของกระแสมีตั้งแต่ไม่กี่สิบมิลลิแอมป์ซึ่งเป็นสเต็ปปิ้งตัวเล็กไปจนถึงเป็นสิบแอมป์ซึ่งจะเป็นขนาดใหญ่

สเต็ปปิ้งมอเตอร์ได้รับการพัฒนามาอย่างต่อเนื่อง จนในปัจจุบันสเต็ปปิ้งมอเตอร์ที่ได้นิยมใช้กันอย่างแพร่หลายมากที่สุด และหาซื้อได้ง่าย คือ สเต็ปปิ้งมอเตอร์แบบยูนิโพล่าร์ (uni-polar step motor) มีลักษณะการพันขดลวดของมอเตอร์ดังรูปที่ 2.13



รูปที่ 2.13 แสดงโครงสร้างพื้นฐานของสเต็ปปิ้งมอเตอร์ยูนิโพล่าร์แบบ 5 และ 6 สาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สเต็ปปิ้งมอเตอร์แบบนี้มีการพันขดลวด 2 ขดบนแต่ละขั้วแม่เหล็กของสเตเตอร์ (Stator) แต่ละขดแบ่งเป็น 2 เฟส รวมมอเตอร์ทั้งตัวจะมี 4 เฟสคือ เฟส 1, 2, 3 และ 4 มีการต่อสายออกมาจากขดลวดแต่ละขดเพื่อจ่ายไฟเลี้ยง ทำให้สเต็ปปิ้งมอเตอร์แบบนี้มีทั้งแบบ 5 สาย และ 6 สาย

2.3.1 การกระตุ้นและควบคุมการหมุนของสเต็ปปิ้งมอเตอร์

การกระตุ้นและควบคุมการหมุนของสเต็ปปิ้งมอเตอร์ให้เคลื่อนที่ไปแต่ละสเต็ปทำได้โดยจ่ายกำลังไฟฟ้าไปยังขดลวดแต่ละขดลวดบนสเตเตอร์ ซึ่งป้อนเป็นลำดับซีเควนเชียล ในรูปแบบที่ถูกต้องด้วย สามารถแบ่งได้เป็น 3 แบบคือ แบบฟูลสเต็ป 1 เฟส (full step 1 phase) แบบฟูลสเต็ป 2 เฟส และแบบฮาล์ฟสเต็ป (half step) แบบฟูลสเต็ป 1 เฟสเป็นการกระตุ้นที่มีรูปแบบง่ายที่สุด โดยทำการกระตุ้นขดลวดทีละขดในเวลาหนึ่งไล่เรียงถัดกันไป เช่น เริ่มต้นที่ขด 1, 2, 3, 4 แล้ววนกลับมาขดที่ 1 วนไปเรื่อยๆ หรือเริ่มที่ขดที่ 1 แล้วย้อนไปยังขดที่ 4, 3, 2 แล้วกับมายังที่ขดที่ 1 อีกครั้ง ซึ่งทำให้ทิศทางการหมุนตรงกันข้ามกัน ในการกระตุ้นรูปแบบนี้จึงมีเพียงขดเดียวในเวลาหนึ่งที่ถูกกระตุ้นเท่านั้น วงจรกระตุ้นแบบฟูลสเต็ป 1 เฟสจึงมีราคาถูกและหาง่าย ขั้นตอนการทำงานต่างๆ แสดงในตารางที่ 2.3

ตารางที่ 2.3 แสดงรูปแบบการขับสเต็ปปิ้งมอเตอร์แบบฟูลสเต็ป 1 เฟส

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

แบบฟลูสเต็ป 2 เฟสเป็นแบบการกระตุ้นซึ่งคล้ายกับแบบ 1 เฟส แต่การกระตุ้นแบบนี้จะทำการกระตุ้นโดยจ่ายกำลังไฟฟ้าไปที่ขด 2 ขด ที่อยู่ใกล้กันในเวลาเดียวกัน และเรียงถัดกันไป เช่นเดียวกับแบบฟูลสเต็ป 1 เฟส ดังตัวอย่าง ขดลวดชุดแรกที่ถูกกระตุ้นจะเป็นขดที่ 1 และ 2 ตามด้วยการกระตุ้นขดที่ 2 และ 3 ต่อไปเป็นขดที่ 3 และ 4 ถัดไปเป็นขดที่ 4 และ 1 และกลับมาที่ขดที่ 1 และ 2 วนไปตามลำดับเช่นนี้ หรือเริ่มขดที่ 1 และ 4 ทิศทางการหมุนจะตรงกันข้ามกัน การกระตุ้นสเต็ปปิ้งมอเตอร์แบบนี้สามารถเพิ่มแรงบิดได้มากกว่าแบบฟูลสเต็ป 1 เฟส โรเตอร์ (Rotor) จะเคลื่อนที่ด้วยแรงดึงอย่างเต็มแรงจาก 2 ขด ขดลวดที่ถูกกระตุ้นพร้อมกันและแรงดึงจากอีก 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขดลวดถัดไป แต่ข้อเสียคือการกระตุ้นต้องใช้กำลังไฟฟ้ามากขึ้นขั้นตอนการทำงานต่างๆ แสดงในตารางที่ 2.4

ตารางที่ 2.4 แสดงรูปแบบการขับสเต็ปปิ้งมอเตอร์แบบฟูลสเต็ป 2 เฟส

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

แบบฮาล์ฟสเต็ปเป็นรูปแบบที่ผสมผสานระหว่างการกระตุ้นแบบฟลูสเต็ป 1 และ 2 เฟสเพื่อเพิ่มจำนวนของสเต็ปต่อรอบอีกเท่าตัวหนึ่ง ในระบบนี้จะทำการกระตุ้นขดลวดเรียงกันไปเป็นลำดับดังนี้ เริ่มจากขดลวดที่ 1, 1 และ 2, 2, 2 และ 3, 3, 3 และ 4, 4, 4 และ 1 และวนกลับมายังขดลวดที่ 1 แรงบิดที่ได้จากการกระตุ้นแบบนี้จะเพิ่มมากขึ้นอีก เพราะช่วงสเต็ปมีระยะสั้นลง แต่ละสเต็ปเกิดแรงดึงจากขดลวด 2 ขดที่ถูกกระตุ้นพร้อมกัน ความถูกต้องของตำแหน่งมีเพิ่มมากขึ้น แต่ต้องพึงระวังไว้อีกประการหนึ่งว่าเมื่อกระตุ้นให้ทำงานในรูปแบบนี้จะต้องใช้การหมุนถึง 2 สเต็ปจึงจะได้เท่ากับ 1 สเต็ปเต็มของการควบคุมใน 2 แบบแรก สำหรับแหล่งจ่ายกำลังไฟฟ้าต้องใช้ขนาดเท่ากับแบบที่ 2 เป็นอย่างน้อยจึงจะเพียงพอ ขั้นตอนการทำงานต่างๆ แสดงในตารางที่ 2.5

ตารางที่ 2.5 แสดงรูปแบบการขับสเต็ปปิ้งมอเตอร์แบบฮาล์ฟสเต็ป

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

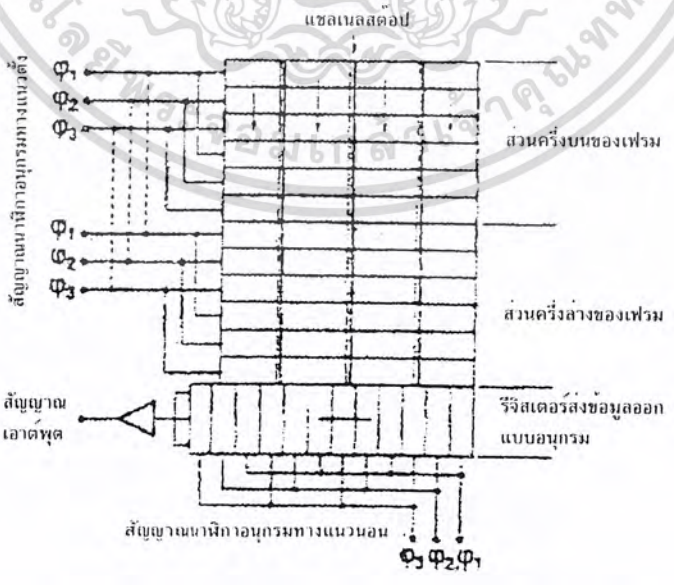
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 กล้อง CCD (Charge – Coupled Device)

ทำหน้าที่เป็นตัวทำการเปลี่ยน พลังงานแสงเป็นพลังงานไฟฟ้า เพื่อนำไปสู่การทำอิเมจโปรเซสซิ่ง ซึ่งถ้าพิจารณาดูแล้วอุปกรณ์จำพวกนี้มีอยู่มากมายหลายชนิดด้วยกัน แต่ CCD มีคุณสมบัติที่ไม่ธรรมดารวมถึงการทำงานก็ไม่ธรรมดาไม่มีการใช้ปืนอิเล็กตรอนเป็นส่วนประกอบ แต่กลับใช้การเปลี่ยนเฟสควบคุมการทำงานแทน

อุปกรณ์ CCD ถือกำเนิดขึ้นเมื่อประมาณ 20 ปีที่แล้วซึ่งขณะนั้นยังอยู่ในรูปแบบอุปกรณ์หน่วยความจำแรม (random access memory) แต่ CCD มีประโยชน์นำไปใช้งานมากกว่านั้น ซึ่งเป็นได้ทั้งดีเลย์ไลน์ ส่วนประมวลผลของสัญญาณ และที่สำคัญที่สุดเป็นอุปกรณ์ตรวจจับแสงคุณภาพสูง CCD มีหลักการทำงานต่อไปนี้

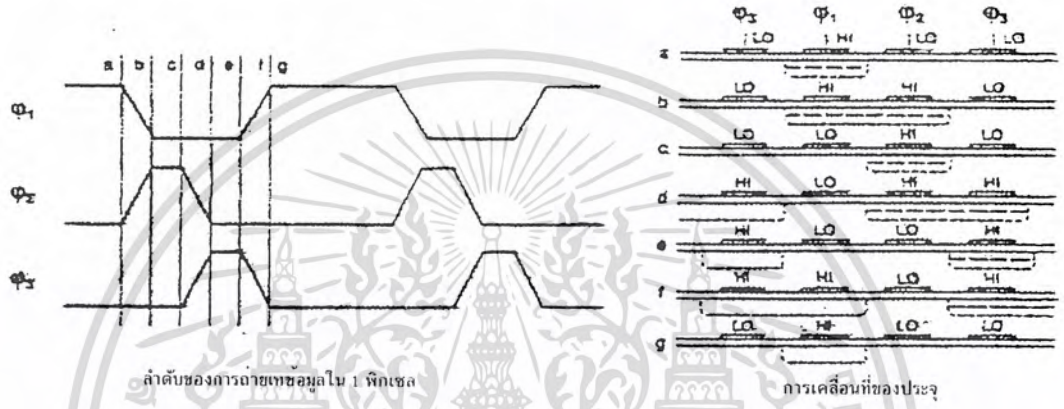
จากรูปที่ 2.14 แสดงให้เห็นถึงโครงสร้างทั่วไป และการคัปปลิ้งสัญญาณของอุปกรณ์จะเห็นได้ทันทีว่ามีการแบ่งพื้นที่ออกเป็นส่วนเล็ก ๆ มากมาย ซึ่งแต่ละชิ้นส่วนนั้นเรียกว่า พิกเซล (pixel) และถ้าสังเกตดูจากโครงสร้างแล้วจะเห็นได้ว่า CCD ไม่มีการแบ่งแต่ละพิกเซลออกเป็นตำแหน่งแอดเดรสเหมือนหน่วยความจำชนิดอื่น แต่มีการแบ่งออกเป็นแถว (row) และหลัก (column) โดยมีส่วนขั้นกลางที่เรียกว่า แชลแนลสต๊อป (channel stop) กั้นระหว่างแต่ละหลักอยู่ภายในแต่ละพิกเซลจะมีประจุอยู่ตัวหนึ่งที่ทำให้เกิดภาพ และจะมีการถ่ายเทเมื่อมีแสงตกกระทบ การถ่ายเทก็ย่อมมีตัวพาหรือยึดเหนี่ยวซึ่ง CCD ใช้หลักการหมุนเฟส (phrase clock voltage) มาใช้แต่ละแถวจะมีการกำหนดสถานะทางไฟฟ้าด้วยการหมุนเฟส 3 เฟส ที่มีความสัมพันธ์กันของช่วงเวลาแต่ละแถวจะมี



รูปที่ 2.14 แสดงโครงสร้างทั่วไปของกล้อง CCD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกำหนดสถานะทางไฟฟ้าด้วยการหมุนเฟส 3 เฟส ที่มีความสัมพันธ์กันของช่วงเวลาและแรงดัน และด้วยการทำงานเช่นนี้ประจุแต่ละพิกเซลจึงมีการถ่ายเทจากจุดหนึ่งไปอีกจุดหนึ่งได้ เมื่อประจุเคลื่อนมาจนถึงสุดขอบของแถวก็มีการส่งผ่านให้กับรีจิสเตอร์รับข้อมูลต่อไป เพื่อนำส่งผ่านข้อมูลสู่ภายนอก โดยการทยอยข้อมูลแบบอนุกรมของรีจิสเตอร์ และถูกควบคุมด้วยการหมุนเฟสเช่นเดียวกัน ดังนั้นผลรวมของการหมุนเฟสทั้งด้านแนวนอนและแนวตั้ง จึงสามารถนำประจุออกมาสู่ภายนอกได้เพื่อใช้ในการประมวลผลและสร้างภาพต่อไป



รูปที่ 2.15 แสดงถึงลักษณะรูปคลื่นของแต่ละเฟสที่มีความสัมพันธ์กัน และลักษณะการเคลื่อนที่ของประจุเมื่อมีการถ่ายเท

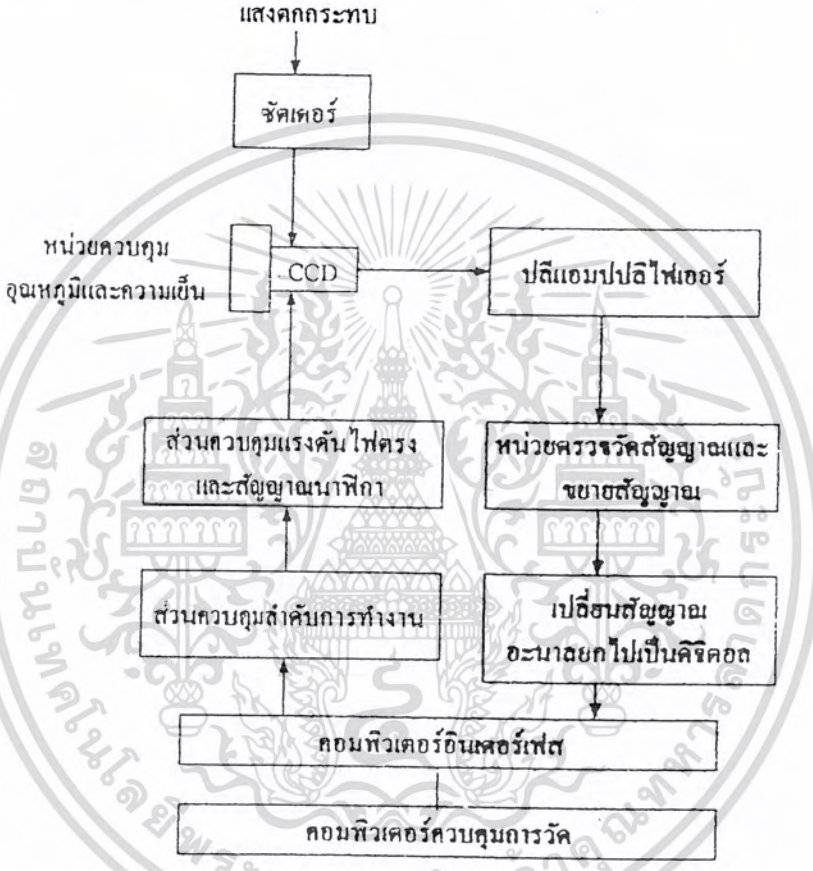
สำหรับรายละเอียดของภาพและความไวต่อแสงสว่างน้อย ๆ ขึ้นอยู่กับจำนวนพิกเซลบนตัว CCD ซึ่งมีให้เลือกหลายขนาดตามความต้องการตั้งแต่ 385 x 578, 1500 x 1500, 2048x2048 พิกเซล เป็นต้น และราคาของแต่ละขนาดต่างกันยิ่งมีจำนวนพิกเซลมากเท่าใด ประสิทธิภาพยิ่งสูงเท่านั้น และที่สำคัญราคาก็ยิ่งสูงตามไปด้วย

รูปที่ 2.16 แสดงลักษณะและขนาดของอุปกรณ์ CCD ในขนาดต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1 การนำภาพ CCD ไปใช้งานถ่ายภาพ

โดยปกติอุปกรณ์ CCD สามารถทำงานได้ด้วยอุณหภูมิห้องปกติ (room temperature) สำหรับใช้งานถ่ายภาพที่มีแสงพอประมาณ เช่น การถ่ายภาพด้วยกล้องโทรทัศน์ กล้องถ่ายรูป และกล้องวิดีโอ ซึ่งก็ทำงานได้เป็นอย่างดี แต่โดยทั่วไปอุปกรณ์อิเล็กทรอนิกส์เกือบทุกชนิดโดยเฉพาะจำพวกสารกึ่งตัวนำ มักมีการรั่วไหลของกระแสส่วนน้อย ๆ เกิดขึ้น CCD ก็มีเช่นเดียวกัน



รูปที่ 2.17 แสดงส่วนประกอบหลักของระบบกล้อง CCD

การถ่ายภาพที่มีแสงสว่างน้อย ๆ นั้น จึงจำเป็นต้องคำนึงถึงกระแสรั่วไหลส่วนนี้เพื่อให้ CCD สามารถเก็บแสงได้ดีที่สุด วิธีแก้ไขก็คือ พยายามลดอุณหภูมิให้ต่ำมาก ๆ ประมาณ 120 องศาเซลเซียส โดยใช้ไนโตรเจนเหลวให้ความเย็นอย่างรุนแรงกับ CCD กระแสรั่วไหลจะลดการสูญเสียเหลือเพียง 1 อิเล็กตรอน/ชั่วโมง/พิกเซล ในขณะที่มีความจุอิเล็กตรอนสูงถึง 100,000 อิเล็กตรอน/พิกเซล นอกจากการแก้ปัญหาเนื่องจากกระแสรั่วไหลแล้วยังต้องมีส่วนอื่นอีกมากมายประกอบอยู่ด้วยดังรูปที่ 2.17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบชัตเตอร์ที่ใช้ทำงานด้วยเครื่องกลไฟฟ้าควบคุมโดยคอมพิวเตอร์ ซึ่งควบคุมการตั้งเวลาเปิดหน้ากล้อง ความไวในการถ่ายภาพเป็นอีกสิ่งหนึ่งที่สำคัญ ดังนั้นการตรวจสอบวัดแสง จึงจำเป็นต้องใช้อุปกรณ์ที่มีสัญญาณรบกวนต่ำ และมีระบบป้องกันที่ดี โดยเฉพาะในส่วน ของวงจรภายนอก ซึ่งต้องมีการออกแบบวงจร ให้ทำงานได้อย่างมีประสิทธิภาพและที่สำคัญต้องทำให้เกิดสัญญาณรบกวนต่ำสุด อาทิวงจรปรีแอมปลิไฟล์และวงจรแซมปลิ้ง

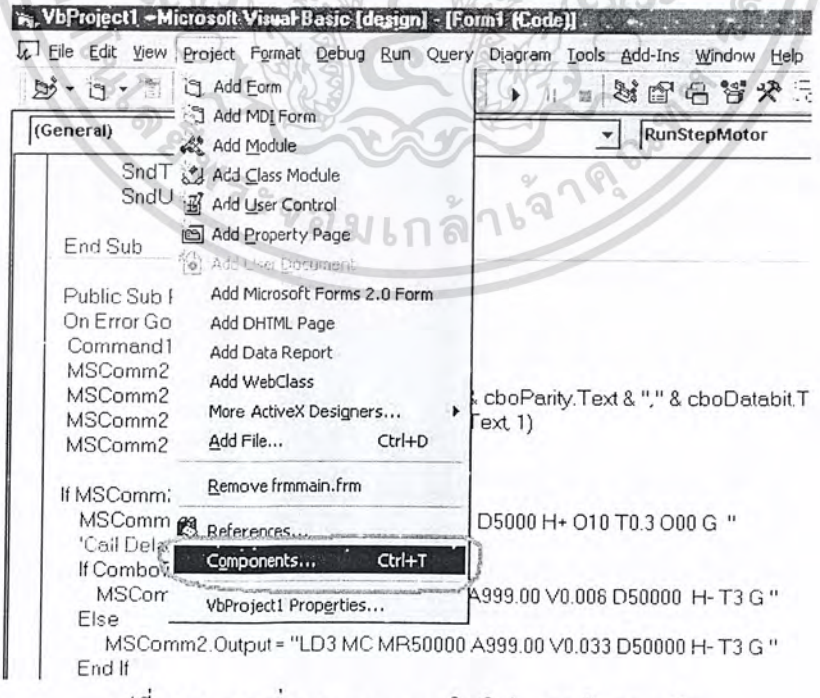
จากรูปที่ 2.17 เห็นได้ว่านอกจากวงจรที่กล่าวมาแล้ว ยังมีส่วนอื่นที่ร่วมอยู่ในกระบวนการอีกหลายส่วน เช่น ส่วนของวงจรขยายและวัดสัญญาณ วงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอลเพื่ออินเตอร์เฟสกับเครื่องคอมพิวเตอร์และวงจรกำเนิดแรงดันไบแอสไฟตรงและสัญญาณนาฬิกา เพื่อควบคุมการทำงานของ CCD ซึ่งก็ถูกควบคุมด้วยคอมพิวเตอร์ทั้งหมด

2.5 การติดต่อกับพอร์ตอนุกรมด้วยโปรแกรมวิชวลเบสิก

2.5.1 การเพิ่ม Component ที่ใช้ติดต่อกับพอร์ตอนุกรมด้วยโปรแกรมวิชวลเบสิก

สามารถทำได้โดยใช้ VB Control ที่ชื่อว่า MSComm โดยที่ต้องกำหนด Custom Control เข้าไปที่ เมนู Project--->Components แล้วเลือกที่ช่อง MSComm ก็จะปรากฏ เป็นรูปไอคอนโทรศัพท์สีเหลือง ให้คลิกที่ไอคอนลากนำมาไว้บน Form ใน Project ของโปรแกรม โดยสามารถทำตามวิธีที่กล่าวมา ได้ดังรูปต่อไปนี้

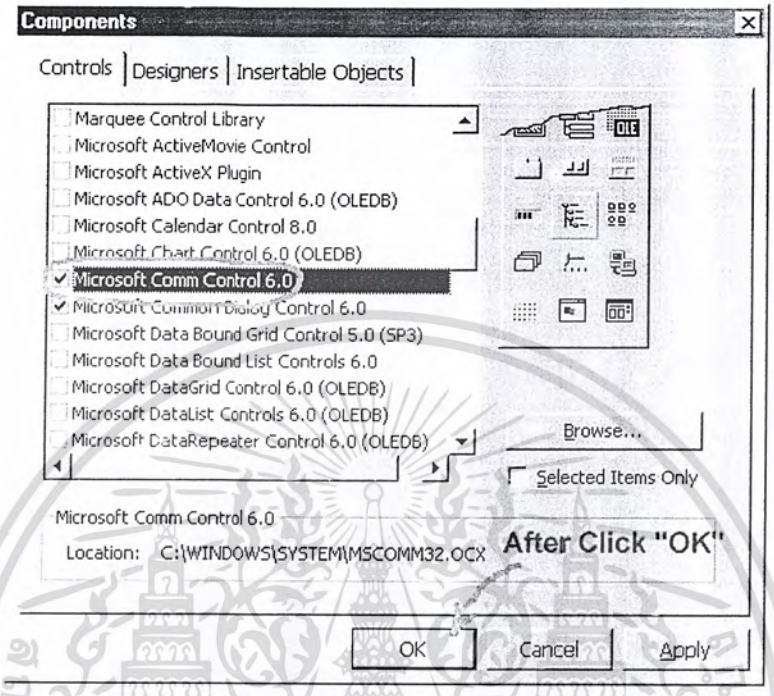
1. เลือกที่เมนูบาร์ด้านบนของโปรแกรมวิชวลเบสิก ดังรูปที่ 2.18



รูปที่ 2.18 การเพิ่ม Component ในโปแกรมวิชวลเบสิก

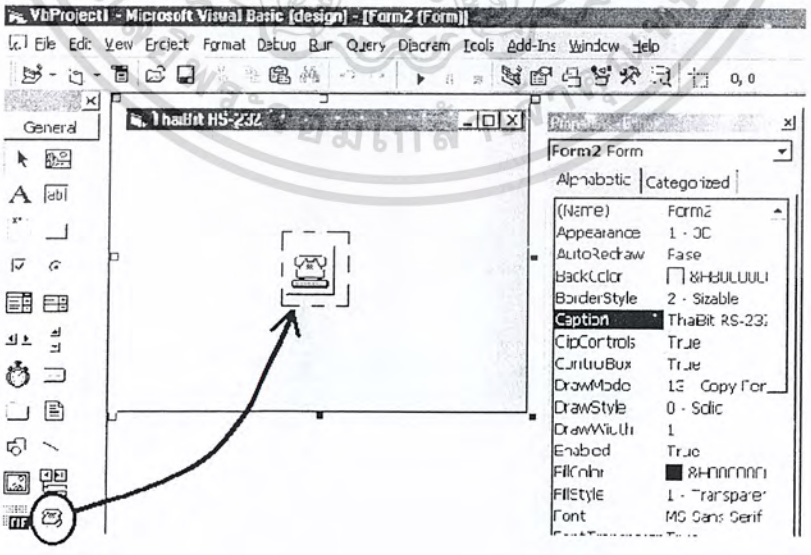
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เลือกชื่อ Control ชื่อ Microsoft Comm Control 6 ดังรูปที่ 2.19



รูปที่ 2.19 การเลือก Component ที่ใช้ติดต่อกับพอร์ตอนุกรม

3. ลาก Control ชื่อ Microsoft Comm จาก ToolBox มาไว้บน Form ดังรูปที่ 2.20



รูปที่ 2.20 การใส่ Control ที่ใช้ติดต่อกับพอร์ตอนุกรมลงในฟอร์ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2 การเขียนโปรแกรมติดต่อกับพอร์ตอนุกรมด้วยโปแกรมวิชวลเบสิก

การเขียนโปรแกรมติดต่อกับพอร์ตอนุกรมสามารถทำได้ 2 วิธี คือ

1. การติดต่อแบบอินเตอร์รัปต์

ขบวนการอินเตอร์รัปต์ อุปกรณ์รอบข้างเกือบทุกชิ้นจะต้องปฏิบัติงานอยู่เพื่อส่งสัญญาณไปให้แก่ซีพียูเสมอ ถ้าอุปกรณ์นั้นพร้อมที่จะรับส่ง ที่เคยเจอจากการทำโครงงานอุปกรณ์ จะส่งเป็นรหัสแอสกี้ เราจะเขียนโปรแกรมอินเตอร์รัปต์ โดยเมื่อที่ข้อมูลเข้ามาก็จะทำให้มี CommEvent กับ OnComm Event

2. การติดต่อแบบโพลลิ่ง

ในระบบพีซี (PC) การโพลลิ่งมีบ้างที่ใช้การส่งผ่านข้อมูลระหว่าง Terminal กับหน่วยประมวลผล กรณีข้อมูลเป็นประเภทไบท์ที่ส่งจากคีย์บอร์ด โดยวิธีการนี้จะตรวจสอบ คีย์บอร์ดว่ามีข้อมูลส่งมาหรือเปล่า โดยจะตรวจสอบตลอดเวลา การทำงานกับข้อมูลที่รับเข้ามาจะตรวจสอบด้วยความเร็วที่สูงกว่าอัตราความเร็วข้อมูลที่ส่งเข้ามาทาง คีย์บอร์ด การที่หน่วยประมวลผล ส่งสัญญาณออกไปตรวจสอบพบว่ามีข้อมูลที่ต้องส่งเข้ามา เรียกว่า"Wet Poll" ซึ่งจะเสียช่วงเวลา 90 เปอร์เซ็นต์ คาบเวลาที่เสียไปนั้น เราเลี่ยงไปใช้เทคนิค การโพลแบบ "Round Robin" แทน แต่ในวิชวลเบสิกจะใช้การตรวจสอบข้อมูลที่มาจากพอร์ตอนุกรมตลอด โดยจะใช้ Control Timer เข้ามาช่วยในการเขียนโปรแกรมซึ่งสามารถตรวจสอบได้ถึงระดับ 1 มิลลิวินาที หรือจะใช้ Do....Loop ก็ได้

ในตัวคอนโทรล MSComm มี Event ที่ใช้เพียง Event เดียวเท่านั้นเอง ก็คือ OnComm Event ซึ่งจะใช้ในการติดต่อแบบอินเตอร์รัปต์ การเขียนโปรแกรมติดต่อพอร์ตอนุกรมแบบธรรมดาจะใช้ comEvent เพียง comEvReceive, comEvSend ถ้าเป็นการติดต่อสื่อสารแบบ โมเด็มจะใช้หลายตัวในการตรวจสอบสัญญาณ

องค์ประกอบในการใช้ MSComm

การตั้งค่าติดต่อกับพอร์ต

- ComPort คือ เราต้องกำหนดหมายเลข Port ที่ใช้ต่อRS-232 (Com1,Com2)รายละเอียดดูในเมนูด้านซ้าย Serial Port Detail

- Setting คือ เราต้องกำหนดอัตรา Baud,Parity,Data(จำนวนบิต),Stop ตัวอย่าง 1200,n,8,1 เป็นต้น

- HandShaking คือ เราจะกำหนดได้ 4 แบบ

1.comNone

2.comXonXoff

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. comRTS

4.comTRSXonXoff

การใช้ Buffer ในการรับส่งข้อมูล

-InBuffersize คือ การกำหนด Buffer ในการรับข้อมูลเข้ามา

-OutBuffersize คือ การกำหนด Buffer ในการส่งข้อมูลออกไป

- Rthreshold คือ การกำหนดการเกิด Event-driven ในการรับข้อมูลเข้ามา

- Sthreshold คือ การกำหนดการเกิด Event-driven ในการรับข้อมูลออกไป

- Inputlen คือ จำนวนของข้อมูลที่ไปอ่านใน Buffer รับข้อมูล

- EOFEnable คือ การที่บอกว่าสิ้นสุดของไฟล์ (End of File)

ด้านฮาร์ดแวร์

- ParityReplace คือ ค่าของคาเลกเตอร์ที่จะแทนเมื่อเกิด Parity Error

- NullDiscard คือ การกำหนดให้รับหรือไม่รับ Null Character

- RTSEnableคือ ทำให้มีสัญญาณ RTS (Request To Send)

- DTSEnableคือ ทำให้มีสัญญาณ DTR(Data Terminal Ready)

การกำหนดคุณสมบัติของ MSComm Control ให้สามารถติดต่อกับพอร์ตอนุกรมได้

1. Property ชื่อ CommPort คือ เลือกคอมพอร์ตที่เราจะต่อใช้งาน

ตัวอย่าง MSComm1.CommPort=1

ในที่นี้เลือกจะใช้ Com1อยู่ที่ด้านหลังเครื่องคอมพิวเตอร์

2. Property ชื่อ Settingsคือ การตั้งค่าของการรับส่งข้อมูล ซึ่งจะต้องรู้ด้วยว่าอัตราบอด ของอุปกรณ์ที่จะติดต่อด้วยเป็นเท่าไร โดยมีรายละเอียดการใส่ต่างๆค่าดังนี้

MSComm1.Settings="Baud(อัตราการรับส่งข้อมูล).Parity(ถ้าไม่ใช้ใส่ N.จำนวนบิตข้อมูล.บิตสต๊อบ")

ตัวอย่าง MSComm1.Settings="1200,N,8,1"

3 .Property ชื่อ InputLenคือ กำหนดขนาดขณะที่มีข้อมูลเข้ามาให้ไปอ่านข้อมูลทั้งหมดที่อยู่ในบัพเฟอร์

ตัวอย่าง MSComm1.InputLen=1

4. Property ชื่อ PortOpen คือ จะเปิดให้พอร์ตใช้งานหรือไม่ ถ้าเปิด =True ถ้าปิด =False

ตัวอย่าง MSComm1.PortOpen=True

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. Property ชื่อ Rthresholdคือ ทำให้เกิดการกระตุ้นด้วย Event-driven เมื่อมีข้อมูลในบัฟเฟอร์รับข้อมูล (Comport) ทำให้เกิด CommEvent ใน OnComm Event

ตัวอย่าง MSComm1.Rthreshold =1

จากรายละเอียดที่กล่าวมา เราจะมาเขียนใน โพรซีเดอร์วิชวลเบสิก ซึ่งจะไว้ที่ Sub Form_Load() หรือจะสร้าง Sub ขึ้นใหม่ในกรณีที่จะเรียกใช้ภายหลัง

```
Private Sub Form_Load()
    MSComm1.Settings="1200,N,8,1"
    MSComm1.CommPort=1
    MSComm1.InputLen=1
    MSComm1.PortOpen=True
    MSComm1.Rthreshold =1
End Sub
```

วิธีของการรับส่งข้อมูลจาก Serial Port

จากวิธีเขียนโค้ดด้านบนเป็นการกำหนดค่าเริ่มต้นให้กับคอมพอร์ตและเปิดใช้การรับและส่งของพอร์ต RS-232 ดังนั้นก็สามารถจะรับและส่งข้อมูลทางพอร์ตได้ โดยใช้ Property ดังนี้

Output = ซึ่งจะเป็นการส่งข้อมูลไปที่พอร์ต

Input = เป็นส่วนของการรับข้อมูลจากพอร์ต แต่ในส่วนนี้จะต้องนำคำสั่งไปเขียนที่ Event Property OnComm จะอยู่ใน Sub MSComm_OnComm ซึ่ง จะอ่านข้อมูลเข้ามาจากทางพอร์ต RS-232

ตัวอย่าง

เช่นถ้าต้องการที่จะพิมพ์ข้อมูลส่งออกพร้อมๆกับขณะที่พิมพ์ไปด้วยก็เพียงไปเขียนโค้ดไว้ที่ Event KeyPress ของ Control TextBox ที่จะให้เป็นตัวส่งข้อมูลโดยเขียนดังนี้

```
Sub txtRXTX_KeyPress (KeyAscii As Integer)
    MSComm1.Output=Chr$(KeyAscii)
End Sub
```

ส่วนการใช้ Prooerty Input ต้องนำมาไว้ที่ Event OnComm ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

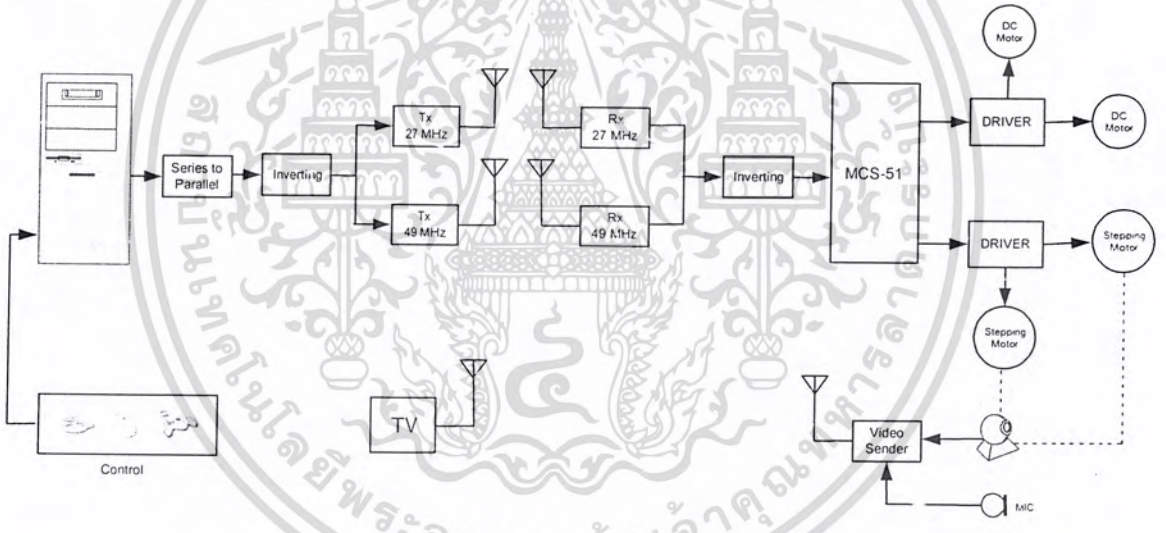
```
Private Sub MSComm1_OnComm ()
Dim StrData As Variant 'กำหนดชนิดตัวแปรเพราะต้องการให้เป็นอะไรก็ได้
    Str = MSComm1.Input
    Text1.Text = StrData
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบ และการสร้าง

3.1 วงจรเปลี่ยนสัญญาณแบบอนุกรมเป็นสัญญาณแบบขนาน

ในการควบคุมหุ่นยนต์จะควบคุมโดยคอมพิวเตอร์โดยสัญญาณควบคุมนั้นจะเป็นสัญญาณแบบอนุกรม จะส่งข้อมูลทีละ 8 บิต ด้วยอัตราเร็วเท่ากับ 9600 ในการเปลี่ยนสัญญาณแบบอนุกรมเป็นสัญญาณแบบขนานจะใช้ไมโครคอนโทรลเลอร์ตระกูล MCS-51 เบอร์ 89C2051 ช่วยในการเปลี่ยนรูปแบบของสัญญาณจากสัญญาณแบบอนุกรมเป็นแบบขนาน โดยให้สัญญาณแบบอนุกรมเข้าที่พอร์ตอนุกรมของไมโครคอนโทรลเลอร์ และให้สัญญาณเอาต์พุตออกที่พอร์ต 1 ของไมโครคอนโทรลเลอร์ โดยข้อมูลควบคุมที่ทำการเปลี่ยนนั้นเป็นดังตารางที่ 3.1



รูปที่ 3.1 แสดงองค์ประกอบของบล็อกการทำงานทั้งหมด

ตารางที่ 3.1 ตารางการเปลี่ยนข้อมูลแบบอนุกรมเป็นแบบขนาน

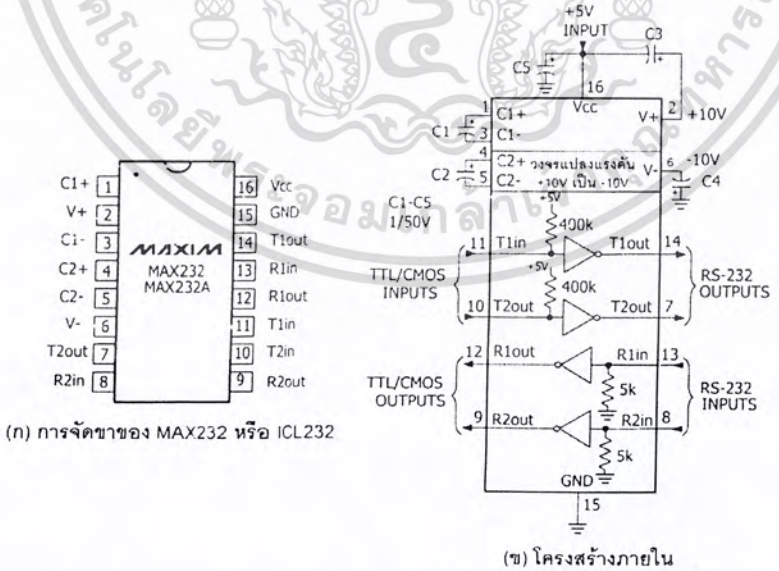
ข้อมูลแบบอนุกรม	ข้อมูลแบบขนาน
10H	01H
20H	02H
30H	04H
40H	08H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลแบบอนุกรม	ข้อมูลแบบขนาน
50H	10H
60H	20H
70H	40H
80H	80H

การใช้งานวงจรพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ตระกูล MCS-51 จะใช้มาตรฐาน RS-232 เป็นส่วนใหญ่ แต่เนื่องจากระดับสัญญาณของพอร์ตอนุกรม RS-232 มีระดับตั้งแต่ ±3 ถึง ±12 V ในขณะที่ระดับของสัญญาณไมโครคอนโทรลเลอร์ตระกูล MCS-51 อยู่ในระดับทีทีแอล ดังนั้นจึงไม่สามารถเชื่อมต่อพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ตระกูล MCS-51 เข้ากับพอร์ตอนุกรมของคอมพิวเตอร์ได้โดยตรง จึงต้องอาศัยการเชื่อมต่อผ่านไอซีพิเศษที่ทำหน้าที่ในการแปลงระดับสัญญาณ

ไอซีที่ทำหน้าที่แปลงระดับสัญญาณนี้ ต้องทำการแปลงข้อมูลของไมโครคอนโทรลเลอร์ตระกูล MCS-51 จากระดับทีทีแอลไปเป็นระดับของ RS-232 และทำการแปลงข้อมูลรับจากคอมพิวเตอร์จากระดับของ RS-232 เป็นระดับทีทีแอลเพื่อให้สามารถถ่ายทอดไปยังไมโครคอนโทรลเลอร์ตระกูล MCS-51 ได้อย่างสมบูรณ์ ไอซีดังกล่าวมีด้วยกันหลายเบอร์จากหลายผู้ผลิต อาทิ MAX232 จาก MAXIM หรือ ICL232 จาก HARRIS เป็นต้น



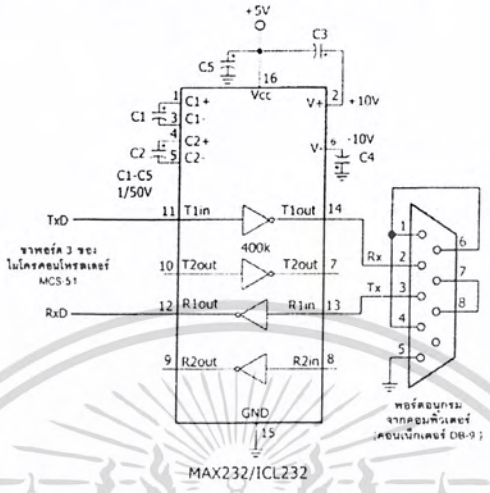
(ก) การจัดขาของ MAX232 หรือ ICL232

(ข) โครงสร้างภายใน
ของ MAX232 หรือ ICL232

รูป 3.2 รายละเอียดเบื้องต้นของไอซีแปลงสัญญาณเพื่อเชื่อมต่อกับพอร์ตอนุกรมของคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 3.2 แสดงการจัดขาของไอซี MAX232 ซึ่งใช้ในการแปลงสัญญาณ RS-232 ส่วนวงจรของการต่อกับไมโครคอนโทรลเลอร์ตระกูล MCS-51 แสดงในรูปที่ 3.3



รูปที่ 3.3 วงจรเชื่อมต่อ MAX232 เข้ากับพอร์ตอนุกรมของ คอมพิวเตอร์ และไมโครคอนโทรลเลอร์ตระกูล MCS-51

3.2 ไมโครคอนโทรลเลอร์

ในการควบคุมการสั่งงานบนตัวหุ่นยนต์โดยใช้ไมโครคอนโทรลเลอร์เบอร์ AT89C52 ความถี่สัญญาณนาฬิกา 11.0592 MHz การต่อพอร์ตใช้งานเป็นดังนี้

พอร์ต 1	P1.0, P1.1	ควบคุมการหมุนของดีซีมอเตอร์ตัวที่ 1
	P1.2, P1.3	ควบคุมการหมุนของดีซีมอเตอร์ตัวที่ 2
	P1.4, P1.7	ตรวจสอบการเคลื่อนที่ของกล้อง
พอร์ต 0	P0.0-P0.3	ควบคุมสเต็ปปิ้งมอเตอร์ที่ใช้ควบคุมกล้อง (ซ้าย-ขวา)
พอร์ต 2	P2.0-P2.3	ควบคุมสเต็ปปิ้งมอเตอร์ที่ใช้ควบคุมกล้อง (ขึ้น-ลง)
พอร์ต 3	P3.0-P3.7	ตรวจสอบสัญญาณอินพุต

การทำงานหลักจะใช้การตรวจสัญญาณอินพุตที่ได้รับจากเครื่องรับสัญญาณควบคุมจากคอมพิวเตอร์แล้วนำข้อมูลมาตรวจสอบว่าตรงกับเงื่อนไขการทำงานใด ไมโครคอนโทรลเลอร์ก็จะทำการตามคำสั่งนั้น ๆ

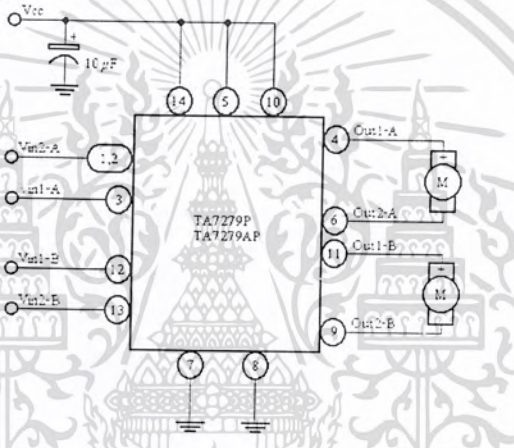
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 วงจรขับดีซีมอเตอร์

การเคลื่อนที่ของหุ่นยนต์จะใช้ดีซีมอเตอร์ โดยการควบคุมดีซีมอเตอร์จะใช้ไอซีเบอร์ TA7279P ภายใน IC จะประกอบไปด้วยวงจรไฮบริดจ์สวิตซ์ 2 ชุด สามารถควบคุมดีซีมอเตอร์ได้ 2 ตัว และแต่ละชุดมีคุณสมบัติดังต่อไปนี้

- รับแรงดันได้ตั้งแต่ 0 -18 V
- เอาท์พุตจ่ายกระแสได้ถึง 1.0A (AVE), 3.0A (Peak)

นอกจากนั้นภายใน ยังมีฟังก์ชั่นหยุดการทำงานเมื่ออุณหภูมิสูงเกินไป และยังมีวงจรจำกัดกระแสอีก โดยมีวงจรใช้งานดังรูปที่ 3.4



รูปที่ 3.4 วงจรใช้งานของไอซี เบอร์ TA7279P

จากรูปที่ 3.3 เห็นว่านอกจากไอซี แล้ว จะต่อตัวเก็บประจุ ค่าประมาณ 10 μF อีกเพียง 1 ตัว มีเอาไว้เพื่อทำหน้าที่ บายพาสสัญญาณรบกวน หากไม่ใส่เอาไว้ อาจส่งผลให้การทำงานผิดพลาด เมื่อต่อเชื่อมร่วมกับวงจรไมโครคอนโทรลเลอร์ การต่อไอซี TA7279P กับไมโครคอนโทรลเลอร์ ที่ ขา P1.0 ต่อกับขา 1, ขา P1.1 ต่อกับขา 3, ขา P1.2 ต่อกับขา 13, ขา P1.3 ต่อกับขา 12

เมื่อต่อใช้งานขาต่าง ๆ ตามวงจรดังรูปที่ 3.3 การควบคุมนั้นสามารถทำได้โดย การป้อนสัญญาณลอจิก "1" หรือสัญญาณลอจิก "0" เข้าที่ด้านอินพุต 1, 2 เท่านั้น วงจรก็สามารถทำงานได้ ส่วนมอเตอร์จะหมุน Forward หรือ Reward หรือหยุดหมุน ขึ้นอยู่กับสัญญาณควบคุมที่ป้อนให้แก่ตัวไอซี ดังแสดงรายละเอียดดังตารางที่ 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.2 ตารางแสดงการควบคุมดีซีมอเตอร์ด้วยไอซี TA7279P

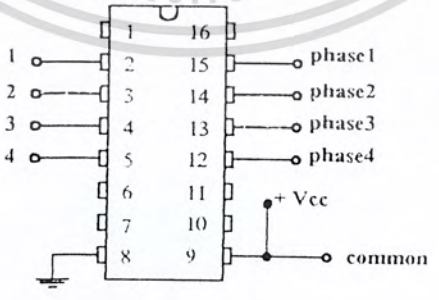
Input		Output		Mode
1	2	1	2	
0	0	High- Impedance		Stop
0	1	L	H	CW/CCW
1	0	H	L	CCW/CW
1	1	H	H	Break

จากตารางที่ 3.2 จะเห็นว่ามีการทำงานที่น่าสนใจอยู่ 2 โหมด คือ Stop และ Break หลายคนอาจจะนึกว่า 2 โหมด นี้ มีการทำงานเหมือนกัน (คือมอเตอร์ไม่หมุน) แต่แท้ที่จริงแล้วมันต่างกันอยู่เล็กน้อยดังนี้

- Stop จะทำงานเปรียบเสมือน การปลดมอเตอร์ออกจากแหล่งจ่าย
- Break จะทำงานเปรียบเสมือน การล็อกมอเตอร์เอาไว้

3.4 วงจรขับสเต็ปปิ้งมอเตอร์

สเต็ปปิ้งมอเตอร์ใช้ในการควบคุมกล้องให้หมุนซ้ายหรือหมุนขวา ขึ้นหรือลง จึงต้องใช้สเต็ปปิ้งมอเตอร์ 2 ตัว การควบคุมสเต็ปปิ้งมอเตอร์จะใช้พอร์ต P0 (P0.0-P0.3) และ P2 (P2.0-P2.3) ส่งข้อมูลไปยังไอซีไดเวอร์กระแสสูงแบบคอลเล็กเตอร์เปิดเบอร์ ULN 2003 โดยมีความสามารถในการจ่ายกระแสได้สูงสุด 500 mA



รูปที่ 3.5 แสดงตำแหน่งขาของไอซีเบอร์ ULN2003 สำหรับขับสเต็ปปิ้งมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 ชุดเครื่องส่งสัญญาณวิดีโอ (Video Sender)

หลักการทำงานจากวงจรสมบูรณ์ดังรูปที่ 3.6 ทำหน้าที่ในส่วนของวงจรกำเนิดความถี่แบบ LC ชนิดกราวด์-เบสโคลฟิตส์ มี L_1 กับ C_4 ประกอบกันเป็นวงจรจูนทางเอาท์พุต สำหรับปรับความถี่คลื่นพาห์ภาพให้ได้ 450 MHz-500MHz R_1, R_2, R_3 เป็นวงจรไบอัสให้ Q_1 โดย C_1 ทำหน้าที่ตัดความถี่สูงที่ขาเบส C_2 กับ C_3 ต่อเป็นวงจรแบ่งแรงดันกำหนดอัตราส่วนของสัญญาณที่จะทำการป้อนกลับจากคอลเล็คเตอร์มายังอิมิตเตอร์

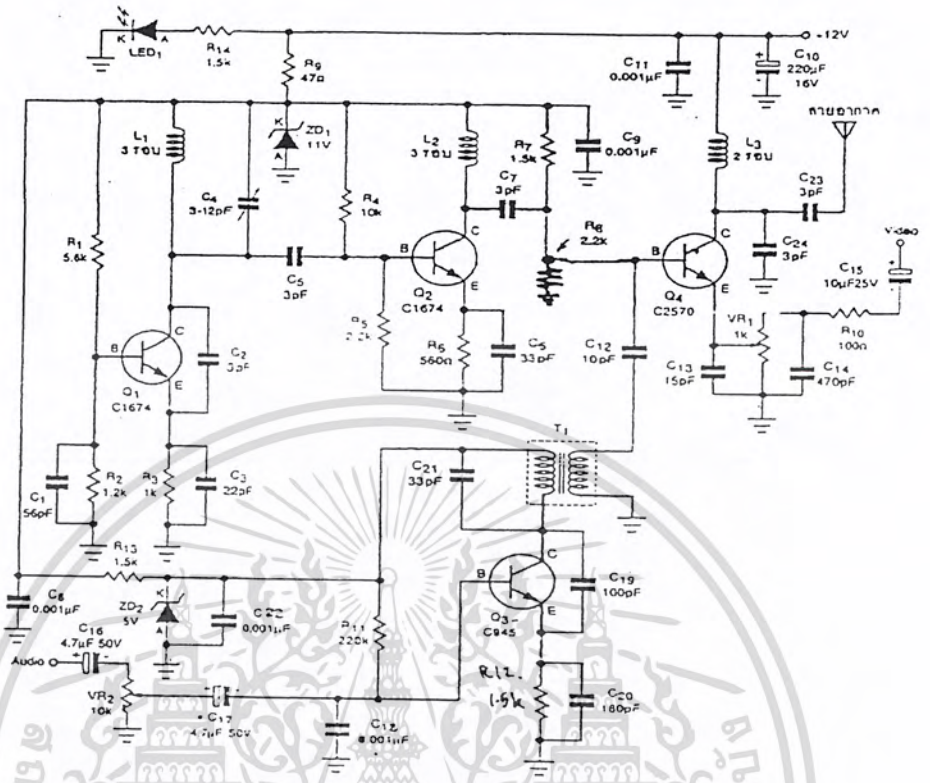
ความถี่ที่กำเนิดได้จะส่งไปขยายให้แรงขึ้นผ่าน C คัปปลิ้ง C_5 ไป Q_2 ซึ่งเป็นวงจรขยายสัญญาณ R_4, R_5, R_6 เป็นวงจรไบอัสให้ Q_2, C_6 เป็น C บายพาสความถี่ลงกราวด์เพื่อไม่ให้เกิดการสูญเสียกำลังงานที่ R_6, L_2 เป็นตัวส่งผ่านกระแสไฟตรงเลี้ยงวงจรให้ Q_2 ในขณะเดียวกันก็เป็น RF โช๊คป้องกันไม่ให้ความถี่คลื่นพาห์ภาพออกไปยังแหล่งจ่ายไฟด้วย

ใช้ซีเนอร์ไดโอดเรกูเลตแหล่งจ่ายที่ป้อนให้วงจรสองภาคแรก เพื่อให้มีเสถียรภาพทางความถี่ โดย ZD_1 มี R_9 เป็นตัวจำกัดกระแสและกรองให้เรียบขึ้นโดย C_8, C_9 สัญญาณที่ผ่านการขยายจะคัปปลิ้งผ่าน C_7 ไปยัง Q_4 ที่เป็นวงจรขยายกำลังและวงจรมอดูเลตแบบ AM โดยถูกลดทอนความแรงด้วย R_8 ให้มีความเหมาะสมกับสัญญาณเสียงที่ผ่านมาที่ขาเบสของ Q_4 L_3 ทำหน้าที่เช่นเดียวกับ L_2 ภาคนี้ใช้แหล่งจ่ายไฟหลักป้อนให้โดยตรงเพื่อให้ได้กำลังที่สูง โดยมี C_{10}, C_{11}, C_{23} เป็นตัวกรองกระแสสำหรับสัญญาณภาคที่เข้ามาทำการมอดูเลตกับคลื่นพาห์ภาพพร้อมกับสัญญาณเสียงแบบ AM จะถูกคัปปลิ้งและปรับความแรงของสัญญาณให้เหมาะสมด้วย C_{14}, C_{15}, C_{13}, R_{10} โดยมี VR_1 เป็นตัวปรับเปอร์เซ็นต์การมอดูเลต ส่วนสัญญาณเสียงจะคัปปลิ้งผ่าน C_{14}, C_{17} ไปยัง Q_3 มี VR_2 เป็นตัวปรับช่วงความถี่เบี่ยงเบน

R_{11}, R_{12} เป็น R ไปอัส C_{18} ทำหน้าที่ตัวความถี่สูงที่ขาเบส C_{19}, C_{20} เป็น C แบ่งแรงดันเช่นเดียวกับ C_2, C_3 ต่อเป็นวงจรกำเนิดความถี่แบบ LC กราวด์เบส คอลฟิตส์ เช่นเดียวกับ Q_1 แต่จัดวงจรเป็นแบบ VCO (Voltage Control Oscillator) เพื่อทำการมอดูเลตสัญญาณเสียงที่เข้ามาทางอินพุตแบบ FM โดยมีความถี่กลางซึ่งเป็นผลต่างระหว่างความถี่คลื่นพาห์สัญญาณภาพกับเสียงโดยกำเนิดจากค่าของ T_1 กับ C_{21}

สัญญาณเสียงที่ผ่านการมอดูเลตแบบ FM แล้วจะถูกคัปปลิ้งผ่าน T_1, C_{12} ไปยังขาเบสของ Q_4 แหล่งจ่ายที่ป้อนให้กับวงจรนี้จะถูกเรกูเลตให้เป็น 5V โดย ZD_2 มี R_{13} เป็นตัวจำกัดกระแสและกรองให้เรียบไม่ให้มีสัญญาณความถี่สูงปะปนมาด้วย C_{22} สัญญาณโทรทัศน์ที่สมบูรณ์จากเอาท์พุต Q_4 จะส่งแพร่กระจายออกจากอากาศทางสายอากาศโดยมี C_{24}, C_{25} ทำหน้าที่เป็นวงจร แมชชิ่ง (Matching) สำหรับสถานการณ์ทำงานจะแสดงด้วย LED_1 โดยมีตัวต้านทาน R_{14} เป็นตัวจำกัดกระแส

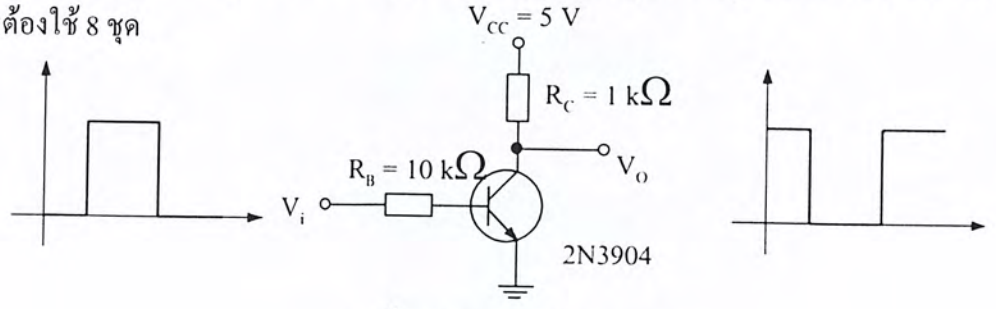
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 แสดงวงจรเครื่องส่งสัญญาณภาพ (Video Sender)

3.6 วงจร Inverting

วงจร Inverting ที่ใช้มี 2 ส่วนคือ ส่วนที่ต่อกับวงจรเปลี่ยนสัญญาณแบบอนุกรมเป็นแบบขนาน และส่วนที่ต่อกับเครื่องรับ เนื่องจากเครื่องส่งจะส่งสัญญาณควบคุมเฉพาะลอจิกศูนย์ แต่สัญญาณที่ถูกเปลี่ยนเป็นแบบขนานจะเป็นลอจิกหนึ่ง และที่พอร์ต 3 ของไมโครคอนโทรลเลอร์ทำหน้าที่เป็นพอร์ตอินพุต มิฉะนั้นไมโครคอนโทรลเลอร์จะรับสัญญาณลอจิกศูนย์ได้ง่ายแต่สัญญาณที่เครื่องรับข้อมูลควบคุมนั้นเป็นลอจิกหนึ่ง ดังนั้นเราจึงต้องทำการ Invert ข้อมูลที่ส่งออกมาด้วยการใช้วงจร Inverting ดังรูปที่ 3.7 เมื่อสัญญาณอินพุตเป็นลอจิกศูนย์ สัญญาณทางเอาต์พุตจะเป็นหนึ่ง เมื่อสัญญาณอินพุตเป็นหนึ่ง สัญญาณเอาต์พุตจะเป็นศูนย์ ข้อมูลมีทั้งหมด 8 บิต หนึ่งวงจรจะต้องใช้ 8 ชุด

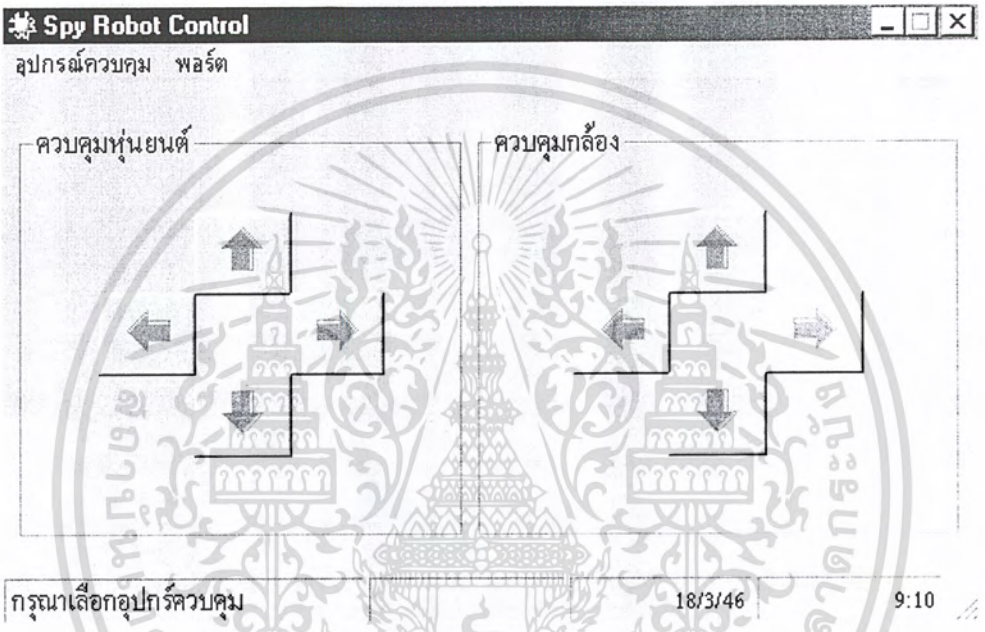


รูปที่ 3.7 แสดงวงจร Inverting

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 โปรแกรมควบคุม

โปรแกรมที่ใช้ในการควบคุมหุ่นยนต์ สร้างขึ้นโดยใช้โปรมแกรมวิชวลเบสิก สามารถควบคุมได้ทั้งเมาส์ (Mouse) และจอยสติ๊ก (Joystick) โดยที่โปแกรมจะมีเมนูให้เลือกว่าผู้ใช้จะใช้อุปกรณ์ใดควบคุม ดังรูป 3.8



รูปที่ 3.8 แสดงหน้าจอของโปรแกรม Spy Robot Control

หน้าต่างของโปรแกรมจะแบ่งการควบคุมเป็น 2 ส่วน คือ ตัวหุ่นยนต์ และกล้อง ปุ่มที่ใช้ในการควบคุมด้วยเมาส์จะสามารถใช้งานได้ก็ต่อเมื่อเลือกการควบคุมโดยใช้เมาส์เท่านั้น ถ้าต้องการควบคุมด้วยจอยสติ๊กจะต้องทำการติดตั้งจอยสติ๊กก่อนใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผลการทดลอง

4.1 การติดต่อสื่อสารระหว่างคอมพิวเตอร์กับหุ่นยนต์นักสืบ

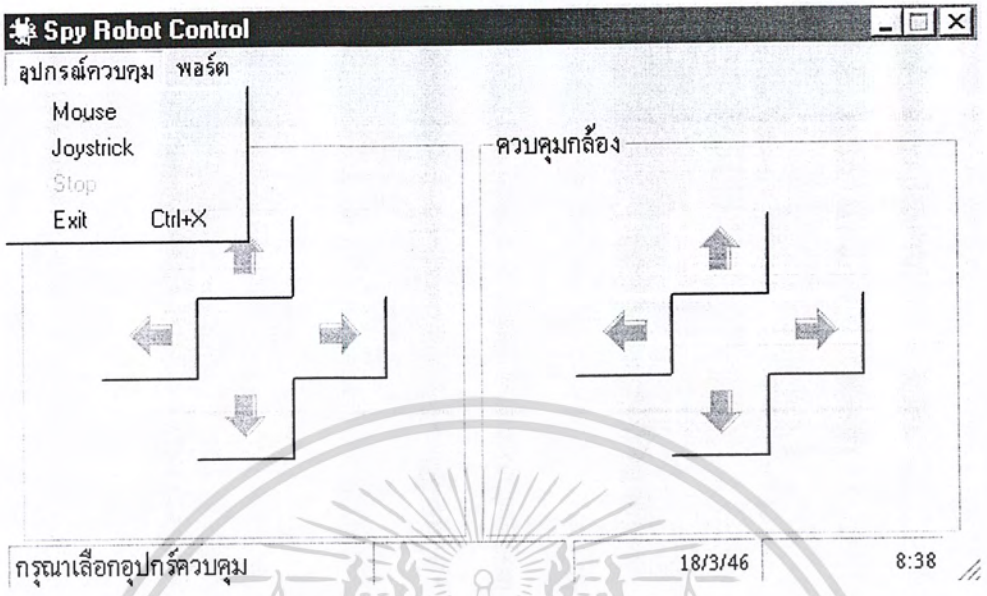
การติดต่อสื่อสารเพื่อส่งข้อมูลในการควบคุมส่วนต่างๆ นั้น ใช้การส่งแบบไร้สาย ซึ่งใช้เครื่องรับ-ส่ง 2 ชุด ความถี่ 27 MHzใช้ในการควบคุมกล้องและ 49 MHz ใช้การควบคุมการเคลื่อนที่ของหุ่นยนต์ การส่งข้อมูลใช้มาตรฐาน RS-232 ติดต่อกันด้วยอัตราบอด 9600 บิตต่อวินาที ในการสั่งงานใช้ข้อมูลขนาด 8 บิต ซึ่งรหัสคำสั่งแสดงไว้ในตารางที่ 4.1

ตารางที่ 4.1 แสดงค่าข้อมูล และความหมายข้อมูล

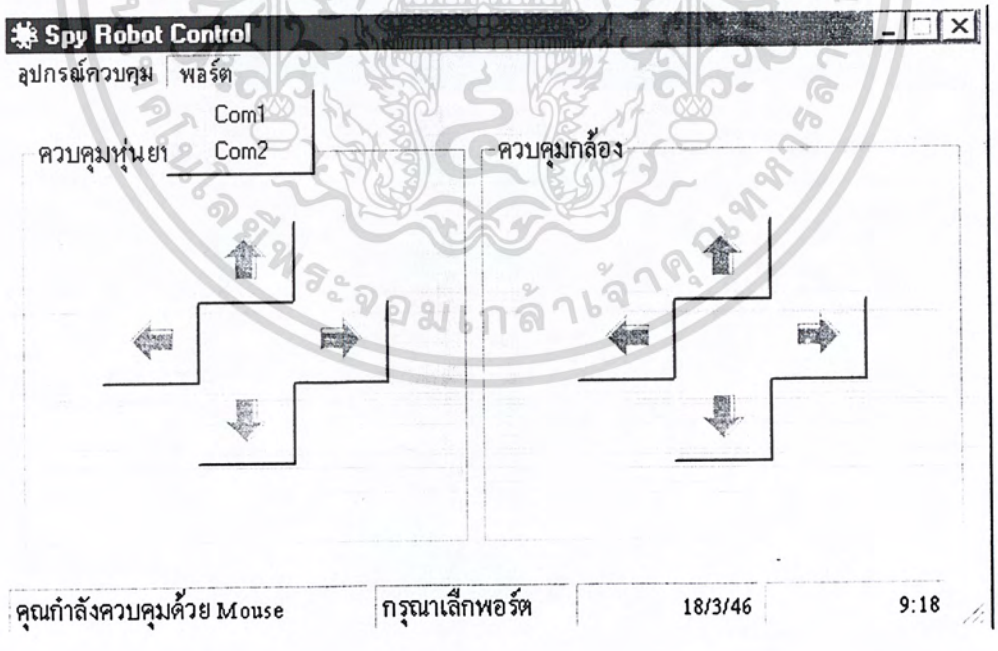
ค่าข้อมูล	ความหมายของข้อมูล
10H	ถอยหลัง
20H	เดินหน้า
30H	เลี้ยวซ้าย
40H	เลี้ยวขวา
50H	สเต็ปปิ้งมอเตอร์หมุนซ้าย
60H	สเต็ปปิ้งมอเตอร์หมุนขวา
70H	สเต็ปปิ้งมอเตอร์หมุนลง
80H	สเต็ปปิ้งมอเตอร์หมุนขึ้น

เมื่อเข้าสู่โปรแกรม Spy Robot Control ทำการเลือกโหมดในการควบคุมดังรูปที่ 4.1 แล้วทำการเลือกพอร์ตที่ต้องการดังรูปที่4.2 ก็จะสามารถควบคุมการเคลื่อนที่ได้โดยกดปุ่มที่มีสัญลักษณ์เดินหน้า ถอยหลัง เลี้ยวซ้าย เลี้ยวขวา หมุนกล้องทางซ้าย หมุนกล้องไปทางขวา เป็นต้น บนหน้าจอเมื่อสั่งให้มีการทำงานใด จะสังเกตที่แทบ status bar จะมีข้อความแสดงการทำงาน หรือถ้าการควบคุมใช้จอยสติ๊กก็จะมีข้อความแสดงการทำงานว่าขณะนี้หุ่นยนต์กำลังทำอะไร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 แสดงวิธีการเลือกโหมดการควบคุม

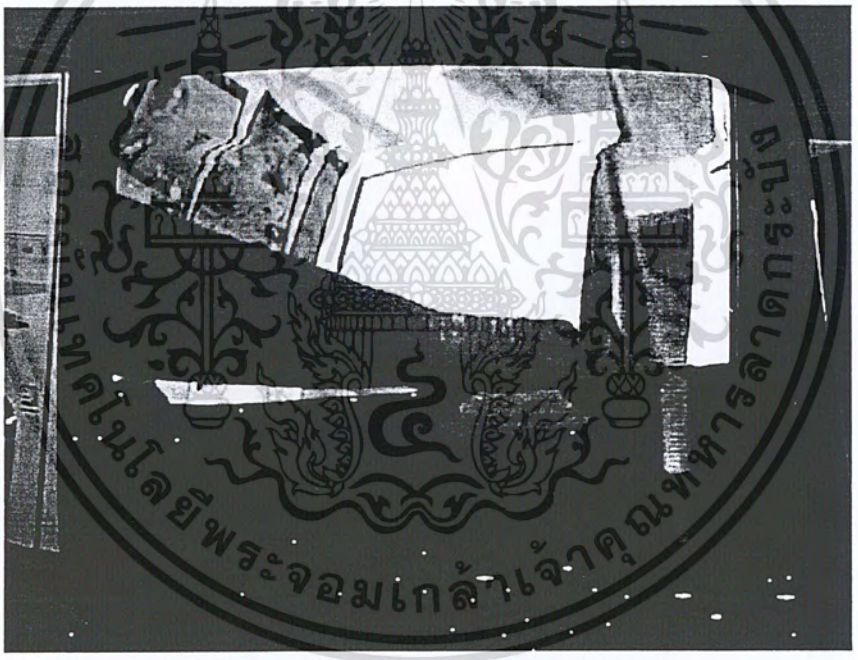


รูปที่ 4.2 แสดงวิธีการเลือกพอร์ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

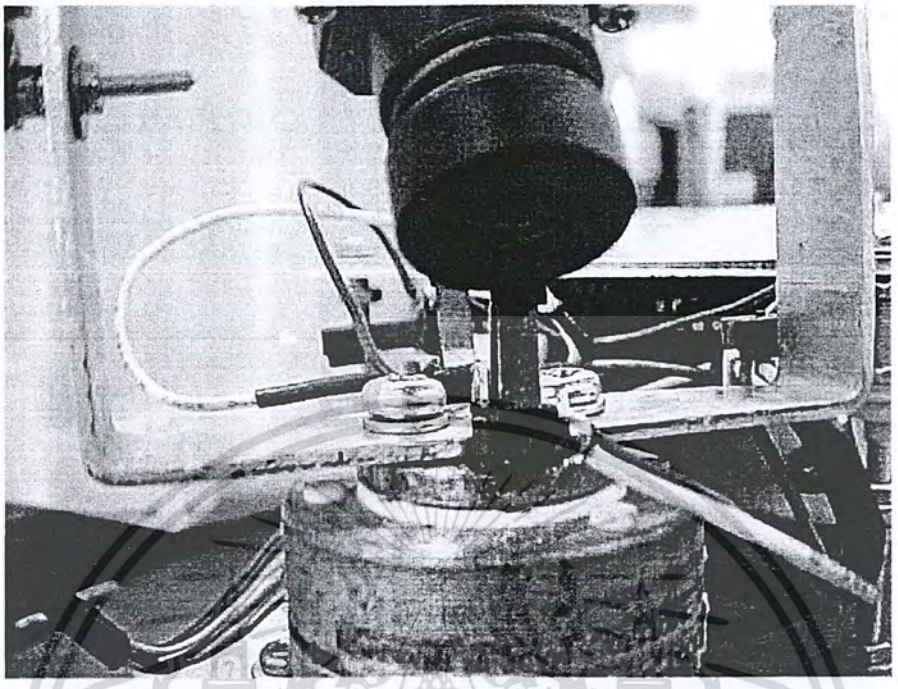
4.2 การแสดงภาพและเสียง

ส่วนของภาพและเสียงจะแสดงที่โทรทัศน์ดังรูปที่ 4.3 เป็นภาพที่มาจากเครื่องส่งสัญญาณภาพบนตัวหุ่นยนต์นักสืบ ซึ่งภาพนี้จะเป็นส่วนช่วยในการมองเห็น ในการมองเห็นจะสามารถควบคุมการมองเห็นได้ประมาณ 120 องศาในแนวระนาบ โดยมีไมโครสวิตซ์ซ้ายขวาเป็นการตรวจสอบว่าหมุนสุดหรือยัง เมื่อชนกับไมโครสวิตช์ก็จะไม่สามารถหมุนต่อได้ และในการหมุนขึ้นลงก็สามารถหมุนได้ประมาณ 300 องศา ซึ่งเราสามารถแสดงภาพได้ทั้งด้านหน้าและด้านหลังโดยมีไมโครสวิตซ์เป็นตัวตรวจสอบเหมือนกันดังรูปที่ 4.4

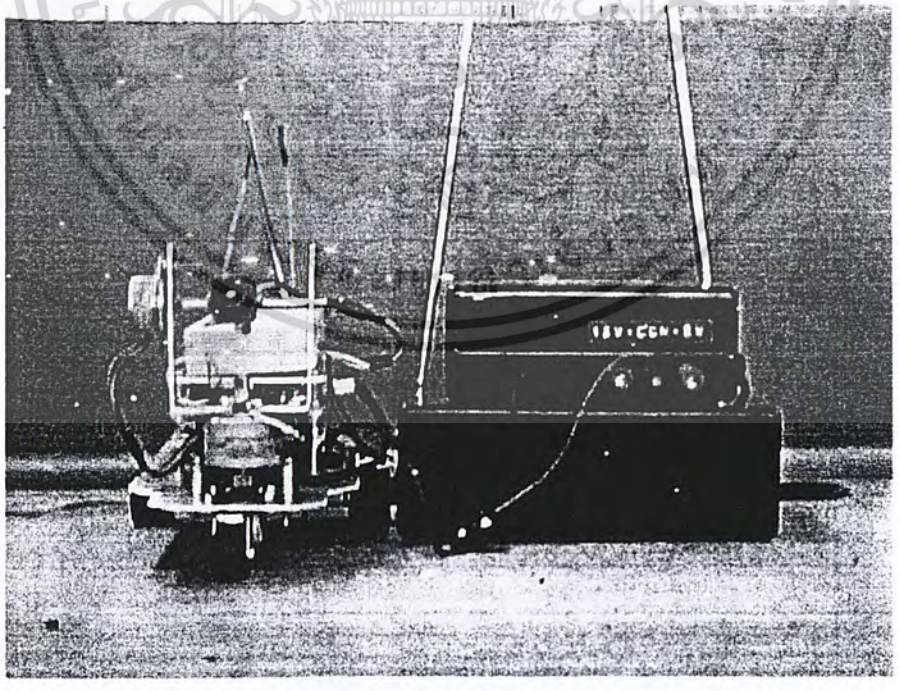


รูปที่ 4.3 แสดงภาพจากหุ่นยนต์นักสืบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 แสดงการตรวจสอบการหมุนของกล้อง



รูปที่ 4.5 แสดงภาพหุ่นยนต์นักสืบและเครื่องส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผล ปัญหาและข้อเสนอแนะ

5.1 สรุปผล

จากการทดสอบการควบคุมให้หุ่นยนต์นักสืบทำตามคำสั่งต่างๆ ซึ่งได้แก่ เดินหน้า ถอยหลัง เลี้ยวซ้าย เลี้ยวขวา การควบคุมกล้อง หุ่นยนต์นักสืบสามารถปฏิบัติตามคำสั่งได้อย่างถูกต้อง แต่มีข้อจำกัดในเรื่องของระยะในการควบคุม ถ้าหุ่นยนต์นักสืบอยู่นอกรัศมีของการทำงานของเครื่องส่งก็จะไม่สามารถควบคุมได้

5.2 ปัญหา

1. ระยะการควบคุมของหุ่นยนต์นักสืบมีข้อจำกัด
2. ระยะการส่งสัญญาณภาพและเสียงมีระยะอยู่ประมาณ 8 – 12 เมตร และมีสัญญาณรบกวนสัญญาณอันเนื่องจากความถี่รอบข้าง
3. การแสดงภาพยังต้องใช้โทรทัศน์ทำให้ส่วนประกอบจำนวนมากขึ้น
4. แบตเตอรี่สามารถจ่ายกระแสไฟฟ้าในระยะเวลาที่สั้นเนื่องจากมีจำนวนโหลดมาก และตัวแบตเตอรี่ก็มีข้อจำกัดในเรื่องแรงดันและกระแส แต่จำเป็นต้องใช้เพราะมีขนาดเล็ก

5.3 ข้อเสนอแนะ

1. ระยะในการทำงานของเครื่องรับ เครื่องส่งควรให้มีระยะไกลกว่านี้
2. ระยะในการส่งสัญญาณภาพควรส่งใกล้ระยะไกลกว่านี้
3. การแสดงภาพและเสียงควรแสดงที่คอมพิวเตอร์โดยการเพิ่มการ์ดทีวีเข้าไปในเครื่องคอมพิวเตอร์
4. ในอนาคตควรปรับปรุงสามารถทำงานได้อัตโนมัติโดยไม่ต้องใช้มนุษย์คอยควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. ชัยวุฒิ จันมา. การเขียนโปรแกรมด้วย Visual Basic 6. กรุงเทพฯ. ซีเอ็ดยูเคชั่น. 2538
2. กฤษฎา ใจเย็น. เรียนรู้แลปฏิบัติการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกผ่านพอร์ตอนุกรม. กรุงเทพฯ. Innovative experiment. 2535
3. สิทธา ศรีวิริยาจารย์. Microsoft Visual Basic Profession 6.0 step by step. กรุงเทพฯ. ซีเอ็ดยูเคชั่น. 2542
4. วารสารอิเล็กทรอนิกส์สมัครเล่น. ฉบับที่44 หน้า 42 -48 . กรุงทพฯ. ซีเอ็ดยูเคชั่น. 2537
5. วรพจน์ กรแก้ววัฒนกุล. เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์ MCS-51. กรุงเทพฯ. Innovative experiment. 2542

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

โปรแกรมเปลี่ยนสัญญาณแบบอนุกรมให้เป็นแบบขนาน

```
                ORG     0000H
                ACALL   INT
MAIN:           MOV     P1,#00H
                ACALL   RECEIVE
                SJMP    MAIN
INT:            MOV     IE,#00000000B
                MOV     TMOD,#00100000B
                MOV     TH1,#0FDH
                MOV     TL1,#0FDH
                MOV     SCON,#01010000B
                SETB    TR1
                RET
RECEIVE:        JNB     RI,$
                CLR     RI
                MOV     A,SBUF
                ACALL   CHECK
                RET
CHECK:          CJNE    A,#10H,CHECK1
                SETB    P1.0
                ACALL   DELAY_S
                RET
CHECK1:         CJNE    A,#20H,CHECK2
                SETB    P1.1
                ACALL   DELAY_S
                RET
CHECK2:         CJNE    A,#30H,CHECK3
                SETB    P1.2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                ACALL   DELAY_S
                RET
CHECK3:         CJNE    A,#40H,CHECK4
                SETB    P1.3
                ACALL   DELAY_S
                RET
CHECK4:         CJNE    A,#50H,CHECK5
                SETB    P1.4
                ACALL   DELAY_S
                RET
CHECK5:         CJNE    A,#60H,CHECK6
                SETB    P1.5
                ACALL   DELAY_S
                RET
CHECK6:         CJNE    A,#70H,CHECK7
                SETB    P1.6
                ACALL   DELAY_S
                RET
CHECK7:         CJNE    A,#80H,CHECK8
                SETB    P1.7
                ACALL   DELAY_S
                RET
CHECK8:         RET
DELAY_S:        MOV     R2,#40H
DELAY_1:        MOV     R3,#0FFH
                DJNZ    R3,$
                DJNZ    R2,DELAY_1
                RET
                END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมควบคุมหุ่นยนต์นักสืบ

```
MSTEP       EQU     020H
CSTEP       EQU     024H
            ORG     0000H
            MOV     P0,#00H
            MOV     P2,#00H
            MOV     P3,#0FFH
            MOV     MSTEP,#08H
            MOV     CSTEP,#08H
MAIN:       MOV     P1,#0F0H
            ACALL   SCAN
            SJMP    MAIN
SCAN:       JNB     P3.0,CMLEFT
            JNB     P3.1,CMRIGHT
            JNB     P3.2,CMUP
            JNB     P3.3,CMDOWN
            JNB     P3.4,FORWORD
            JNB     P3.5,REWORD
            JNB     P3.6,LEFT
            JNB     P3.7,RIGHT
            RET
CMLEFT:     JNB     P1.6,SCAN
            MOV     R0,CSTEP
            CJNE    R0,#80H,LEFTA
            MOV     CSTEP,#08H
LEFTA:      MOV     A,CSTEP
            MOV     P0,A
            ACALL   STEP_DELAY
            RR      A
            MOV     CSTEP,A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                RET
CMRIGHT:        JNB     P1.7,SCAN
                MOV     R0,CSTEP
                CJNE    R0,#10H,RIGHTA
                MOV     CSTEP,#01H
RIGHTA:         MOV     A,CSTEP
                MOV     P0,A
                ACALL   STEP_DELAY
                RL      A
                MOV     CSTEP,A
                RET
CMDOWN:         JNB     P1.4,SCAN
                MOV     R0,MSTEP
                CJNE    R0,#80H,DOWN
                MOV     MSTEP,#08H
DOWN:           MOV     A,MSTEP
                MOV     P2,A
                ACALL   STEP_DELAY
                RR      A
                MOV     MSTEP,A
                RET
CMUP:           JNB     P1.5,SCAN
                MOV     R0,MSTEP
                CJNE    R0,#10H,UP
                MOV     MSTEP,#01H
UP:             MOV     A,MSTEP
                MOV     P2,A
                ACALL   STEP_DELAY
                RL      A
                MOV     MSTEP,A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                RET
FORWORD:        ACALL   DELAY_S
                CLR     P1.1
                SETB    P1.0
                ACALL   DELAY_S
                RET
REWORD:         ACALL   DELAY_S
                CLR     P1.0
                SETB    P1.1
                ACALL   DELAY_S
                RET
LEFT:           ACALL   DELAY_S
                CLR     P1.3
                SETB    P1.2
                ACALL   DELAY_S
                RET
RIGHT:          ACALL   DELAY_S
                CLR     P1.2
                SETB    P1.3
                ACALL   DELAY_S
                RET
DELAY_S:        MOV     R1,#03H
DELAY_1:        MOV     R2,#0FFH
                DJNZ    R2,$
                DJNZ    R1,DELAY_1
                RET
STEP_DELAY:     MOV     R3,#0AAH
DELAY_2:        MOV     R4,#0E6H
                DJNZ    R4,$
                DJNZ    R3,DELAY_2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
RET
END
```

โปรแกรมควบคุมหุ่นยนต์นักสืบด้วยคอมพิวเตอร์

```
Const t1 = "Control Car :  "
 Const t2 = "Control Carmera  :"
 Dim num As Integer
 Dim Con_Robot As Byte
---------------------------------------------------------------------------------------
  Sub Mouseup()
 Stb.Panels.Item(1).Text = ""
 num = 0
 End Sub
---------------------------------------------------------------------------------------
Sub Connecting(b As Integer)
 MSC.Settings = "9600,N,8,1"
 MSC.CommPort = b
 MSC.InputLen = 1
 MSC.PortOpen = True
 MSC.RThreshold = 1
 End Sub
---------------------------------------------------------------------------------------
Sub Button(X As Boolean)
If X = True Then
    FrBton.Enabled = X
    ReBton.Enabled = X
    LeBton.Enabled = X
    RigBton.Enabled = X
    UpBton.Enabled = X
    DwnBton.Enabled = X
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    CLeBton.Enabled = X
    CRigBton.Enabled = X
Else
    FrBton.Enabled = X
    ReBton.Enabled = X
    LeBton.Enabled = X
    RigBton.Enabled = X
    UpBton.Enabled = X
    DwnBton.Enabled = X
    CLeBton.Enabled = X
    CRigBton.Enabled = X
End If
End Sub
---------------------------------------------------------------------------------------
Private Sub CLeBton_KeyPress(KeyAscii As Integer)
Stb.Panels.Item(1).Text = t2 & "หมุนซ้าย"
MSC.Output = Chr(&H50)
End Sub
---------------------------------------------------------------------------------------
Private Sub CLeBton_KeyUp(KeyCode As Integer, Shift As Integer)
Mouseup
End Sub
---------------------------------------------------------------------------------------
Private Sub CLeBton_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As
Single)
Stb.Panels.Item(1).Text = t2 & "หมุนซ้าย"
num = 7
End Sub
---------------------------------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub CLeBton_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
Mouseup
End Sub
-----------------------------------------------------------------------------------------------
Private Sub Com1_Click()
   Connecting (1)
   Stb.Panels.Item(2).Text = "Serial Port : Com " & MSC.CommPort
 Select Case Con_Robot
 Case 0:     Button (True)
              Mouseup
Case 1: JKJoystick1.TStart
 End Select
    com2.Enabled = False
   Com1.Enabled = False
   mStop.Enabled = True
End Sub
-----------------------------------------------------------------------------------------------
Private Sub com2_Click()
   Connecting (2)
   Stb.Panels.Item(2).Text = "Serial Port : Com " & MSC.CommPort
   Select Case Con_Robot
        Case 0:    Button (True)
                   Mouseup
        Case 1:    JKJoystick1.TStart
   End Select
   com2.Enabled = False
   Com1.Enabled = False
    mStop.Enabled = True
End Sub
-----------------------------------------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub CRigBton_KeyPress(KeyAscii As Integer)
Stb.Panels.Item(1).Text = t2 & "หมุนขวา"
MSC.Output = Chr(&H60)
End Sub
---------------------------------------------------------------------------------------------
Private Sub CRigBton_KeyUp(KeyCode As Integer, Shift As Integer)
Mouseup
End Sub
---------------------------------------------------------------------------------------------
Private Sub CRigBton_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As
Single)
Stb.Panels.Item(1).Text = t2 & "หมุนขวา"
num = 8
End Sub
---------------------------------------------------------------------------------------------
Private Sub CRigBton_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
Mouseup
End Sub
---------------------------------------------------------------------------------------------
Private Sub DwnBton_KeyPress(KeyAscii As Integer)
Stb.Panels.Item(1).Text = t2 & "หมุนลง"
MSC.Output = Chr(&H70)
End Sub
---------------------------------------------------------------------------------------------
Private Sub DwnBton_KeyUp(KeyCode As Integer, Shift As Integer)
Mouseup
End Sub
---------------------------------------------------------------------------------------------
Private Sub DwnBton_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As
Single)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Stb.Panels.Item(1).Text = t2 & "หมุนลง"
num = 6
End Sub
---------------------------------------------------------------------------------------------
Private Sub DwnBton_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
Mouseup
End Sub
---------------------------------------------------------------------------------------------
Private Sub Exit_Click()
End
End Sub
---------------------------------------------------------------------------------------------
Private Sub Form_Load()
mStop.Enabled = False
Me.Top = Screen.Height / 2 - Me.ScaleHeight / 2
Me.Left = Screen.Width / 2 - Me.ScaleWidth / 2
Button (False)
Stb.Panels.Item(1).Text = "กรุณาเลือกอุปกรณ์ควบคุม"
Stb.Panels.Item(2).Text = " "
FrBton.Value = False
com2.Enabled = False
Com1.Enabled = False
mStop.Enabled = False
End Sub
---------------------------------------------------------------------------------------------
Private Sub FrBton_KeyDown(KeyCode As Integer, Shift As Integer)
Stb.Panels.Item(1).Text = t1 & "เดินหน้า"
MSC.Output = Chr(&H20)
End Sub
---------------------------------------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub FrBton_KeyUp(KeyCode As Integer, Shift As Integer)
Mouseup
End Sub
---------------------------------------------------------------------------------------------
Private Sub FrBton_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
        Stb.Panels.Item(1).Text = t1 & "เดินหน้า"
        num = 1
End Sub
---------------------------------------------------------------------------------------------
Private Sub FrBton_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
Mouseup
End Sub
---------------------------------------------------------------------------------------------
Private Sub JKJoystick1_ButtonPressed(ID As Integer)
Select Case ID
Case 1: Lbl2.Caption = "หมุนกล้องขึ้น"
         MSC.Output = Chr(&H80)
Case 2: Lbl2.Caption = "หมุนกล้องไปทางซ้าย"
        MSC.Output = Chr(&H50)
Case 3: Lbl2.Caption = "หมุนกล้องลง"
        MSC.Output = Chr(&H70)
Case 4: Lbl2.Caption = "หมุนกล้องไปทางขวา"
        MSC.Output = Chr(&H60)
End Select
End Sub
---------------------------------------------------------------------------------------------
Private Sub JKJoystick1_ButtonReleased(ID As Integer)
Lbl1.Caption = ""
Lbl2.Caption = ""
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub JKJoystick1_PosChange(NewX As Long, NewY As Long, NewThrottle As Long,
NewRudder As Long)
If NewX = 32678 And NewY = 0 Then
    Lbl1.Caption = "หุ่นยนต์กำลังเดินหน้า"
    MSC.Output = Chr(&H20)
ElseIf NewX = 32678 And NewY = 65446 Then
    Lbl1.Caption = "หุ่นยนต์กำลังถอยหลัง"
    MSC.Output = Chr(&H10)
ElseIf NewX = 0 And NewY = 32678 Then
   Lbl1.Caption = "หุ่นยนต์กำลังเลี้ยวซ้าย"
    MSC.Output = Chr(&H30)
ElseIf NewX = 65446 And NewY = 32678 Then
   Lbl1.Caption = "หุ่นยนต์กำลังเลี้ยวขวา"
    MSC.Output = Chr(&H40)
Else
   Lbl1.Caption = ""
End If
End Sub
---------------------------------------------------------------------------------------
Private Sub Joy_Click()
Con_Robot = 1
Stb.Panels.Item(1).Text = "คุณกำลังควบคุมด้วย Joystick"
Stb.Panels.Item(2).Text = "กรุณาเลือกพอร์ต"
Joy.Enabled = False
Mouse.Enabled = False
Com1.Enabled = True
com2.Enabled = True
mStop.Enabled = True
End Sub
---------------------------------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub LeBton_KeyDown(KeyCode As Integer, Shift As Integer)
Stb.Panels.Item(1).Text = t1 & "เลี้ยวซ้าย"
MSC.Output = Chr(&H30)
End Sub
---------------------------------------------------------------------------------------------
Private Sub LeBton_KeyUp(KeyCode As Integer, Shift As Integer)
Mouseup
End Sub
---------------------------------------------------------------------------------------------
Private Sub LeBton_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
Stb.Panels.Item(1).Text = t1 & "เลี้ยวซ้าย"
num = 3
End Sub
---------------------------------------------------------------------------------------------
Private Sub LeBton_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
Mouseup
End Sub
---------------------------------------------------------------------------------------------
Private Sub Mouse_Click()
Con_Robot = 0
Stb.Panels.Item(1).Text = "คุณกำลังควบคุมด้วย Mouse"
Stb.Panels.Item(2).Text = "กรุณาเลือกพอร์ต"
Mouse.Enabled = False
JKJoystick1.TStop
Com1.Enabled = True
com2.Enabled = True
mStop.Enabled = True
Joy.Enabled = False
End Sub
---------------------------------------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub mStop_Click()
If MSC.PortOpen = True Then
   MSC.PortOpen = False
End If
Com1.Enabled = False
com2.Enabled = False
mStop = False
Button (False)
Stb.Panels.Item(1).Text = "กรุณาเลือกอุปกรณ์ควบคุม"
Stb.Panels.Item(2).Text = " "
Mouse.Enabled = True
Joy.Enabled = True
End Sub
---------------------------------------------------------------------------------
Private Sub ReBton_KeyDown(KeyCode As Integer, Shift As Integer)
Stb.Panels.Item(1).Text = t1 & "ถอยหลัง"
MSC.Output = Chr(&H10)
End Sub
---------------------------------------------------------------------------------
Private Sub ReBton_KeyUp(KeyCode As Integer, Shift As Integer)
Mouseup
End Sub
---------------------------------------------------------------------------------
Private Sub ReBton_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
Stb.Panels.Item(1).Text = t1 & "ถอยหลัง"
num = 2
End Sub
---------------------------------------------------------------------------------
Private Sub ReBton_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
Mouseup
```

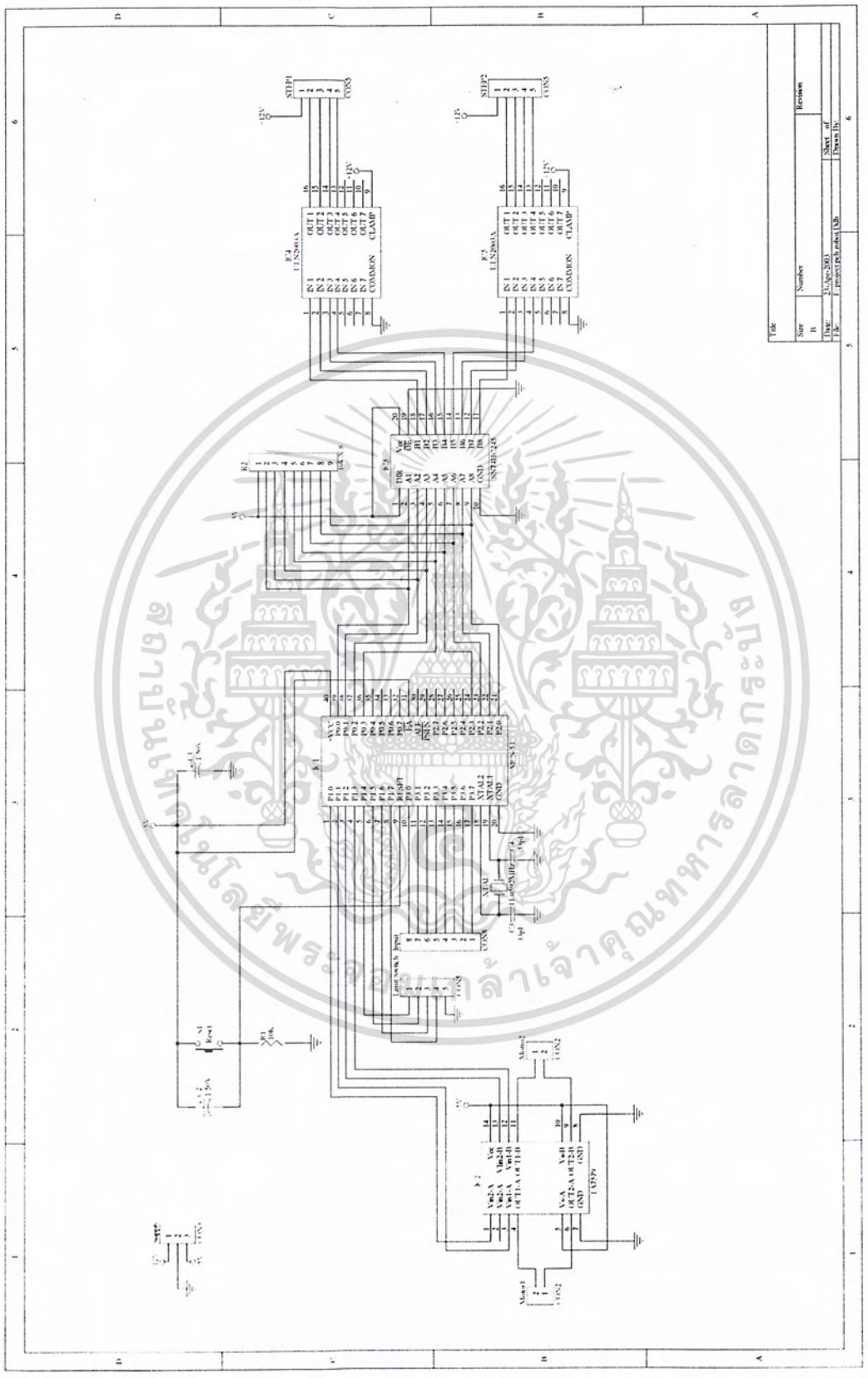
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Sub
---------------------------------------------------------------------------------------------
Private Sub RigBton_KeyDown(KeyCode As Integer, Shift As Integer)
Stb.Panels.Item(1).Text = t1 & "เลี้ยวขวา"
MSC.Output = Chr(&H40)
End Sub
---------------------------------------------------------------------------------------------
Private Sub RigBton_KeyUp(KeyCode As Integer, Shift As Integer)
Mouseup
End Sub
---------------------------------------------------------------------------------------------
Private Sub RigBton_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
Stb.Panels.Item(1).Text = t1 & "เลี้ยวขวา"
num = 4
End Sub
---------------------------------------------------------------------------------------------
Private Sub RigBton_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
Mouseup
End Sub
---------------------------------------------------------------------------------------------
Private Sub Timer1_Timer()
Select Case num
   Case 1: MSC.Output = Chr(&H20)
   Case 2: MSC.Output = Chr(&H10)
   Case 3: MSC.Output = Chr(&H30)
   Case 4: MSC.Output = Chr(&H40)
   Case 5: MSC.Output = Chr(&H80)
   Case 6: MSC.Output = Chr(&H70)
   Case 7: MSC.Output = Chr(&H50)
   Case 8: MSC.Output = Chr(&H60)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

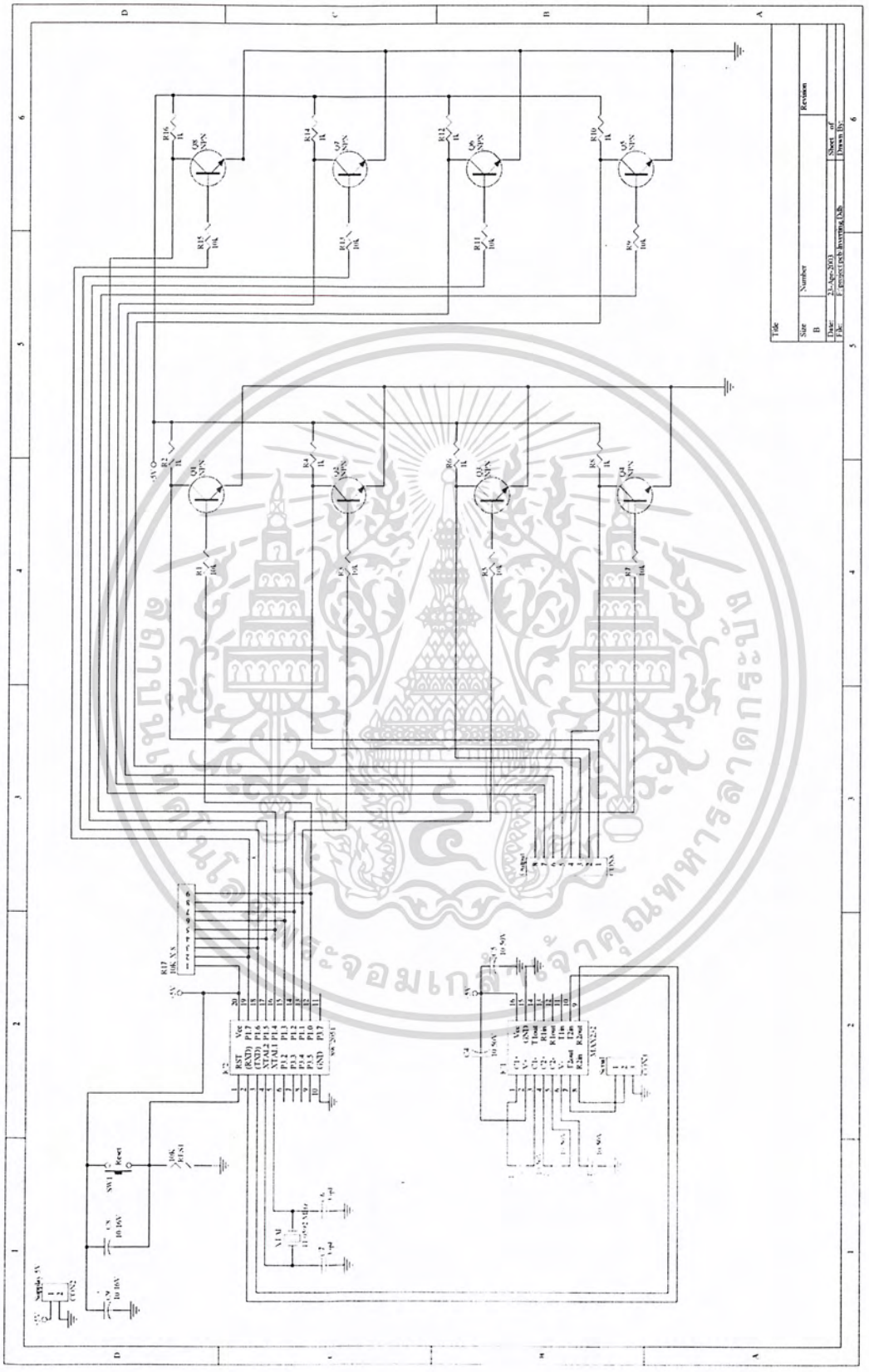
```
End Select
End Sub
---------------------------------------------------------------------------------------------
Private Sub UpBton_KeyDown(KeyCode As Integer, Shift As Integer)
Stb.Panels.Item(1).Text = t2 & "หมุนขึ้น"
MSC.Output = Chr(&H80)
End Sub
---------------------------------------------------------------------------------------------
Private Sub UpBton_KeyUp(KeyCode As Integer, Shift As Integer)
Mouseup
End Sub
---------------------------------------------------------------------------------------------
Private Sub UpBton_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
Stb.Panels.Item(1).Text = t2 & "หมุนขึ้น"
num = 5
End Sub
---------------------------------------------------------------------------------------------
Private Sub UpBton_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
Mouseup
End Sub
---------------------------------------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



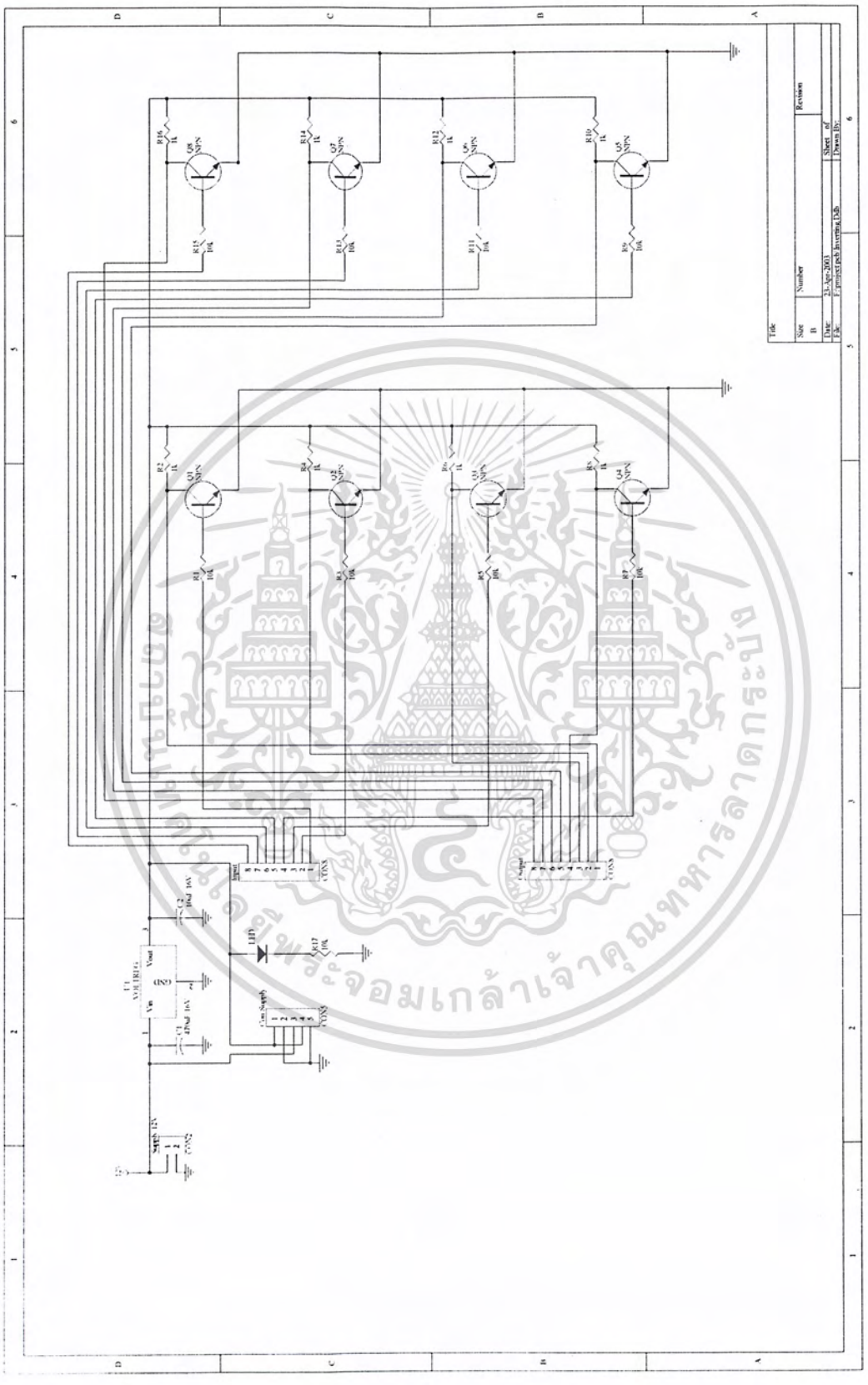
รูปวงจรควบคุมคุณสมบัติมอเตอร์และสเต็ปมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



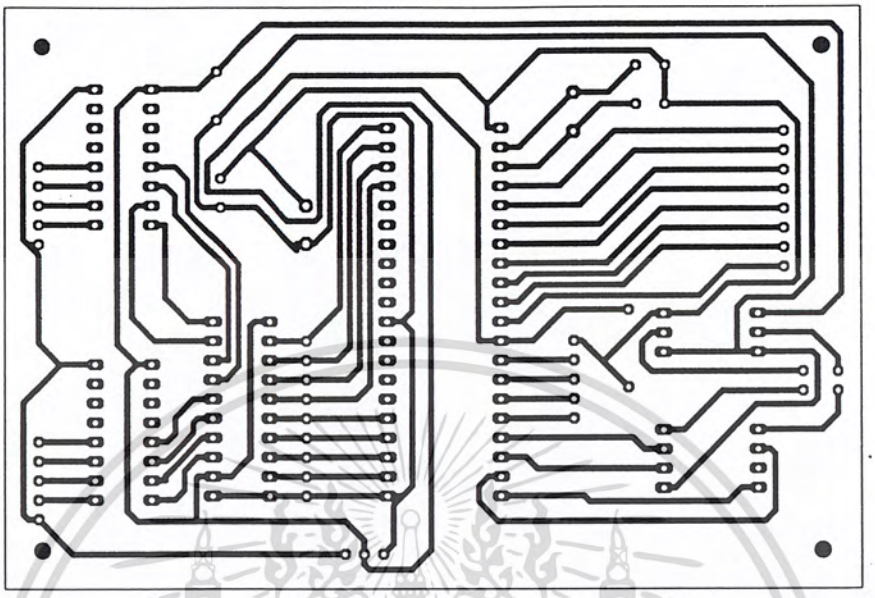
รูปวงจรเปลี่ยนสัญญาณแบบอนุกรมเป็นแบบขนานและวงจร Inverting

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

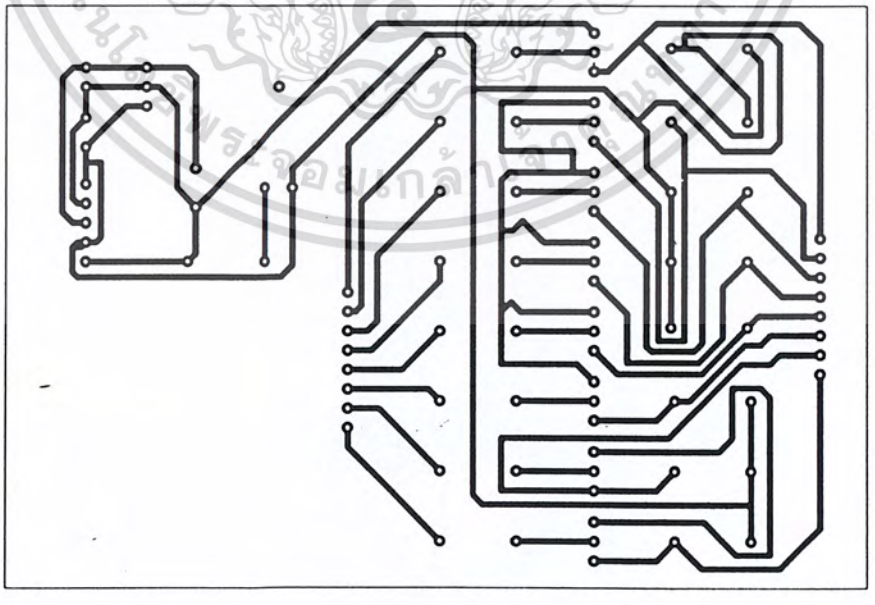


รูปวงจร Inverting

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

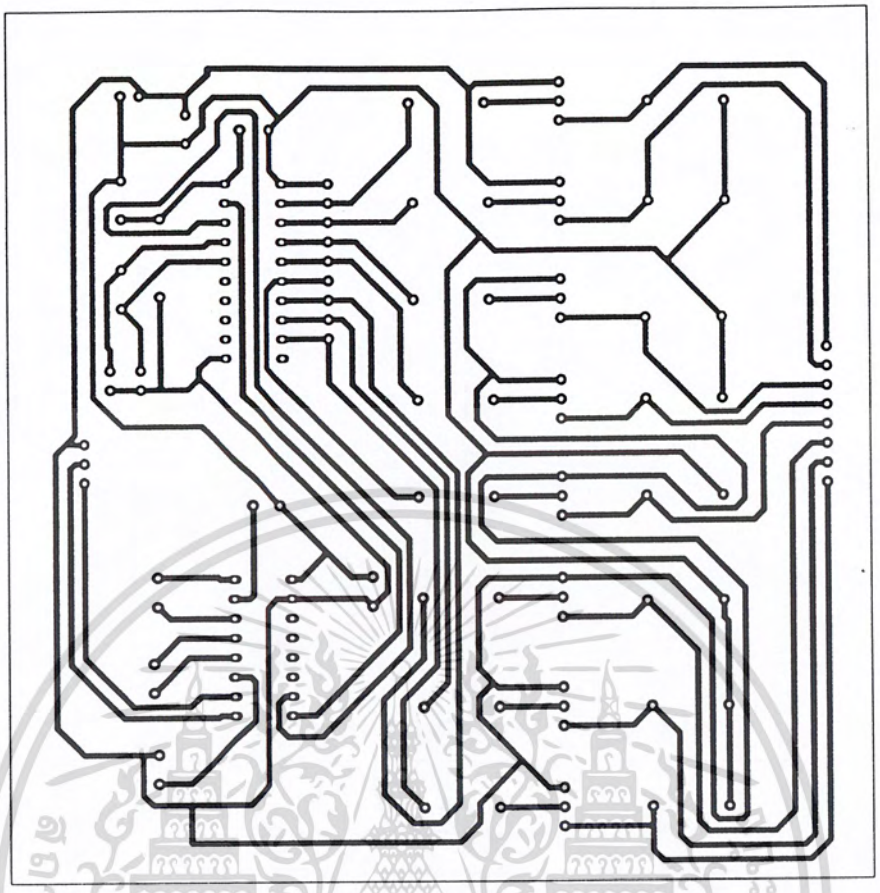


รูปลายวงจรควบคุมดีซีมอเตอร์และสเต็ปปิ้งมอเตอร์



รูปลายวงจร Inverting

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปลายวงจรเปลี่ยนสัญญาณแบบอนุกรมเป็นแบบขนานและวงจร Inverting

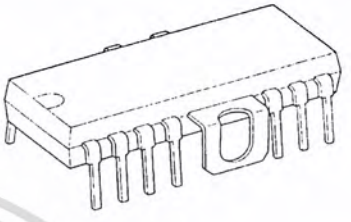
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TOSHIBA BIPOLAR LINEAR INTEGRATED CIRCUIT SILICON MONOLITHIC

TA7279P,TA7279AP

DUAL BRIDGE DRIVER

The TA7279P, TA7279AP are dual bridge driver designed for DC motor rotation control.



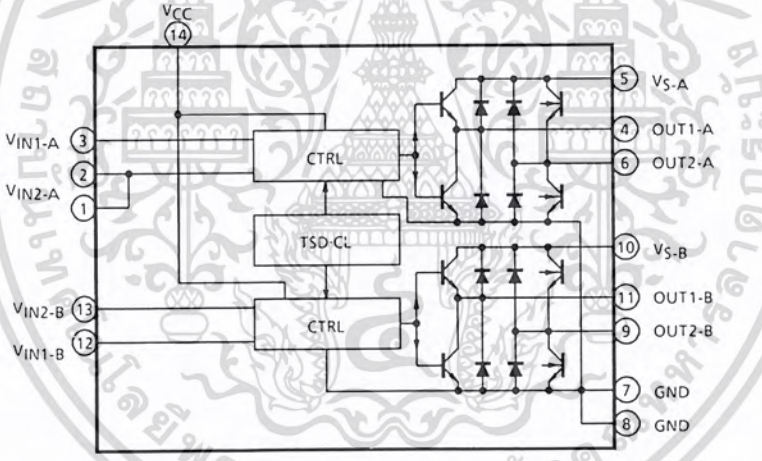
HDIP14-P-500-2.54A

Weight: 3.00 g (Typ.)

FEATURES

- Wide Range of Operating Voltage
  : V_{CC} (opr.) = 6~18 V (P, AP),
  V_S (opr.) = 0~16 V (P) / = 0~18 V (AP)
- Output Current Up to 1.0 A (AVE.), 3.0 A (PEAK)
- Built-in Thermal Shut Down and Current Limiter
- Input Hysteresis for Stable Operation

BLOCK DIAGRAM



961001EBA1

- TOSHIBA is continually working to improve the quality and the reliability of its products. Nevertheless, semiconductor devices in general can malfunction or fail due to their inherent electrical sensitivity and vulnerability to physical stress. It is the responsibility of the buyer, when utilizing TOSHIBA products, to observe standards of safety, and to avoid situations in which a malfunction or failure of a TOSHIBA product could cause loss of human life, bodily injury or damage to property. In developing your designs, please ensure that TOSHIBA products are used within specified operating ranges as set forth in the most recent products specifications. Also, please keep in mind the precautions and conditions set forth in the TOSHIBA Semiconductor Reliability Handbook.
- The products described in this document are subject to foreign exchange and foreign trade control laws.
- The information contained herein is presented only as a guide for the applications of our products. No responsibility is assumed by TOSHIBA CORPORATION for any infringements of intellectual property or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any intellectual property or other rights of TOSHIBA CORPORATION or others.
- The information contained herein is subject to change without notice.

สงวนลิขสิทธิ์ เอกสารนี้จัดทำขึ้นเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

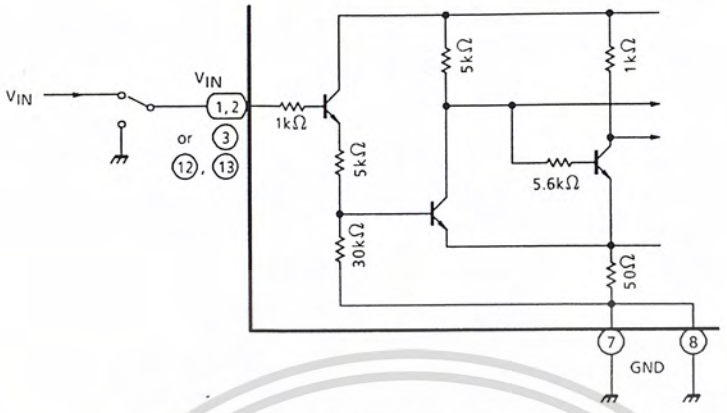
PIN FUNCTION

PIN No.	SYMBOL	FUNCTIONAL DESCRIPTION
1	V_{IN2-A}	A-ch input terminal
2	V_{IN2-A}	
3	V_{IN1-A}	A-ch input terminal
4	OUT1-A	A-ch output terminal
5	V_{S-A}	A-ch Motor drive power supply
6	OUT2-A	A-ch output terminal
7	GND	GND terminal
8	GND	
9	OUT2-B	B-ch output terminal
10	V_{S-B}	B-ch Moter drive power supply
11	OUT1-B	B-ch output terminal
12	V_{IN1-B}	B-ch input terminal
13	V_{IN2-B}	B-ch input terminal
14	V_{CC}	Logic power supply

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

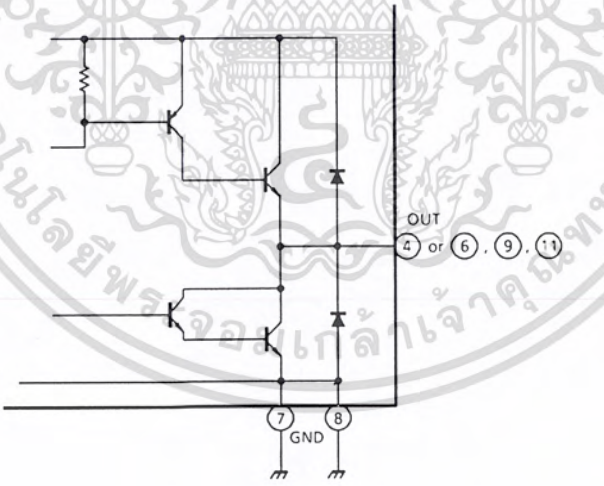
APPLICATION NOTE

(1) Input circuit



Input terminals of (2), (3), (12) and (13) Pin are all high active type and have a hysteresis.
3 μA Typ. of input current is required.
The input circuit is an active high type, as shown in the diagram. When voltage higher than the specified $V_{IN\ (H)}$ is applied, the output is logic "H". When voltage lower than the specified $V_{IN\ (L)}$ is applied or if the input is grounded, the output is logic "L". Since the input current IN flows to the input when logic "H", be careful with the output impedance at the previous step.

(2) Output circuit



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FUNCTION

IN1	IN2	OUT1	OUT2	MODE
1	1	L	L	Brake
0	1	L	H	CW / CCW
1	0	H	L	CCW / CW
0	0	High Impedance		STOP

MAXIMUM RATINGS (Ta = 25°C)

CHARACTERISTIC		SYMBOL	RATING	UNIT
Supply Voltage	AP	V_{CC} (MAX.)	25	V
	P		20	
Motor Drive Voltage	AP	V_S (MAX.)	25	V
	P		18	
Output Current	PEAK	I_O (PEAK)	3.0	A
	AVE.	I_O (AVE.)	1.0	
Power Dissipation		P_D (Note)	2.3	W
Operating Temperature		T_{opr}	−30~75	°C
Storage Temperature		T_{stg}	−55~150	°C

Note: No heat sink.

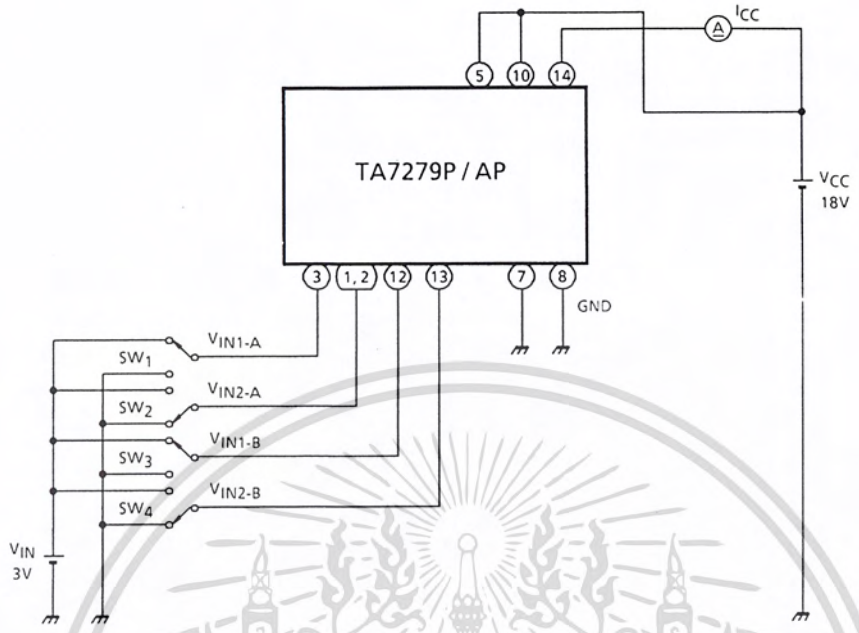
ELECTRICAL CHARACTERISTICS (Ta = 25°C)

CHARACTERISTIC		SYMBOL	TEST CIR-CUIT	TEST CONDITION	MIN	TYP.	MAX	UNIT
Supply Current		I_{CC1}	1	V_{CC} = 18 V, Output Off, Stop mode	14	28	41	mA
		I_{CC2}	1	V_{CC} = 18 V, Output Off, CW / CCW mode	10	29	38	
		I_{CC3}	1	V_{CC} = 18 V, Output Off, Brake mode	8	20	35	
Input Operating Voltage	1 (High)	$V_{IN\,(H)}$	—	T_j = 25°C	3.0	—	V_{CC}	V
	2 (Low)	$V_{IN\,(L)}$	—	T_j = 25°C	—	—	0.8	
Input Current		I_{IN}	2	Sink, V_{IN} = 3 V	—	3	10	μA
Output Saturation Voltage	Upper	V_{SATU-1}	3	I_O = 0.1 A, V_{CC} = V_S = 18 V	—	—	1.1	V
	Lower	V_{SATL-1}	3	I_O = 0.1 A, V_{CC} = V_S = 18 V	—	—	1.0	
	Upper	V_{SATU-2}	3	I_O = 1.0 A, V_{CC} = V_S = 18 V	—	1.2	1.5	
	Lower	V_{SATL-2}	3	I_O = 1.0 A, V_{CC} = V_S = 18 V	—	1.05	1.4	
Leakage Current	Upper	I_{LU}	—	V_S = 25 V	—	—	50	μA
	Lower	I_{LL}	—	V_S = 25 V	—	—	50	
Diode Forward Drop	Upper	V_{FU}	4	I_F = 1 A	—	2.0	—	V
	Lower	V_{FL}	4	I_F = 1 A	—	1.3	—	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

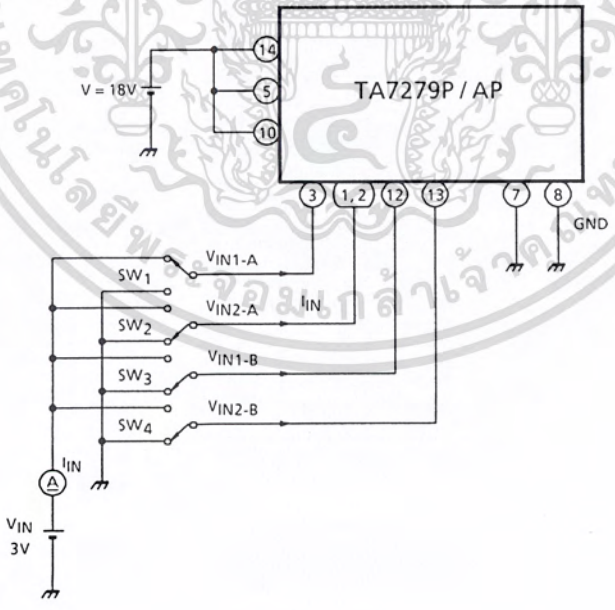
TEST CIRCUIT 1.

$I_{CC1, 2, 3}$



TEST CIRCUIT 2.

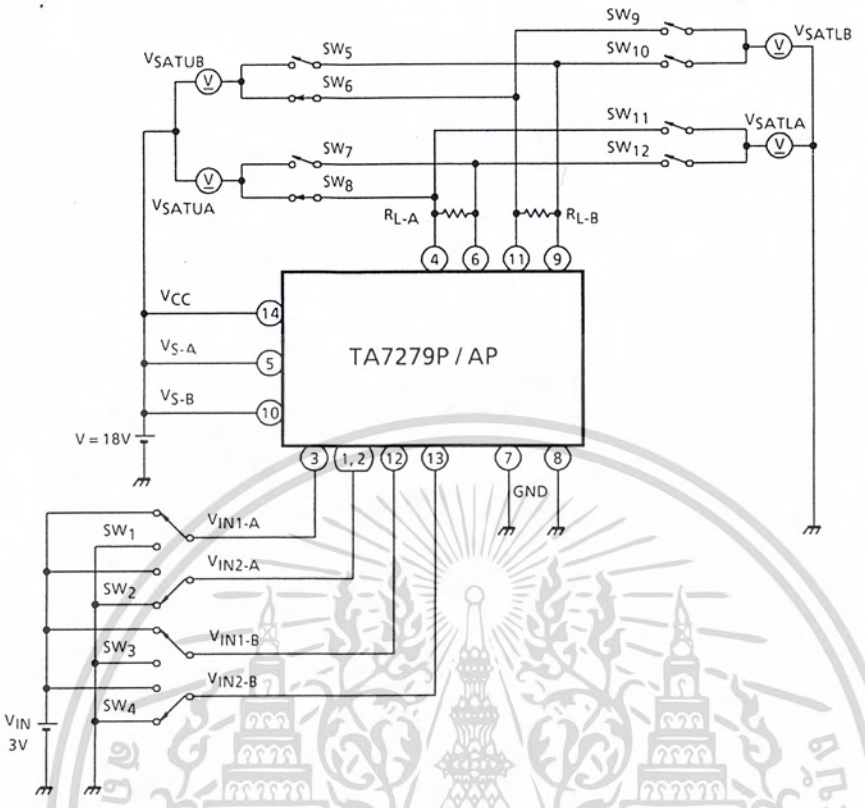
I_{IN} (H), (L)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

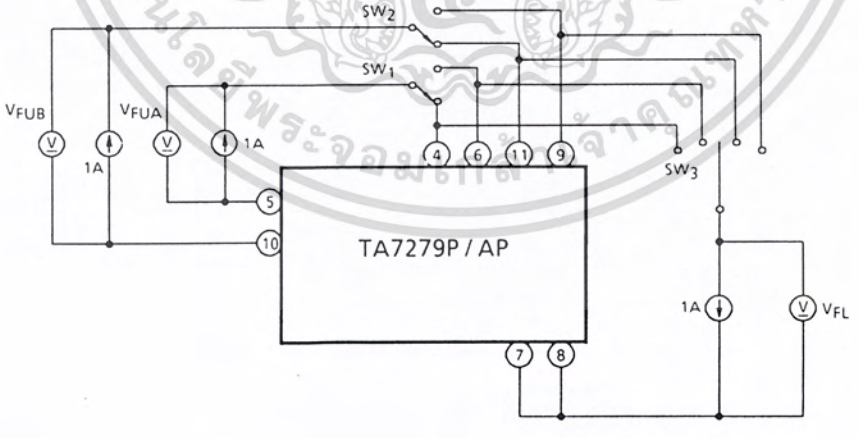
TEST CIRCUIT 3.

$V_{SATU-1, 2}$ / $V_{SATL-1, 2}$

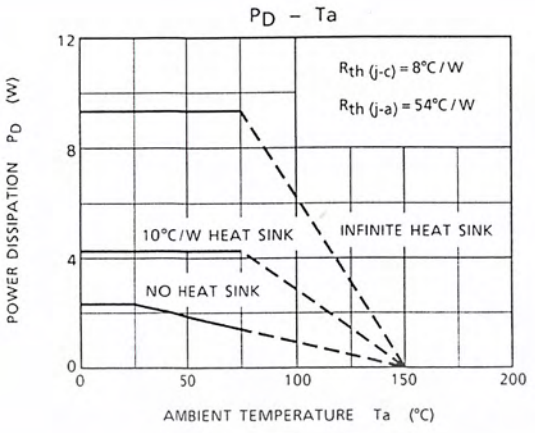


TEST CIRCUIT 4.

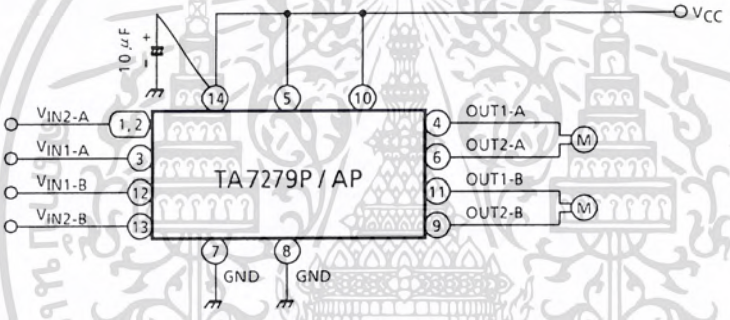
$V_{FU, L}$



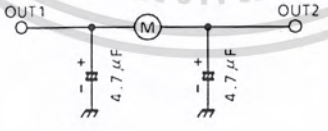
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



APPLICATION CIRCUIT



Problems may result if a capacitor is inserted in parallel to the motor as a measure against noise.
If measures against noise are necessary, connect capacitors as shown in the diagram below.
A larger bypass capacitor between V_{CC} and GND is effective against noise and other problems.
(A capacitance higher than 100 μF is recommended.)



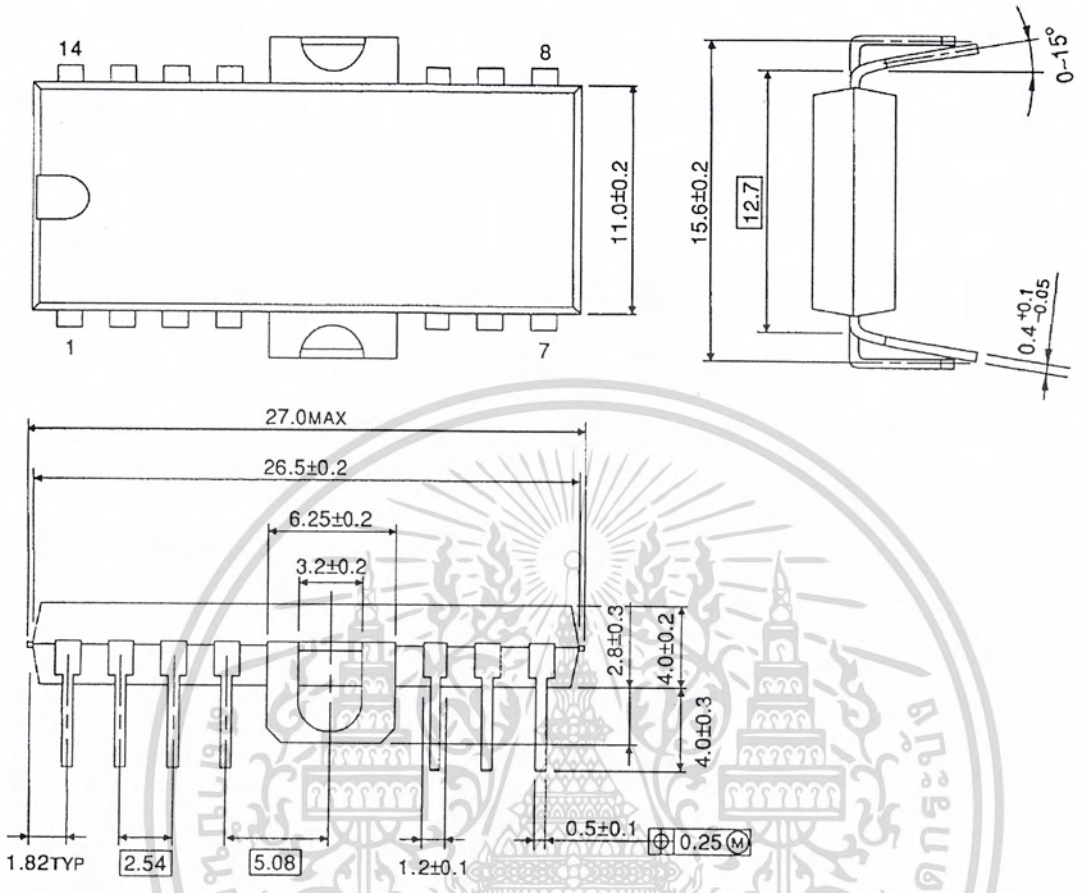
Note: Utmost care is necessary in the design of the output line, V_S and GND line since IC may be destroyed due to short-circuit between outputs, air contamination fault, or fault by improper grounding.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTLINE DRAWING

HDIP14-P-500-2.54A

Unit: mm



Weight: 3.00 g (Typ.)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้