สำนักวิทยบริการฯ พระจอมเกล้าลาดกระบัง

หุ่นยนต์ไต่ผนัง

A CLIMBING ROBOT

โดย

นายจักรกฤษณ์ ทรัพย์เมฆ

นายปิยะพงษ์ ชนะวิทย์

เลขหมู่.....
เลขทะเบียน 42685
วัน, เดือน, ปี - 6 มิ.ย. 2545

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตรบัณฑิต

สาขาเทคโนโลยีอิเล็กทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์ หุ่นยนต์ไต่ผนัง

A CLIMBING ROBOT

จัดทำโดย นายจักรกฤษณ์ ทรัพย์เมฆ

นายปิยะพงษ์ ชนะวิทย์

อาจารย์ที่ปรึกษา ผศ.ดร.ปิติเขต สู้รักษา

ภาควิชาเทคนิคอุตสาหกรรม

ปีการศึกษา 2544

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีเจ้าคุณทหารลาดกระบัง

อนุมัติให้นับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตร์บัณฑิต

คณะกรรมการควบคุมการสอบปริญญานิพนธ์

_____ประธานกรรมการ

( )

_____กรรมการ

( )

_____กรรมการ

( )

_____กรรมการ

( )

_____กรรมการ

( )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หุ่นยนต์ไต่ผนัง

จัดทำโดย นายจักรกฤษณ์ ทรัพย์เมฆ 42015548

นายปิยะพงษ์ ชนะวิทย์ 42015559

อาจารย์ผู้ควบคุมปริญญานิพนธ์ ผศ.ดร. ปิติเขต สู้รักษา

ปีการศึกษา 2544

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ เป็นการศึกษาและสร้างหุ่นยนต์ไต่ผนังที่ควบคุมได้ด้วยอุปกรณ์ควบคุมระยะไกลหรือรีโมตคอนโทรล (REMOTE CONTROL) ให้สามารถเคลื่อนที่ไปได้ในผนังที่เป็นตาข่ายลวด สามารถที่จะเคลื่อนตัวขึ้นหรือลงในแนวดิ่ง และ สามารถเคลื่อนตัวไปทางซ้ายหรือขวาได้อีกด้วย ซึ่งเราจะใช้ทั้งอุปกรณ์ทางเครื่องกลและเครื่องกลไฟฟ้า อุปกรณ์ทางไฟฟ้า และอุปกรณ์ทางอิเล็กทรอนิกส์ มาประยุกต์ใช้งาน โดยอุปกรณ์ทางเครื่องกลและเครื่องกลไฟฟ้าที่สำคัญนั้นได้แก่ เซอร์โวมอเตอร์(SERVO MOTOR) ส่วนอุปกรณ์ทางอิเล็กทรอนิกส์ได้แก่ ไมโครคอนโทรลเลอร์ ตระกูล MCS-51 โดยแนวทางการสร้างหุ่นยนต์ที่ได้ สามารถนำไปประยุกต์ใช้งานกับ การทำความสะอาดตึกสูงได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A CLIMBING ROBOT

JACKKRIT SABMEK 42015548

PIYAPONG CHANAWIT 42015559

Academic Adviser

Asst.Prof.Dr.PITIKHATE SOORAKSA

Academic Year 2001

ABSTRACT

The study aims to realize and build a "climbing robot" that can control remotely. The robot can move to desire direction along the balustrade net. The climbing robot composes of mechanical and electronic parts and components. The major components are servomotors and MCS-51 microcontrollers . For application example, the robot can be used in cleaning high-rise buildings.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอบพระคุณอาจารย์ ผู้ช่วยศาสตราจารย์ ดร.ปิติเขต สู้รักษา ที่ให้โอกาสช่วยเหลือให้คำแนะนำต่างๆ จนปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยดี ขอบคุณภาควิชาเทคนิคอุตสาหกรรม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ให้การสนับสนุนจนกระทั่งโครงงานนี้สำเร็จลุล่วงไปได้ด้วยดี ขอขอบคุณบิดา มารดา ที่ได้ให้การเลี้ยงดูให้ได้รับโอกาสทางการศึกษา และยังเป็นกำลังใจให้ตลอด ขอบคุณพี่ๆเพื่อนๆและน้องๆที่คอยช่วยเหลือและให้คำปรึกษา

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทที่ 1 บทนำ	1
1.1 ที่มาของโครงงาน	1
1.2 จุดประสงค์	1
1.3 ขอบเขตของการศึกษาและหลักการเบื้องต้น	1
1.4 ระยะเวลาในการดำเนินงาน	2
1.5 มีใครทำมาแล้วบ้าง	3
บทที่ 2 ทฤษฎีและหลักการ	5
2.1 ทฤษฎีพื้นฐาน	5
2.2 อุปกรณ์และหน้าที่การทำงานของอุปกรณ์	11
บทที่ 3 การดำเนินการวิจัย	27
3.1 การออกแบบทางฮาร์ดแวร์	27
3.1.1การออกแบบโครงสร้างของหุ่นยนต์	27
3.1.2การออกแบบท่าเดินของหุ่นยนต์ไต่กำแพง	27
3.1.2.1การเดินหน้า	28
3.1.2.2การถอยหลัง	32
3.1.2.3 การเลี้ยวขวา	34
3.1.2.4 การเลี้ยวซ้าย	36
บทที่ 4 การทดลองและผลการทดลอง	38
4.1 การทดลองจับรูปคลื่นด้วยออสซิลโลสโคป	38
4.2 บันทึกผลการทดลองที่ได้ลงในตาราง	43
4.3 สรุปผลการทดลอง	43
บทที่ 5 การวิเคราะห์	44
บทที่ 6 สรุปโครงงานและปัญหาการทำโครงงาน	46
6.1 สรุปโครงงาน	46
6.2 สรุปปัญหาการทำโครงงาน	46
ภาคผนวก ก. รูปโครงสร้างหุ่นยนต์	
ภาคผนวก ข. วงจรอิเล็กทรอนิกส์	
ภาคผนวก ค. โปรแกรมควบคุม	
ภาคผนวก ง. รายละเอียด MCS-51	
บรรณานุกรม	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่ 1.1 ขณะทำการอธิบายหลักการหุ่นยนต์ไต่กำแพงตัวแรก 2

รูปที่ 1.2 เป็นรูป3มิติของ หุ่นยนต์ Caterpillar Cable Walker 3

รูปที่ 1.3 หุ่นยนต์ Caterpillar Cable Walker 4

รูปที่ 1.4 การเดินของหุ่นยนต์ Roma 4

รูปที่ 2.1 ลักษณะหุ่นยนต์ระบบทรงกระบอก 7

รูปที่ 2.2 ลักษณะหุ่นยนต์ระบบทรงกลม 8

รูปที่ 2.3 ลักษณะหุ่นยนต์ทรงลูกบาศก์ 8

รูปที่ 2.4 แสดงโมเมนต์ 10

รูปที่2.5 แสดงคาบเวลาของพัลซ์ที่เป็นตัวกำหนดองศาของเซอร์โวมอเตอร์ 11

รูปที่2.6 รูปแสดงเซนเซอร์ต่อแกนร่วมเดียวกันกับแกนของมอเตอร์ 12

รูปที่2.7 รูปแสดงเฟืองทดต่อแกนร่วมเข้ากับแกนของเฟืองทดของ มอเตอร์ 12

รูปที่2.8แสงตัวส่งแสง อินฟาเรดไปยังตัวรับสัญญาณแสง อินฟาเรดซึ่งจะอยู่ในด้านตรงกันข้าม13

รูปที่2.9แสดงสัญญาณ เอาท์พุท ของ ชุดรับแสง อินฟาเรดมีลักษณะเป็น พัลซ์ 13

รูปที่ 2.10แสดงวิธีสร้าง ช่อง เป็น 2 ชุดเหลื่อมกัน 90 องศา 14

รูปที่ 2.11แสดงวิธีสร้าง ช่อง เป็น 2 ชุดเหลื่อมกัน 90 องศา 14

รูปที่ 2.12 แสดงสัญญาณ เอาท์พุท จาก เซอร์โวมอเตอร์ ค่าเท่ากับ 2 บิท 15

รูปที่ 2.13 แสดงตัวอย่างสัญญาณ เอาท์พุท จาก เซอร์โวมอเตอร์ ค่าเท่ากับ 2 บิท 15

รูปที่ 2.14 a แสดงไดอะแกรมตัวต้านทานปรับค่าได้ 16

รูปที่ 2.14 b รูปแสดงตัวต้านทานปรับค่าได้ 16

รูปที่ 2.15 แสดงการนับพัลซ์ด้วยไมโครโปรเซสเซอร์ 17

รูปที่ 2.16 แสดง FLOW CHART ลำดับการทำงานของไมโครโปรเซสเซอร์ 18
ที่ทำการตรวจสอบการหมุนของมอเตอร์

รูปที่ 2.17 แสดงโครงสร้างภายในของไมโครคอนโทรลเลอร์ตระกูล MCS-51 20

รูปที่ 2.18 การต่อสัญญาณนาฬิกาจากภายนอกจะต่อ XTAL1 ลงกราวน์และต่อสัญญาณ 22

รูปที่ 2.19 การใช้สัญญาณนาฬิกาจากภายนอก 23

รูปที่ 2.20 การต่อสัญญาณรีเซต 23

รูปที่3.1 แสดงโครงสร้างของหุ่นยนต์เมื่อมองจากด้านหน้า 27

รูปที่ 3.2 แสดงโครงสร้างหุ่นยนต์เมื่อมองจากด้านบน 28

รูปที่ 3.3 หุ่นยนต์ขณะอยู่ในสภาวะเริ่มต้นก่อนทำการเดิน 29

รูปที่ 3.4แสดงการยกแขนเมื่อมองจากด้านข้าง 29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.5 ขณะหุ่นยนต์เริ่มทำการไต่ขึ้นในจังหวะแรก 29
รูปที่ 3.6 ขณะหุ่นยนต์ใช้ขาขวาจับลวดและขาซ้ายปล่อยลวด 30
รูปที่ 3.7 แสดงขณะหุ่นยนต์ก้าวขาเพื่อเพิ่มระยะทาง 31
รูปที่ 3.8 แสดงสภาวะสิ้นสุดหลังจากเดินไปข้างหน้า 31
รูปที่ 3.9 หุ่นยนต์อยู่ในสภาวะเริ่มต้นเตรียมที่จะถอยหลัง 32
รูป่ที่ 3.10 การเดินถอยหลังในจังหวะแรก 32
รูปที่ 3.11 แสดงการเดินในจังหวะต่อไปเพื่อเพิ่มระยะทาง 33
รูปที่ 3.12 แสดงสภาวะสิ้นสุดจากทำการเดินถอยหลัง 33
รูปที่ 3.13 แสดงสภาวะเริ่มต้นก่อนทำการเลี้ยวขวา 34
รูปที่ 3.14 ขณะทำการเลี้ยวขวาเมื่อสิ้นสุดจังหวะแรก 34
รูปที่ 3.15 หุ่นยนต์ในสภาวะที่ทำการเลี้ยวขวาเสร็จสิ้นแล้ว 35
รูปที่ 3.16 แสดงสภาวะเริ่มต้นก่อนทาการเลี้ยวซ้าย 36
รูปที่ 3.17 ขณะทำการเลี้ยวซ้ายเมื่อสิ้นสุดจังหวะแรก 36
รูปที่ 3.18 หุ่นยนต์ในสภาวะที่ทำการเลี้ยวซ้ายเสร็จสิ้นแล้ว 37
รูปที่ 4.1 แสดงรูปคลื่นพัลซ์ขนาด 1 ms 38
รูปที่ 4.2 แสดงการทิศการหมุนของเซอร์โวมอเตอร์ขณะสร้าง พัลซ์ 1 ms 39
รูปที่ 4.3 แสดงรูปคลื่นพัลซ์ขนาด 1.5 ms 39
รูปที่ 4.4 แสดงการทิศการหมุนของเซอร์โวมอเตอร์ขณะสร้าง พัลซ์ 1.5 ms 40
รูปที่ 4.5แสดงรูปคลื่นพัลซ์ขนาด 2 ms 40
รูปที่ 4.6 แสดงการทิศการหมุนของเซอร์โวมอเตอร์ขณะ พัลซ์ 2 ms 41
รูปที่ 4.7แสดงรูปคลื่นพัลซ์ขนาด 1.25 ms 41
รูปที่ 4.8แสดงการทิศการหมุนของเซอร์โวมอเตอร์ขณะ พัลซ์ 1.25 ms 42
รูปที่ 4.9 แสดงรูปคลื่นพัลซ์ขนาด 1.65 ms 42
รูปที่ 4.10 แสดงการทิศการหมุนของเซอร์โวมอเตอร์ขณะ พัลซ์ 1.65ms 43
รูปที่ 5.1 การต่อรีเลย์ให้กับมอเตอร์แต่ละตัว 44
รูปที่ 5.2 แสดงบล๊อกการทำงานของมอเตอร์ 45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 2.1 แสดงการ XOR ด้วยค่าต่างๆ	16
ตารางที่ 2.2 ไมโครคอนโทรลเลอร์ตระกูล MCS-51	19
ตารางที่ 2.3 สัญญาณต่าง ๆ ของพอร์ต P3	26
ตารางที่ 4.1 แสดงการทิศทางการหมุนของมอเตอร์และองศาที่เปลี่ยนไป	43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่1
บทนำ

1.1 ที่มาของโครงงาน

ในภาวะปัจจุบันความเจริญก้าวหน้าทางเทคโนโลยีมีบทบาทสำคัญเป็นอย่างยิ่ง ไม่ว่าจะเป็นหน่วยงานใด ทั้งภาครัฐและเอกชนต่างก็พึ่งพาเทคโนโลยีทั้งสิ้น และ ก่อนที่จะนำเทคโนโลยีต่างๆมาใช้งาน เทคโนโลยีเหล่านั้นจะต้องผ่านกระบวนการทดลอง กำหนดทฤษฎี ต่างๆ และ สุดท้ายนำมาประยุกต์ใช้กับชีวิตประจำวัน ดังนั้นโครงงานนี้ก็มีจุดมุ่งหมายที่จะให้โครงงานนี้เป็นต้นแบบในการสร้างหุ่นยนต์ไต่ผนัง ที่เป็นผนังซึ่งมีลักษณะคล้ายกับตาข่ายลวด เพื่อที่ว่าในอนาคตหากมีความต้องการที่จะนำไปใช้ทำงาน ในสภาวะแวดล้อมที่ประกอบไปด้วยตาข่ายลวด แทนที่เราจะให้มนุษย์เข้าไปทำงานตรงจุดนี้ เราก็สามารถหาสิ่งที่จะมาทำงานตรงนี้แทนมนุษย์ได้ด้วยการติดตั้งอุปกรณ์เสริมไปกับหุ่นยนต์ แล้วสามารถควบคุมหุ่นยนต์ได้จากระยะทางไกลได้ ซึ่งก็จะเป็นการลดอัตราเสี่ยงต่อการเกิดอุบัติเหตุของมนุษย์ได้

ดังนั้นก่อนที่เราจะดำเนินการถึงขั้นติดตั้งอุปกรณ์เสริมที่ตัวหุ่นยนต์ เราควรจะต้องสร้างหุ่นยนต์ที่สามารถเดินบนตาข่ายลวดได้ก่อน หากสามารถทำได้สำเร็จแล้วก็เพียงแต่ติดตั้งอุปกรณ์เสริม ให้เหมาะสมกับงานที่เราต้องการจะทำต่อไป

1.2 จุดประสงค์

จุดประสงค์ของโครงงานนี้ คือ การสร้างหุ่นยนต์ที่สามารถเดินบนตาข่ายลวด ได้ในแนวตั้งฉากกับพื้นโลก โดยสามารถกำหนดให้หุ่นยนต์สามารถเดินได้ทั้งในแนวดิ่ง ขึ้นหรือลงได้ และสามารถเดินไปทางซ้ายหรือทางขวา ได้ในแนวตั้งฉากกับพื้นโลก โดยมีอุปกรณ์ควบคุมระยะไกลรีโมตคอนโทรล(Remote Control) เป็นตัวเชื่อมโยงการควบคุมหุ่นยนต์ ระหว่างมนุษย์ และ หุ่นยนต์

1.3 ขอบเขตของการศึกษาและหลักการเบื้องต้น

จากความต้องการในเบื้องต้นจากโครงงานแรกเราต้องการให้หุ่นยนต์สามารถที่จะเกาะยึดติดอยู่กับตาข่ายลวดให้ได้ก่อนด้วยการสร้างเป็นแขนและมือของหุ่นยนต์ โดยที่มือมีนิ้วจำนวน4 นิ้วไว้ยึดเกาะกับตาข่ายลวด โดยเราเขียนโปรแกรมออกพอร์ตของ MCS-51 และนำไปส่งให้วงจรขับไปขับ DC MOTOR ให้หมุนตามเข็มหรือทวนเข็มเพื่อที่จะไปทำให้นิ้วของหุ่นสามารถจับ หรือ ปล่อยวัตถุได้ ซึ่งได้ผลเป็นที่น่าพอใจ แต่กลับมีปัญหาในด้านของมอเตอร์ที่จะใช้ยกแขน เนื่องจากว่าเราใช้อุปกรณ์มากชิ้น โดยเฉพาะในส่วนที่เป็นเหล็ก ทำให้มีน้ำหนักมาก ดังนั้นมอเตอร์ที่ใช้ยกแขนของหุ่นยนต์จึงไม่สามารถที่จะยกแขนของหุ่นยนต์ได้ ทำให้เฟืองที่อยู่ในมอเตอร์แตก ประกอบกับการจับของนิ้วกับรูตาข่ายเกิดการไม่เสถียร จึงได้ทำการทดลองใหม่ โดยเริ่มจากการออกแบบหุ่นยนต์ใหม่ โดยกำหนดให้หุ่นยนต์มี 2 ขา และที่ปลายนิ้วเราจะทำให้เป็นปลายแหลมเพื่อที่จะสามารถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สอดเข้าไปในรูของตาข่ายได้ซึ่งจะทำให้เสถียรมากกว่าการที่ใช้นิ้ว 4 นิ้ว จับตาข่าย โดยเราจะกำหนดเขียนโปรแกรมในการเดินขึ้น เดินลง เลี้ยวซ้าย และเลี้ยวขวา แยกเอาไว้แล้วแต่โปรแกรมและทำการส่งสัญญาณพัลซ์ ซึ่งมีความถี่ต่างกัน มาเป็นตัวกระตุ้นให้แต่ละโปรแกรมทำงานว่าจะทำงานในคำสั่งใด

1.4 ระยะเวลาในการดำเนินงาน

ใช้ระยะเวลาทั้งสิ้นประมาณ 1ปี เนื่องจากในช่วงแรกได้มีการออกแบบตัวหุ่นยนต์ผิดพลาด ไม่สามารถที่จะรองรับน้ำหนัก มากๆได้ และอุปกรณ์ แต่ละชนิดมีราคาค่อนข้างสูงจึงทำให้การดำเนินงานขาดระยะไป โดยสามารถบอกขันตอนการดำเนินงานได้โดยคร่าวดังนี้

- ต.ค.43-ม.ค. 44 ค้นหาข้อมูลจากอินเตอร์เนต และทำการออกแบบสร้างหุ่นตัวแรกขึ้นมาเป็นหุ่นยนต์ 4ขา สามารถควบคุมกลไกมือได้ในระดับหนึ่ง
- ก.พ. – มิ.ย. 44 พยายามแก้ไขปัญหาหุ่นยนต์ 4 ขา เนื่องจากมีปัญหาเซอร์โวมอเตอร์ไม่สามารถรับน้ำหนักของตัวหุ่นได้

รูปที่ 1.1 ขณะทำการอธิบายหลักการหุ่นยนต์ไต่กำแพงตัวแรก

- มิ.ย. – ก.ค.44 ค้นคว้าหาข้อมูลเพิ่มเติมและได้ทำการออกแบบหุ่นยนต์ 3ขา ขึ้นมาซึ่งได้ทำโครงเสร็จเรียบร้อย แต่คาดว่าไม่สำเร็จเนื่องจากมีขนาดใหญ่และน้ำหนักมาก
- ส.ค – ก.ย. 44 เปลี่ยนโครงสร้างหุ่นยนต์เป็นครั้งที่ 3 โดยทำเป็นหุ่นยนต์ 2 ขา และตัวโครงทำขึ้นจากพลาสติก
- ต.ค. 44 ทำการทดลองโปรแกรมและการทำงานของหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

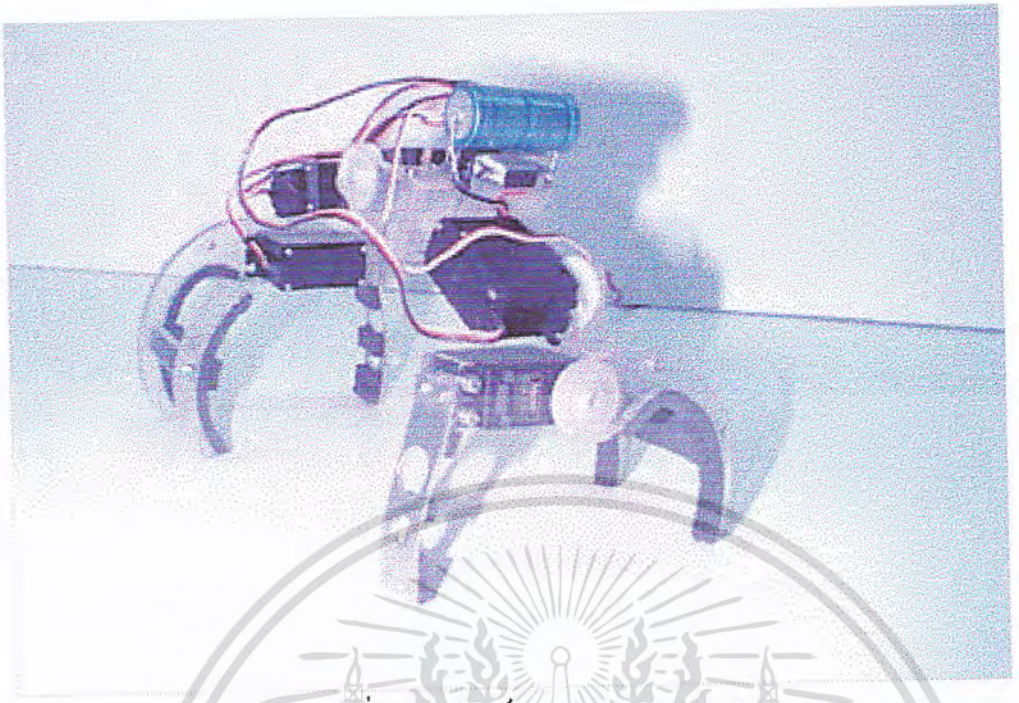
1.5 มีใครทำมาแล้วบ้าง

จากการสำรวจค้นคว้าทางอินเตอร์เนท ผลปรากฏว่าหุ่นยนต์ส่วนใหญ่ที่สร้างขึ้นมานั้น มักจะอยู่ใน รูปแบบของกำแพงผิวเรียบ หรือ กระจก บางครั้งก็อยู่ในรูปแบบของหุ่นยนต์ไต่เชือกบ้าง แต่หุ่นยนต์ที่จะไต่ตาข่ายลวดนั้น หาได้น้อยมาก เท่าที่สำรวจมา มี เพียง 2 แห่งเท่านั้นได้แก่

1.5.1 Caterpillar Cable Walker ของมหาวิทยาลัย McGrill University ประเทศแคนาดา เป็นของ Es 419 Machanical Engineering Project โดยมีที่ปรึกษาคือ Dr.G knopf และยังได้รางวัล Faculty of engineering technic night competition ในปี 1997-1998 โดยใช้ชื่อหุ่นยนต์ว่า "Caterpillar Cable Walker" สามารถเข้าไปดูได้ใน http:// www.cim.mcgill.ca และเป็นที่น่าเสียดายที่หุ่นยนต์ไม่ได้บอกรายละเอียดอะไรมากนัก และยังไม่มีรูปขณะทำงานให้ดู เราจึงไม่ทราบว่า สามารถทำงานได้จริงหรือไม่ เนื่องจากผลงานชิ้นนี้ได้ทำในปี 1998 จนกระทั่งปัจจุบัน ไม่มีรายละเอียดใดแสดงต่อไปเลยว่า สามารถทำงานได้จริง

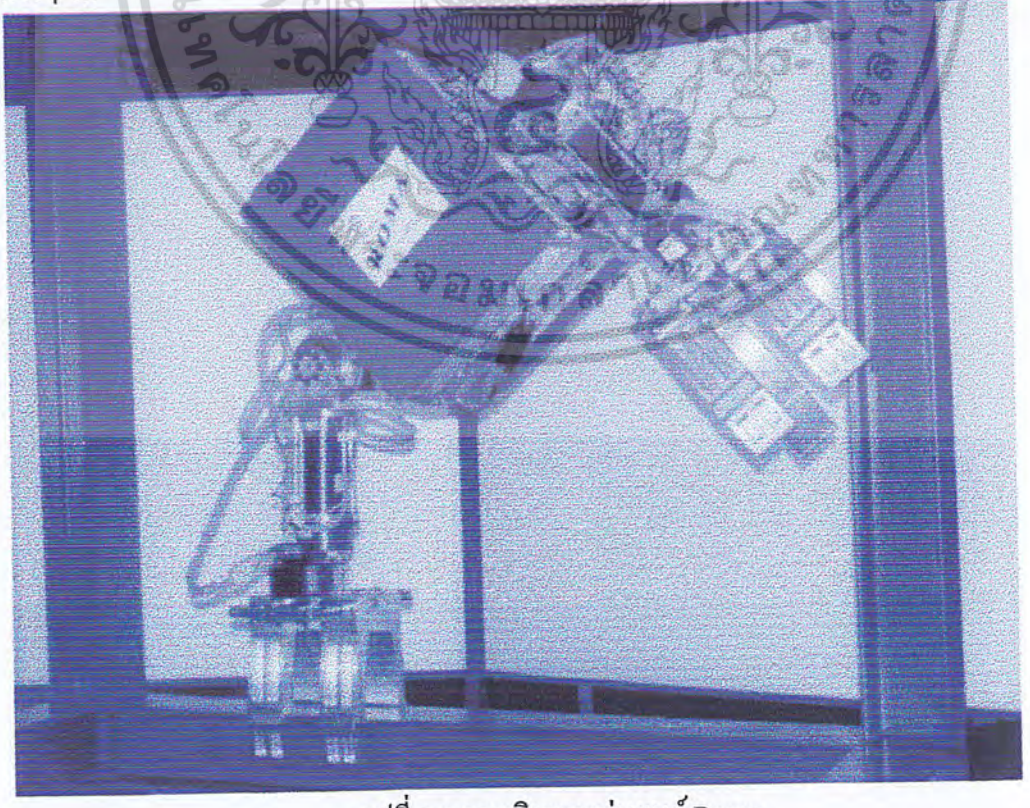
รูปที่ 1.2 เป็นรูป3มิติของ หุ่นยนต์ Caterpillar Cable Walker

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.3 หุ่นยนต์ Caterpillar Cable Walker

1.5.2 หุ่นยนต์ Roma ของมหาวิทยาลัย Universidad Carlos lll de Madrid ประเทศเสปน สามารถเข้าไปดูได้ใน http://www.uwe.ac.uk/clawer/newsletters/issue2/Roma.html/ หุ่นยนต์ตัวมีลักษณะการปีน ท่อ หรือ โครงเหล็ก และยังสามารถเคลื่อนไหวได้ 3 มิติ ทั้งยังสามารถนำไปประยุกต์งานต่างๆ ได้



รูปที่ 1.4 การเดินของหุ่นยนต์ Roma

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

ในการออกแบบหุ่นยนต์ไต่ผนังนั้น เราได้ทำการออกแบบมาหลายแบบด้วยกัน และแต่ละแบบก็มีข้อผิดพลาดต่างๆกันไป แต่ปัญหาส่วนใหญ่มักจะอยู่ที่ เซอร์โวมอเตอร์นั้นไม่สามารถที่จะมีแรงยกน้ำหนักของหุ่นยนต์ไต่กำแพงได้ จนสุดท้ายหุ่นตัวที่ 3 ที่เราได้ทำการออกแบบนั้น ได้นำเรื่องของ คาน มาใช้ ซึ่งก็สามารถช่วยแก้ปัญหาลงได้

สำหรับการออกแบบนั้นเราได้ทำการออกแบบส่วนที่เป็นโครงสร้างของหุ่นยนต์ไต่กำแพงก่อนว่า ควรจะมีรูปร่างลักษณะเช่นไร และจากนั้นเราก็ทำการออกแบบส่วนที่เป็นวงจรทางอิเล็กทรอนิกส์เพื่อที่จะมา ควบคุมแต่ละส่วนของหุ่นยนต์ไต่กำแพง ส่วนสุดท้ายก็เป็นเรื่องของการออกแบบโปรแกรมว่าเราจะสามารถ สั่งงานให้อุปกรณ์ต่างๆทำงานอย่างไร ในบทที่ 2 นี้เราจะกล่าวถึงทฤษฎีต่างๆที่เกี่ยวกับ อุปกรณ์ที่ใช้และทฤษฎีพื้นฐานทั่วไป ส่วนในการออกแบบวงจรทางอิเล็กทรอนิกส์ และการออกแบบโปรแกรมนั้นเราจะกล่าวถึงในบทต่อๆไป

สำหรับอุปกรณ์ที่ใช้สร้างหุ่นยนต์ไต่กำแพงประกอบด้วยอุปกรณ์ดังต่อไปนี้

1.เซอร์โวมอเตอร์สามารถยกน้ำหนักได้ 21 กิโลกรัม จำนวน	2	ตัว
2.เซอร์โวมอเตอร์สามารถยกน้ำหนักได้ 2 กิโลกรัม จำนวน	4	ตัว
3.ไมโครคอนโทรลเลอร์ เบอร์ 89 S 52 จำนวน	3	ตัว
4. รีเลย์ จำนวน	12	ตัว

หลังจากที่เราได้ทราบแล้วว่า เราจะต้องใช้อุปกรณ์ใดบ้างในการออกแบบหุ่นยนต์ไต่กำแพง ขั้นตอนต่อไปที่เราควรทราบก็คือ การทำงานของอุปกรณ์แต่ละอย่าง และทฤษฎีพื้นฐาน เสียก่อนเพื่อที่จะทำให้ทราบว่า เราควรจะใช้อุปกรณ์ตัวใด ทำหน้าที่ที่เราต้องการ และยังจะช่วยให้เราทำการออกแบบการขึ้นรูปโครงสร้างของหุ่นยนต์ไต่กำแพงได้ดีอีกด้วย

2.1ทฤษฎีพื้นฐาน

ความหมายของหุ่นยนต์ คือ อุปกรณ์ หรือ เครื่องจักร อัตโนมัติชนิดหนึ่ง ซึ่งสามารถปฏิบัติงานเลียนแบบการทำงานของมนุษย์ หรือปฏิบัติการได้เทียบเคียงกับ ภูมิปัญญาของมนุษย์ เป็นความหมายตามพจนานุกรม แต่ความหมายในมุมมองของเครื่องมืออุตสาหกรรม จะหมายถึง เครื่องมือชนิดหนึ่งที่สามารถ โปรแกรมการทำงานใหม่ได้ สามารถออกแบบให้ทำงานได้หลากหลายเพื่อเคลื่อนย้ายวัสดุชิ้นส่วนงาน เครื่องมือชนิดต่างๆ หรือ เครื่องมือพิเศษชนิดหนึ่ง ซึ่งเคลื่อนที่ตามโปรแกรมที่กำหนดไว้เพื่องานต่างๆ ตามต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1 ส่วนประกอบพื้นฐานของหุ่นยนต์

2.1.1.1 ตัวปฏิบัติการ (Manipulator) คือส่วนประกอบของชุดข้อต่อที่ใช้เป็นตัวเชื่อมระหว่าง อุปกรณ์ต่ออุปกรณ์ แบ่งเป็น 3 ส่วนย่อย คือ ชุดเชื่อมต่อหลัก ชุดเชื่อมต่อรอง และตัวกระทำการปลายสุด

1. ชุดเชื่อมต่อหลัก คือชุดคู่แกนเชื่อมต่อ-จุดหมุน ซึ่งจะเป็นตัวนำชุดกระทำการไปที่ตำแหน่งต่างๆในที่ว่าง โดยปกติจะเป็นสามชุดแรก (นับจากฐานตัวหุ่นยนต์ขึ้นมา)

2. ชุดเชื่อมต่อรอง คือจุดหมุนและแกนเชื่อมต่อ ที่ใช้ในการหาตำแหน่งอย่างละเอียด

3. ตัวกระทำการปลายสุด จะมีความสามารถในการนำฐานยึดเครื่องมือ และตัวปฏิบัติการปลายสุดเข้าไปยังจุดทำงาน เมื่อแกนเชื่อมต่อหลักเคลื่อนเข้ามาใกล้กับตำแหน่งที่ต้องการทำงานในระดับหนึ่ง ตัวปฏิบัติการปลายสุดอาจเป็นเครื่องมือที่ใช้เชื่อมหรือใช้เจาะ หรือนิ้วหยิบจับเคลื่อนย้ายชิ้นงานไปตำแหน่งต่างๆก็ได้

2.1.1.2 อุปกรณ์ตรวจจับ (Sensory devices) คืออุปกรณ์ที่จะทำให้ทราบว่าสภาวะการในขณะนั้นว่าเป็นเช่นไร อยู่ในสถานะที่เราต้องการหรือไม่ซึ่งจะกลายเป็นข้อมูลในการควบคุมให้ตัวปฏิบัติการทำงานได้ถูกต้องตามความต้องการ

2.1.1.3 ตัวควบคุม (Controller) คือ ภูมิปัญญาหรือสมองของตัวปฏิบัติการที่จะสั่งให้ ตัวปฏิบัติการทำงานตามความต้องการของผู้ใช้ ซึ่งประกอบไปด้วย

1. ส่วนความจำเพื่อเก็บข้อมูลตำแหน่ง ที่แขนได้เคลื่อนที่ไป และข้อมูลอื่นๆที่เกี่ยวข้องกับลำดับขั้นการทำงานของระบบหุ่นยนต์

2. ส่วนจัดลำดับขั้นการทำงาน ใช้แปลความหมายของข้อมูลที่เก็บ อยู่ในหน่วยความจำและทำข้อมูลนั้นให้อยู่ในรูปที่อุปกรณ์ต่างๆ สามารถนำไปใช้เชื่อมต่อกับชุดควบคุมได้

3. หน่วยทางคณิตศาสตร์ จะจัดการคำนวณทางคณิตศาสตร์ที่จำเป็น เพื่อช่วยในการทำงานของตัวจัดลำดับการทำงาน

4. ส่วนเชื่อมข้อมูลการตรวจจับ เพื่อป้อนให้กับตัวจัดลำดับการทำงาน

5. หน่วยเชื่อมต่ออุปกรณ์ร่วมงาน เป็นการเชื่อมต่อเพื่อโอนถ่ายข้อมูลจากตัวจัดลำดับการทำงาน ให้กับหน่วยแปลงกำลังงาน เพื่อทำให้จุดหมุนเคลื่อนที่ไปได้ตามความต้องการ

6. หน่วยเชื่อมต่อกับอุปกรณ์หลักอื่นๆ ในระบบ ชุดควบคุมหุ่นยนต์ สามารถทำงานพ้องจังหวะกับชุดควบคุมอุปกรณ์ภายนอกได้

7. หน่วยควบคุมอีกบางชนิด สำหรับผู้ใช้ที่ใช้แสดงตำแหน่ง หรือจุด เพื่อบ อกลำดับการปฏิบัติงานในการควบคุมระบบหุ่นยนต์ รูปแบบการควบคุมอาจอยู่ในลักษณะเป็นแผงควบคุมที่มีฟังก์ชันการควบคุมแบบคงที่ แผงป้อนโปรแกรมภาษาควบคุม และแป้นสอนการทำงาน หรืออุปกรณ์คล้ายคลึงกัน ที่มีเมนูชุดคำสั่งสำหรับให้ผู้ปฏิบัติงาน สามารถโปรแกรมหุ่นยนต์ได้

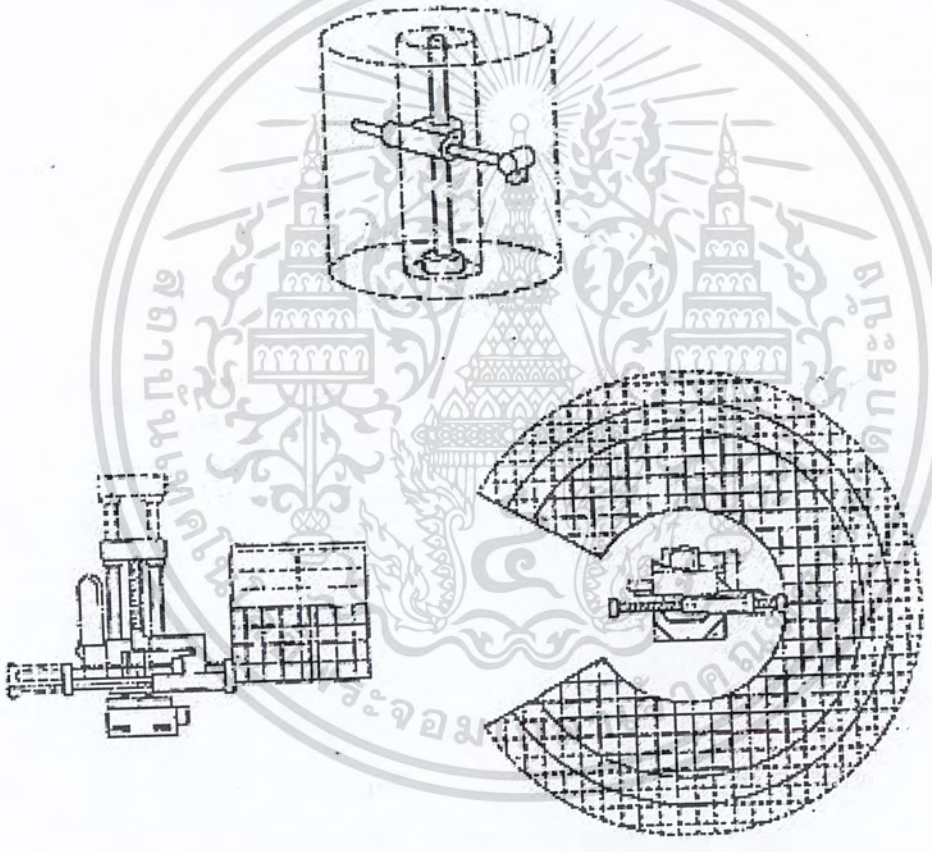
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1.4 หน่วยแปลงกำลัง (Power conversion unit) คือหน่วยที่ประกอบด้วยอุปกรณ์ที่จำเป็นการนำสัญญาณจากหน่วยจัดลำดับขั้นการทำงาน ซึ่งอาจเป็นสัญญาณดิจิตอลหรือสัญญาณอนาลอกระดับต่ำ แล้วแปลงให้สูงขึ้นเพียงพอที่จะไปขับเคลื่อนตัวกระทำการได้ เช่น ตัวขยายสัญญาณกำลังอิเล็กทรอนิกส์ คอมเพรสเซอร์ ฯลฯ เป็นต้น

2.1.2 การกำหนดชนิดของหุ่นยนต์อุตสาหกรรม กำหนดตามคำแนะนำของ ENGELBERGER แบ่งได้ 2 ชนิดคือ

2.1.2.1 แบ่งตามรูปแบบทางกลไก จำแนกได้เป็น

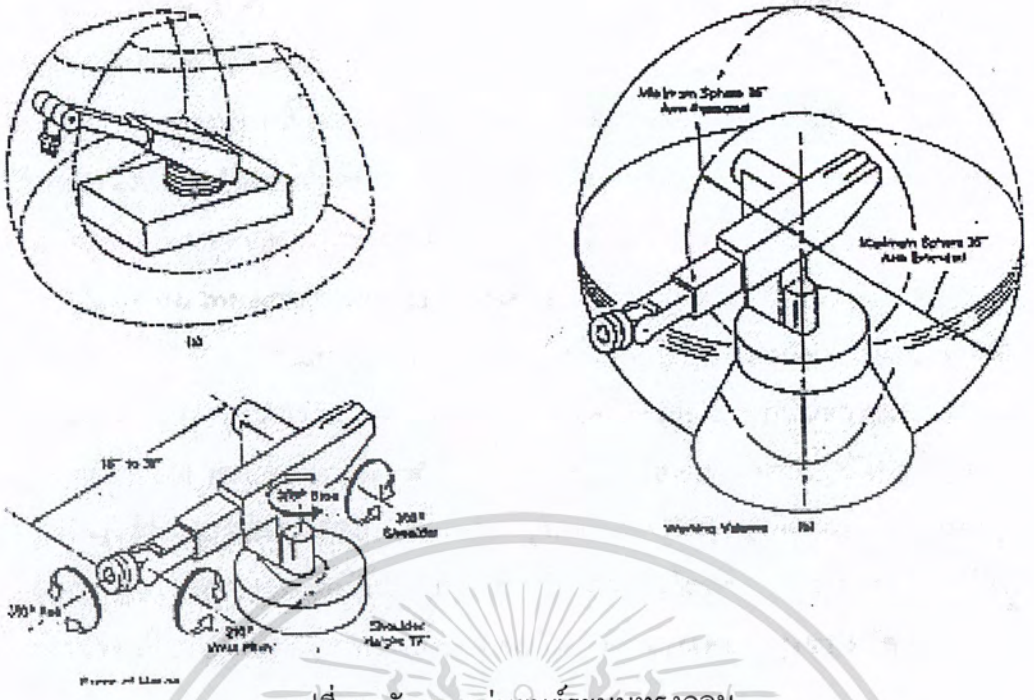
1. หุ่นยนต์ระบบทรงกระบอก คือหุ่นยนต์ที่แขนในแนวนอนถูกยึดติดกับแกนในแนวตั้ง และแกนนี้ถูกยึดกับฐานหมุนไปมาได้ ดังรูป 2.1



รูปที่ 2.1 ลักษณะหุ่นยนต์ระบบทรงกระบอก

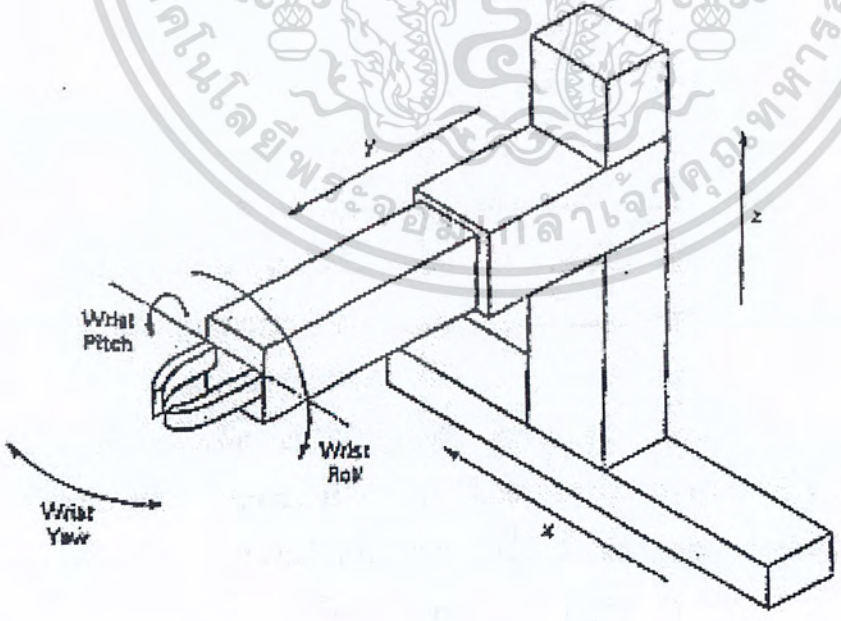
2. หุ่นยนต์ระบบทรงกลม คือหุ่นยนต์ที่แขนของหุ่นยนต์เคลื่อนที่ เข้า – ออก ในแนวรัศมีได้ตามแนวนอนซ้ายขวา และแขนทั้งชุดหมุนได้ตามแนวตั้งขึ้นลง สามารถแสดงได้ดังรูปที่ 2.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 ลักษณะหุ่นยนต์ระบบทรงกลม

3. หุ่นยนต์ระบบทรงลูกบาศก์ คือ หุนยนต์ที่ส่วนปฏิบัติการจะเคลื่อนที่ในแนวเส้นตรง โดยพื้นที่ปฏิบัติการจะมีลักษณะ เป็นรูปสี่เหลี่ยมตามลักษณะแนวแกนการเคลื่อนที่ ซึ่งมี 2 แบบ คือ แบบคานเคลื่อนที่ และ แบบ แกนทรี-ไสตล์ (Gantry – Style)



รูปที่ 2.3 ลักษณะหุ่นยนต์ทรงลูกบาศก์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2.2 แบ่งประเภทตามรูปแบบทางการควบคุม จำแนกได้เป็น

1. หุ่นยนต์ควบคุมแบบไม่ใช้เซอร์โวมอเตอร์ หมายถึง เมื่อตัวปฏิบัติการเริ่มเคลื่อน มันเคลื่อนต่อไปจนกระทั่งถึงจุดที่ควรจะหยุด จะไม่มีการตรวจสอบใดๆ ในการเคลื่อนที่ ที่ระหว่างจุดสองจุดที่กำลังเคลื่อน วิธีการเคลื่อนที่ลักษณะนี้บางทีเรียกว่า ชนิดแบบลูป-เปิด

2. หุ่นยนต์ควบคุมแบบใช้เซอร์โวมอเตอร์ หมายถึง สามารถสั่งการให้ตัวปฏิบัติการเคลื่อนที่ หรือหยุดได้ทุกตำแหน่งตลอดย่านระยะของแกนแนวปฏิบัติการ เป็นลูป-ปิด แบ่งได้ 2 แบบ คือ แบบเคลื่อนที่แบบต่อเนื่อง และ แบบเคลื่อนที่แบบจุดต่อจุด

2.1.3 คุณลักษณะเฉพาะของระบบหุ่นยนต์

1. ความสามารถในการรับภาระโหลด (Load-caring Capacity) ความสามารถในการรับโหลด หรือคุณลักษณะเฉพาะในการรับโหลด มิใช่ข้อกำหนด ในความสามารถในการรับน้ำหนัก เพิ่มขึ้นของตัวปฏิบัติการซึ่งกำลังรับน้ำหนักของเครื่องมือ หรือ ตัวปฏิบัติปลายสุดอยู่แล้ว เป็นเรื่องที่ผู้ใช้ต้องทราบเองว่าต้องการแรงหน่วงสูงสุดเท่าใดที่ตัวปฏิบัติการปลายสุดของหุ่นยนต์ที่ตัวเองต้องการใช้ที่เป็นเช่นนี้ ก็เนื่องมาจากว่าเหตุผลที่แรงหน่วงมีผลกระทบถึงทอร์คที่ต้องใช้ในการเพิ่มอัตราเร่ง เป็นไปได้ว่าผู้ใช้อาจต้องาการเคลื่อนที่โหลดที่มีแรงหน่วงสูงถ้าอัตราเร่งลดลง

2. อัตราการทำซ้ำ (Repeatability) อัตราการซ้ำตำแหน่งเดิมของหุ่นยนต์ไม่ใช้ความแม่นยำ เป็นการวัด ความสามารถของตัวปฏิบัติการในการกลับมายังตำแหน่งเดิมที่เคยอยู่ว่าแน่นอนเพียงใด เป็นการวัดการเข้าสู่ตำแหน่งพิกัดเดิมด้วยวิธีการเดิมทุกประการ ไม่ว่าจะเป็นเส้นทางที่ใช้ น้ำหนัก ความเร็ว อัตราเร่ง อุณหภูมิ ฯลฯ โดยทำซ้ำหลายๆครั้ง เนื่องจากถูกออกแบบ ให้มีค่าความหน่วงต่ำ ตัวปฏิบัติการจึงมักสั่น กระตุก ที่ตำแหน่งหยุดทำงานจึงจำเป็นต้องหยุดรอสักระยะเวลาหนึ่ง ก่อนทำการวัดค่าอัตราการทำซ้ำ

3. ความเร็วสูงสุดที่ปลายแขนในสภาพไม่มีโหลด (No load maximum tip speed) ความเร็วสูงสุดที่ปลายแขนเมื่อไม่มีโหลด คือความเร็วในการเคลื่อนของตัวปฏิบัติโดยนัยของความหมายนี้เป็นที่แน่นอนว่าจะต้องแตกต่างจากความเร็วเมื่อมีโหลด

4. ระบบพิกัดตำแหน่ง (Coordinate System) รูปแบบพิกัดตำแหน่งของหุ่นยนต์แบ่งได้เป็น 3 ชนิด คือ แบบทรงกระบอก แบบทรงกลม และแบบลูกบาศก์ ทั้ง 3 ชนิดจะมีโครงสร้างและคุณสมบัติเฉพาะงานแตกต่างกันขึ้นอยู่กับความต้องการของผู้ใช้

5. ความเร็วสูงสุดในการเคลื่อนที่ (Maximum movement) จะหมายถึงความเร็วที่ปลายแขนปฏิบัติงาน อาจมีข้อเพิ่มเติมคือ กำหนดให้ข้อต่อไม่เคลื่อนไหวยึดติดอยู่กับที่ได้ ด้วยขีดจำกัดลักษณะนี้ ทำให้ปลายแขนของหุ่นยนต์ไม่สามารถเคลื่อนที่เข้าไปยังตำแหน่งบางตำแหน่งของพื้นที่ปฏิบัติได้ เพราะข้อจำกัดทางโครงสร้างของตัวหุ่นยนต์เอง

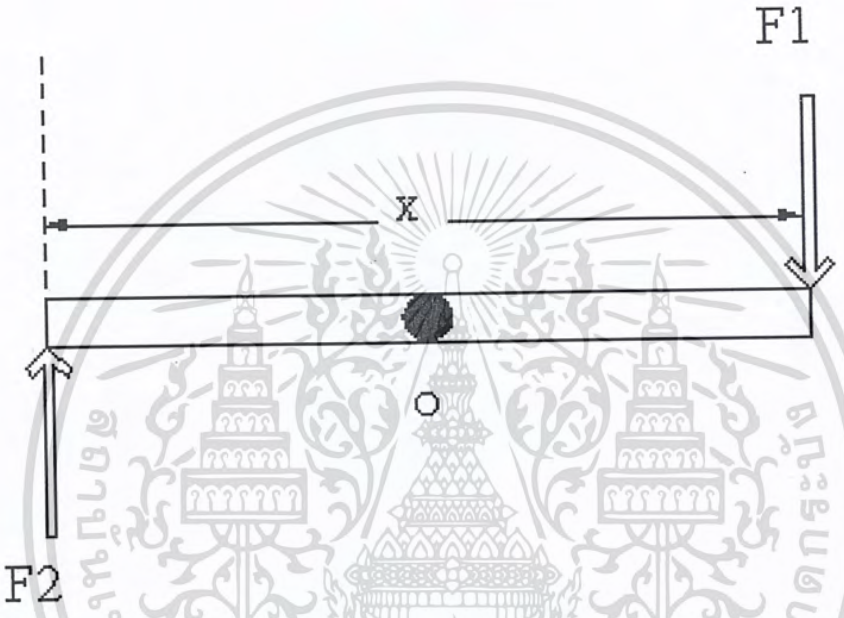
6. ชนิดของชุดต้นกำลัง (Types of drivers) หมายถึงชุดต้นกำลังสำหรับขับเคลื่อนข้อต่อของตัวปฏิบัติการมี 3 แหล่ง คือ ทางไฟฟ้า ไฮโดรลิก และ นิวเมติก โดยระบบไฮโดลิก มีคุณสมบัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการรับแรงได้ดีกว่าแต่ต้องระวังการรั่วไหลของน้ำมัน ซึ่งอาจนำไปใช้ในอุตสาหกรรมอาหาร หรืออุตสาหกรรมอิเล็กทรอนิกส์ น้ำมันไฮโดรลิก จะส่งผลร้ายต่อแผ่นวงจรพิมพ์ในช่วงบัดกรีล

7. วิธีการควบคุม (Controlling methods) หมายถึงวิธีการที่ใช้ควบคุมแกนการเคลื่อนที่มีอยู่ 2 วิธี คือแบบลูป-ปิด และ แบบลูปเปิด ซึ่งก็คือการเคลื่อนที่แบบจุดต่อจุด หรือเคลื่อนที่ต่อเนื่องตลอดเส้นทาง

2.1.4 โมเมนต์ (Moment) เมื่อมีแรงหลายแรงมากระทำต่อวัตถุโดยแนวแรงเหล่านั้นไม่ตัดกัน วัตถุอาจเกิดการหมุนขึ้นได้ ถึงแม้ว่าตำแหน่งของมันไม่เปลี่ยนแปลงก็ตาม



รูปที่ 2.4 แสดงโมเมนต์

รูปที่ 2.4แสดงให้เห็นถึงแรง F1 และ F2 ซึ่งเป็นแรงสองแรงที่มีขนาดเท่ากันแต่ทิศทางตรงกันข้าม แรงทั้งสองนี้กระทำต่อคานที่ปลายทั้งสองข้าง โดยแรงF1กระทำที่ปลายขวา และแรง F2 กระทำที่ปลายซ้าย ผลของแรงทั้งสองนี้จะทำให้คานหมุนรอบจุด O ซึ่งเป็นจุดศูนย์กลางมวลของคาน แต่จุด O ไม่มีการเคลื่อนที่ หรือกล่าวได้ว่า จุด O หมุนอยู่กับที่ เรียกสภาพของคานขณะนี้ว่า “สภาพที่สมดุลต่อการเปลี่ยนตำแหน่ง แต่ไม่สมดุลต่อการหมุน”

ได้มีการกำหนดปริมาณเวกเตอร์ที่เรียกว่า “โมเมนต์” ขึ้นเพื่อใช้ในการอธิบายผลหมุนอันเนื่องมาจากแรงโดยกำหนดขนาดของโมเมนต์ไว้ว่า โมเมนต์ของแรงที่กระทำต่อวัตถุมีค่าเท่ากับผลคูณระหว่างขนาดของแรงกับระยะทางตั้งฉากจากจุดหมุนถึงแนวแรง ส่วนทิศทางของโมเมนต์ก็คือ ทิศทางของการหมุนว่าเป็นทิศตามเข็ม หรือ ทิศทวนเข็มนาฬิกา โมเมนต์นี้มีชื่อเรียกอีกชื่อหนึ่งว่า ทอร์ก (torque) ในระบบ SI โมเมนต์มีหน่วยเป็น นิวตัน-เมตร

วัตถุที่ได้ชื่อว่าอยู่ในสภาพสมดุลต่อการหมุน คือ วัตถุที่มีโมเมนต์ตามเข็มนาฬิกาเท่ากับวัตถุที่มีโมเมนต์ทวนเข็มนาฬิกา วัตถุที่สมดุลต่อการหมุนไม่จำเป็นต้องสมดุลต่อการเปลี่ยนตำแหน่ง และวัตถุที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

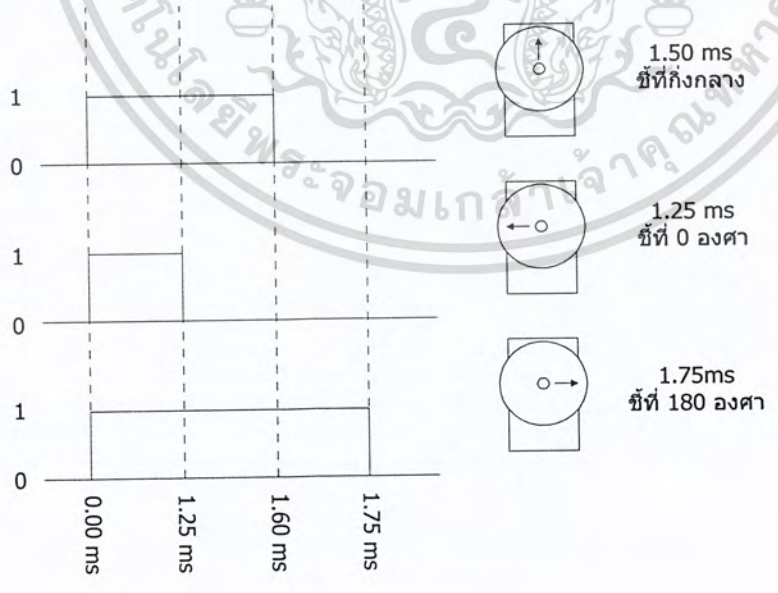
สมดุลต่อการเปลี่ยนตำแหน่ง ก็ไม่จำเป็นต้องสมดุลต่อการหมุนเช่นกัน วัตถุใดที่สมดุลต่อการหมุนและสมดุลต่อการเปลี่ยนตำแหน่งเราเรียกว่าเป็นวัตถุที่อยู่ในสมดุลสมบูรณ์ (absolute equilibrium)

2.2 อุปกรณ์และหน้าที่การทำงานของอุปกรณ์

2.2.1 เซอร์โวมอเตอร์ (Servo Motor)

เซอร์โวมอเตอร์ เป็นมอเตอร์ทดเฟืองขนาดเล็กโดยมีแกนส่งกำลัง 1 อัน ปกติเซธร์โวมอเตเตอร์จะหมุนได้ 180 องศาหรือบางตัวอาจจะได้ถึง 210 องศา แกนส่งกำลังนี้สามารถที่จะควบคุมทิศทางที่หันไปหรือตำแหน่งที่ต้องการได้ โดยการส่งรหัสไปควบคุมให้กับตัวเซอร์โวมอเตอร์ ตำแหน่งของแกนส่งกำลังจะอยู่มุมเดิมได้นั้น จะต้องมีการส่งสัญญาณเข้าที่ขาอินพุทตลอดเวลา เซอร์โวมอเตอร์ สามารถนำไปประยุกต์ใช้งานกับอุปกรณ์ได้หลากหลายส่วนใหญ่เราจะเห็นอยู่กับเครื่องเล่นที่บังคับด้วยวิทยุทั้งหลาย เช่น นำมาบังคับเลี้ยวและควบคุมอัตราวิ่งของรถบังคับวิทยุควบคุมปีกเครื่องบิน เป็นต้น

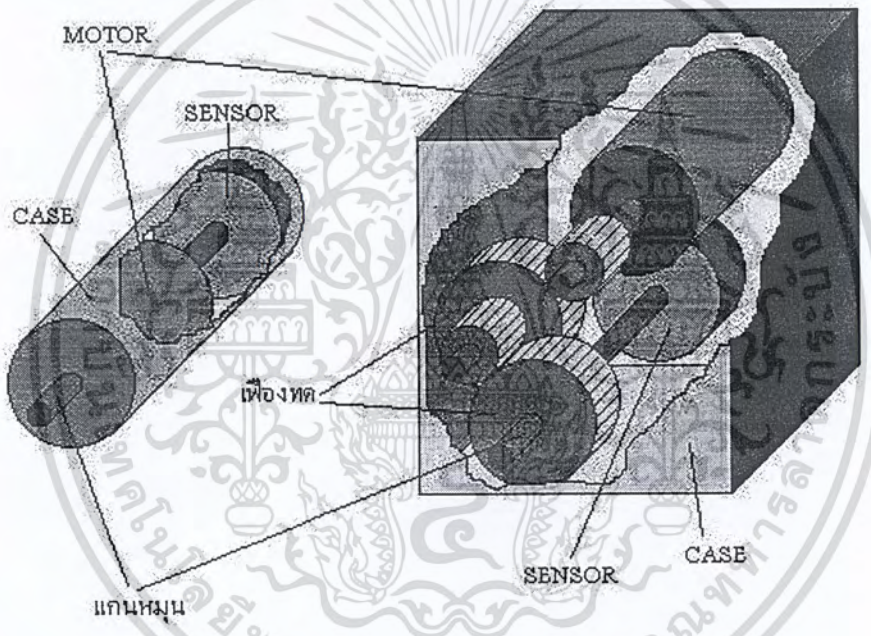
การทำงานของเซอร์โวมอเตอร์ทำได้โดย เซอร์โวมอเตอร์มีวงจรควบคุมการหมุนโดยใช้ตัวต้านทานปรับค่าได้ หรือ VR 5k ตัวต้านทานนี้จะติดอยู่กับแกนส่งกำลังเพื่อใช้วัดระยะองศาของแกนหมุนโดยใช้ร่วมวงจรควบคุมการกำหนดมุมของเซอร์โวมอเตอร์ถ้าแกนอยู่ในมุมที่ถูกต้องและจะหยุดตรงนั้นตลอดไป จนกว่าจะมีการป้อนข้อมูลเข้ามาใหม่ตัวแกนเซอร์โวมอเตอร์นี้สามารถหมุนได้เพียง 180-210 องศา (เนื่องจาก VR หมุนได้เท่านั้น) และสามารถทำให้เซอร์โวทำการกำหนดองศาของการหมุนได้โดยการส่งสัญญาณพัลซ์ค่าระหว่าง 1.5 มิลลิวินาที ถึง 2 มิลลิวินาที ดังรูปเป็นการแสดงลักษณะความกว้างของพัลซ์กับทิศทางที่เซอร์โวมอเตอร์หมุนไป



รูปที่2.5 แสดงคาบเวลาของพัลซ์ที่เป็นตัวกำหนดองศาของเซอร์โวมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป สรุปได้ว่าถ้าเราส่งพัลซ์ขนาด 1.5 มิลลิวินาทีจะทำให้เซอร์โวปรับมุมไปที่ตำแหน่ง90 องศา ส่วนถ้าป้อนสัญญาณพัลซ์น้อยกว่า 1.5 มิลลิวินาที เซอร์โวมอเตอร์จะหมุนไปทางซ้ายหรือเขาสู่ศูนย์ และถ้าป้อนพัลซ์มากกว่า 1.5 มิลลิวินาที ก็จะทำให้เซอร์โวมอเตอร์หมุนไปทางขวาหรือเข้าสู่องศาที่180 นั้นเอง สัญญาณพัลซ์ที่ส่งก็คือ สัญญาณ Pulse Width Modulation (PWM) นั่นเอง และเพื่อที่จะทำให้เราได้เข้าใจว่าเหตุใดเซอร์โวมอเตอร์จึงสามารถทำงานได้แบบดิจิตอล(Digital) ดั้งนั้นเราจึงต้องเข้าใจส่วนประกอบภายในของเซอร์โวมอเตอร์เสียก่อน เซอร์โวมอเตอร์ซึ่งจะมีทั้งตัวต้นกำลังและเซนเซอร์(Sensor) อยู่ในชุดเดียวกันโดยตัวต้นกำลังก็คือ มอเตอร์ และ เซนเซอร์นั้นจะต่อแกนร่วมเดียวกันกับแกนของ มอเตอร์(ดังรูปที่ 2.6) แต่ใน เซอร์โวมอเตอร์ บางชนิดก็อาจต่อแกนร่วมเข้ากับแกนของเฟืองทดของ มอเตอร์ (ดังรูปที่2.7)



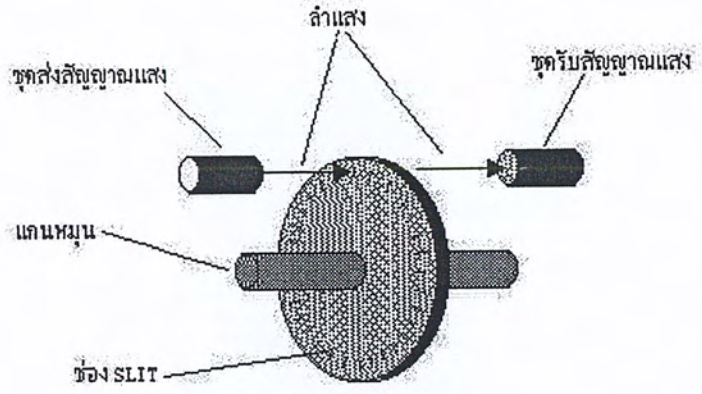
รูปที่ 2.6 รูปที่ 2.7

รูปที่2.6 รูปแสดงเซนเซอร์ต่อแกนร่วมเดียวกันกับแกนของมอเตอร์

รูปที่2.7 รูปแสดงเฟืองทดต่อแกนร่วมเข้ากับแกนของเฟืองทดของ มอเตอร์

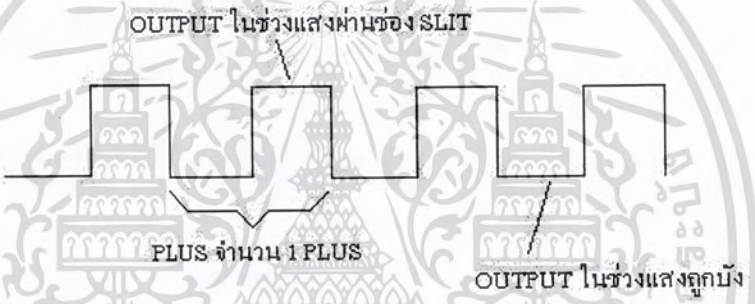
เซอร์โวมอเตอร์ ในรูปที่ 2.6 นั้นจะใช้ sensor ที่เรียกว่า rotary encoder มีลักษณะเป็นแผ่นกลมมีแกนอยู่ตรงกลางและ ที่แผ่นกลม จะมีช่องเล็ก ที่แสงสามารถ ส่องผ่านได้ เป็นจำนวนมากเราเรียกช่องนี้ว่า ช่อง slit ซึ่งที่ด้านหนึ่งของ แผ่นกลม นี้ จะมีตัวส่งแสง อินฟาเรด(Infared) ไปยังตัวรับสัญญาณแสงอินฟาเรดซึ่งจะอยู่ในด้านตรงกันข้าม ดังรูปที่ 2.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่2.8แสงตัวส่งแสง อินฟาเรดไปยังตัวรับสัญญาณแสง อินฟาเรดซึ่งจะอยู่ในด้านตรงกันข้าม

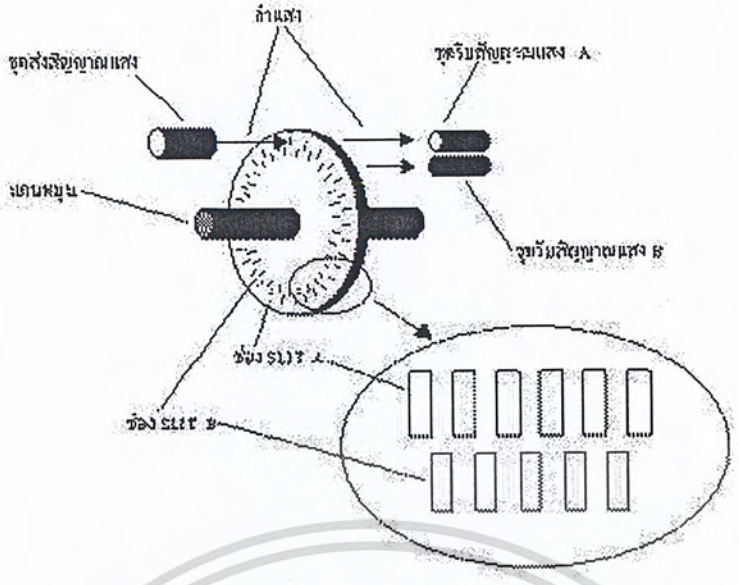
เมื่อหมุนแกนหมุนทำให้แผ่นกลมหมุนไปตัดลำแสง อินฟาเรดดังนั้นชุดรับแสง อินฟาเรด จึงมีแสงมากระทบเป็นช่วงๆ (เป็น พัลซ์) ตามจังหวะที่แสงผ่านช่อง (Slit) จึงทำให้สัญญาณ เอาท์พุท ของชุดรับแสง อินฟาเรดมีลักษณะเป็น พัลซ์ ดังรูปที่ 2.9



รูปที่2.9แสดงสัญญาณ เอาท์พุท ของ ชุดรับแสง อินฟาเรดมีลักษณะเป็น พัลซ์

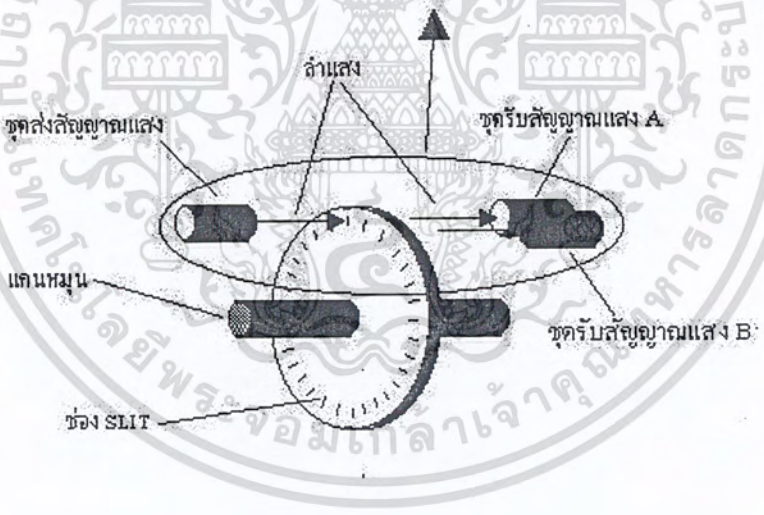
จำนวน พัลซ์ ที่ได้ออกมานี้ จะเป็นตัวที่ชี้บ่งว่า มอเตอร์ หมุนไปกี่องศา หรือกี่รอบ ซึ่งเราสามารถคำนวนได้จาก สูตร จำนวนรอบที่ มอเตอร์ หมุนไป เท่ากับ จำนวนพัลซ์ค่าความละเอียดของ โรตารี่เอนโคดเดอร์(Rotary Encoder) โดยที่ ค่าความละเอียดของ โรตารี เอนโคดเดอร์นั้นนิยมใช้หน่วยเป็น พัลซ์ / รอบ หรือ PPR เช่น 1000 PPR ก็หมายถึงว่าเมื่อ มอเตอร์หมุนไป 1 รอบจะมี พัลซ์ ออกมา 1000 พัลซ์ เป็นต้นส่วนในเรื่องที่ว่าเราจะทราบได้อย่างไรว่ามอเตอร์ หมุนไปทิศทางใดนั้นส่วนใหญ่แล้วจะใช้วิธีสร้าง ช่อง เป็น 2 ชุดเหลื่อมกัน 90 องศา ดังรูปที่ 2.10และ รูปที่ 2.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10แสดงวิธีสร้าง ช่อง เป็น 2 ชุดเหลื่อมกัน 90 องศา

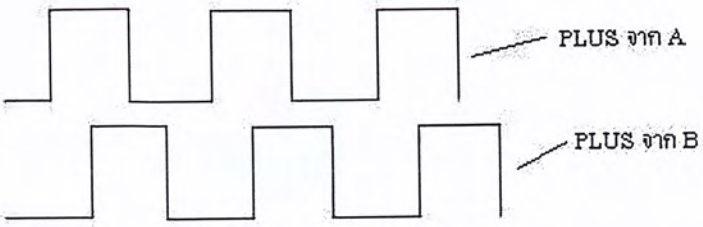
หรืออาจจะใช้ช่องเพียง1ชุดแต่มีการจัดวางชุดรับสัญญาณแสงดังรูปข้างล่างแต่ข้อสำคัญคือจะต้องมี เฟส ต่างกัน 90



รูปที่ 2.11แสดงวิธีสร้าง ช่อง เป็น 2 ชุดเหลื่อมกัน 90 องศา

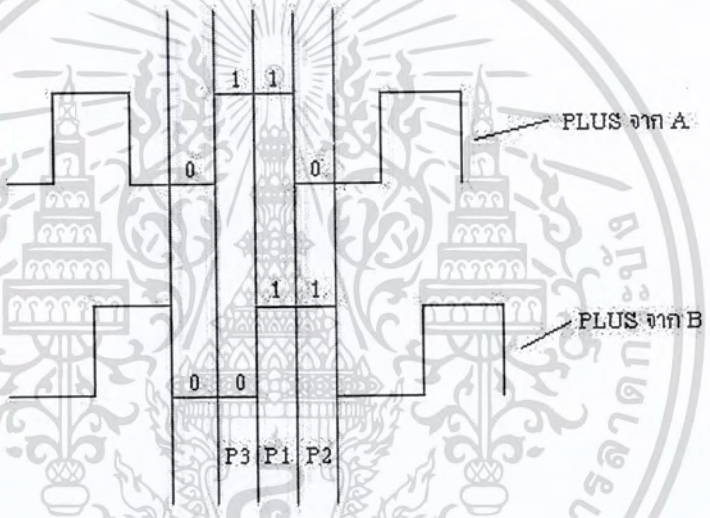
ดังนั้นสัญญาณ เอาท์พุท จาก เซอร์โวมอเตอร์ จึงมี 2 ชุด คือ A และ B โดยที่สัญญาณ พัลซ์ จาก A และ Bก็จะเหลื่อมกัน 90 องศาด้วย อาจกล่าว ได้ว่าสัญญาณ เอาท์พุท จาก เซอร์โวมอเตอร์ ค่าเท่ากับ 2 บิท คือหนึ่งบิท มาจาก A และอีกหนึ่งบิท มาจาก B ดังรูปที่ 2.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.12 แสดงสัญญาณ เอาท์พุท จาก เซอร์โวมอเตอร์ ค่าเท่ากับ 2 บิท

ถ้าเราให้ พัลซ์ ในช่วง สูง มีค่าเป็น “1” และ พัลซ์ ในช่วง ต่ำ มีค่าเป็น “0”เราสามารถใช้ค่าดังกล่าวมาคำนวณหาทิศทางที่ เซอร์โวมอเตอร์ หมุนได้ด้วยใช้วิธีการทางดิจิตอล คือการนำค่าที่อ่านได้มาทำการ Excusive OR (XOR) กันโดยการนำบิททางขวาของค่าเก่ามา XOR กับ บิท ทางซ้ายของค่าใหม่ที่อ่านได้ตามตัวอย่างข้างล่าง



รูปที่ 2.13 แสดงตัวอย่างสัญญาณ เอาท์พุท จาก เซอร์โวมอเตอร์ ค่าเท่ากับ 2 บิท

สมมุติในตอนแรก เซอร์โวมอเตอร์ อยู่ที่ตำแหน่ง P1 เพราะฉะนั้นค่าของตัวเลขฐานสอง ที่อ่านได้จาก เอนโคดเดอร์ จะมีค่าเป็น 11 ถ้า เซอร์โวมอเตอร์ หมุนไปที่ตำแหน่ง P3 จะทำให้ค่าของตัวเลขฐานสองที่อ่านได้จากเอนโคดเดอร์ จะมีค่าเป็น 10 ดังนั้นถ้าเรานำเอาค่าทางขวาจากค่าที่อ่านได้ในครั้งแรก (11) ซึ่งก็คือ 1 มา XORกับค่าทางซ้ายที่อ่านได้ในครั้งต่อมา (10) ซึ่งก็คือ 1 จะได้ผลลัพธ์เท่ากับ 0 เพราะฉะนั้นค่า 0 จึงเป็นค่าที่บอกว่า เซอร์โวมอเตอร์ หมุนไปทางขวาในทางกลับกันถ้าในตอนแรก เซอร์โวมอเตอร์ อยู่ที่ตำแหน่ง P1และซึ่งก็คือ 1 มา XOR กับค่าทางซ้ายที่อ่านได้ในครั้งต่อมา (01) ซึ่งก็คือ 0 ต่อมา เซอร์โวมอเตอร์หมุนไปที่ตำแหน่ง P2 ถ้าเรานำเอา ค่าทางขวา จากค่าที่อ่านได้ในครั้งแรก (11)จะได้ผลลัพธ์เท่ากับ 1 เพราะฉะนั้นค่า 1 จึงเป็นค่าที่บอกว่าเซอร์โวมอเตอร์หมุนไปทางซ้าย และการ XOR ค่าต่างๆได้แสดงไว้ดังตารางที่ 2.1

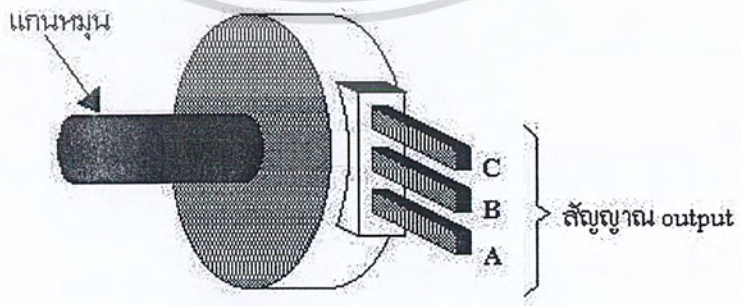
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A	B	AX OR B
0	0	0
0	1	1
1	0	1
1	1	0

ตารางที่ 2.1 แสดงการ XOR ด้วยค่าต่างๆ

พัลซ์ ที่ได้จาก โรตารี่ เอนโคดเดอร์ จะถูกนำไปเชื่อมต่อเข้ากับวงจรนับพัลซ์ หรือวงจรคำนวน เพื่อที่จะตรวจสอบว่าเซอร์โวมอเตอร์หมุนไปทิศทางใดและอยู่ ที่ตำแหน่งใด ตามวิธีที่ได้กล่าวไว้ข้างต้นโดยส่วนมากแล้ว เซอร์โวมอเตอร์ ที่ใช้โรตารี่ เอนโคดเดอร์ แบบนี้ จะมีจำนวนรอบ ของการหมุน ที่ไม่จำกัด โดยจะหมุนไปกี่รอบ ก็ได้ ขึ้นอยู่ กับ ความสามารถของวงจรนับพัลซ์ และวงจรคำนวน ยังมีเอนโคดเดอร์ อีกหนึ่งชนิดที่มักจะใช้กันบ่อยๆคือ เอนโคดเดอร์ แบบ โพเทนชิโอมิเตอร์(Potentiometer) หรือตัวต้านทานแบบปรับค่าได้ ซึ่งมีไดอะแกม(DIAGRAM)m แสดงดังรูปที่ 2.14a และมีรูปแสดงดังรูปที่ 2.14b

รูปที่ 2.14 a แสดงไดอะแกรมตัวต้านทานปรับค่าได้



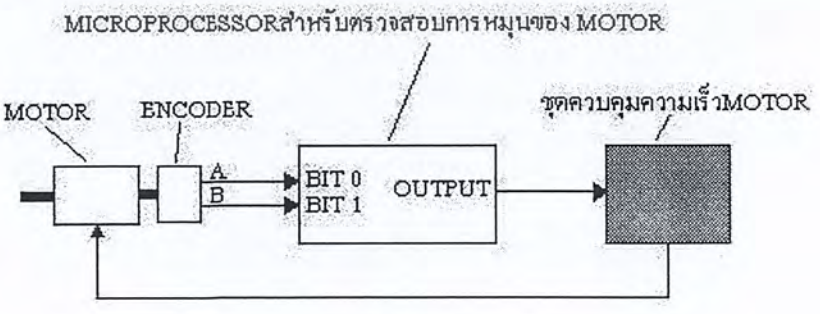
รูปที่ 2.14 b รูปแสดงตัวต้านทานปรับค่าได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เมื่อแกนหมุนถูกหมุนไม่ว่าจะเป็นในทิศทางทวนเข็มนาฬิกาหรือตามเข็มนาฬิกาก็ตามจะทำให้ค่าความต้านทานที่วัดได้ระหว่างจุด A – B และจุด B – C มีค่าเปลี่ยนไป นั่นคือถ้าค่าความต้านทานที่วัดได้ระหว่างจุด A – B มีค่ามากขึ้น ค่าความต้านทานที่วัดได้ระหว่างจุด B –C จะมีค่าน้อยลง หรือในทางกลับกัน ถ้าค่าความต้านทาน ที่วัดได้ระหว่างจุด A – B มีค่าน้อยลง ค่าความต้านทานที่วัดได้ระหว่างจุด B –C จะมีค่ามากขึ้น เป็นสัดเป็นส่วนที่แน่นอนซึ่งหมายความว่าค่าความต้านทานของจุด A – B เมื่อนำมารวมกับ ค่าความต้านทานของจุด B – C จะมีค่าเท่าเดิมเสมอ ดังนั้นค่าความต้านทานระหว่างจุด A – B และจุด B – C นี้เองจะเป็นตัวที่บ่งชี้ถึงตำแหน่งของ เซอร์โวมอเตอร์ แต่อย่างไรก็ตาม เซอร์โวมอเตอร์ที่ใช้ เอนโคดเดอร์แบบนี้จะมีจำนวนรอบของการหมุนที่จำกัดโดยจะมีค่าสูงสุดเท่ากับจำนวนรอบที่ โพเทนชิโอมิเตอร์ สามารถจะหมุนได้โดยอาจจะมีตั้งแต่ 1 – 25 รอบขึ้นอยู่กับชนิดของ โพเทนชิโอมิเตอร์ที่เลือกใช้

ดังนั้นเราอาจจะกล่าวได้ว่า เอนโคดเดอร์ ที่ใช้กันอยู่โดยทั่วไปนั้นมีอยู่ 2 แบบใหญ่ๆคือเป็นแบบมีเอาท์พุทที่ไม่จำกัด(Incremental Encoder) และแบบที่มีค่าเอาท์พุทที่จำกัด (Absolute Encoder) แบบมีเอาท์พุทที่ไม่จำกัดมักจะเป็น เอนโคดเดอร์ แบบ โรตารี่ เอนโคดเดอร์ ดังที่ได้กล่าวไว้ในตอนต้นแล้วว่าขนาดของ เอาท์พุท ของ เอนโคดเดอร์ แบบนี้ขึ้นอยู่กับการออกแบบวงจรนับและวงจรคำนวณ แบบที่มีค่าอาท์พุทที่จำกัด ที่พบเห็นส่วนใหญ่จะเป็นแบบ โพเทนชิโอมิเตอร์ แต่ในบางครั้งก็อาจเป็นแบบ โรตารี่ เอนโคดเดอร์ ก็ได้แต่จะมีการออกแบบลักษณะของช่อง ที่ต่างไปจาก โรตารี่เอนโคดเดอร์ที่ได้ไว้ในตอนต้น โดยอาจจะมีค่าของ เอาท์พุท ตั้งแต่ 4 บิท ถึง 16บิทในแง่ของการใช้งานแล้ว เซอร์โวมอเตอร์ ที่มี เอนโคดเดอร์ แบบนั้นจะใช้งานยุ่งยากกว่า เซอร์โวมอเตอร์ ที่มี เอนโคดเดอร์ แบบที่มีค่าอาท์พุทที่จำกัดเพราะ เอาท์พุทแบบมีเอาท์พุทที่ไม่จำกัดนั้นไม่สามารถเชื่อมต่อกับชุดควบคุม มอเตอร์ได้ทันทีโดยต้องต่อผ่านวงจรนับพัลซ์ และวงจรคำนวนที่ยุ่งยากซับซ้อนก่อนถึงจะต่อเข้ากับกับ ชุดควบคุมได้ แต่อย่างไรก็ตาม ในปัจจุบันเรามี ไมโครโปรเซสเซอร์(MICROPROCESSOR)เช่น ไมโครโปรเซสเซอร์ ตระกูล PIC เป็นต้น ดังนั้นเราจึงสามารถนำมาใช้แทนวงจรนับพัลซ์ และวงจรคำนวณได้จึงทำให้การใช้งานมีความง่ายขึ้นดังรูปที่แสดงไว้ข้างล่างดังรูปที่ 2.15

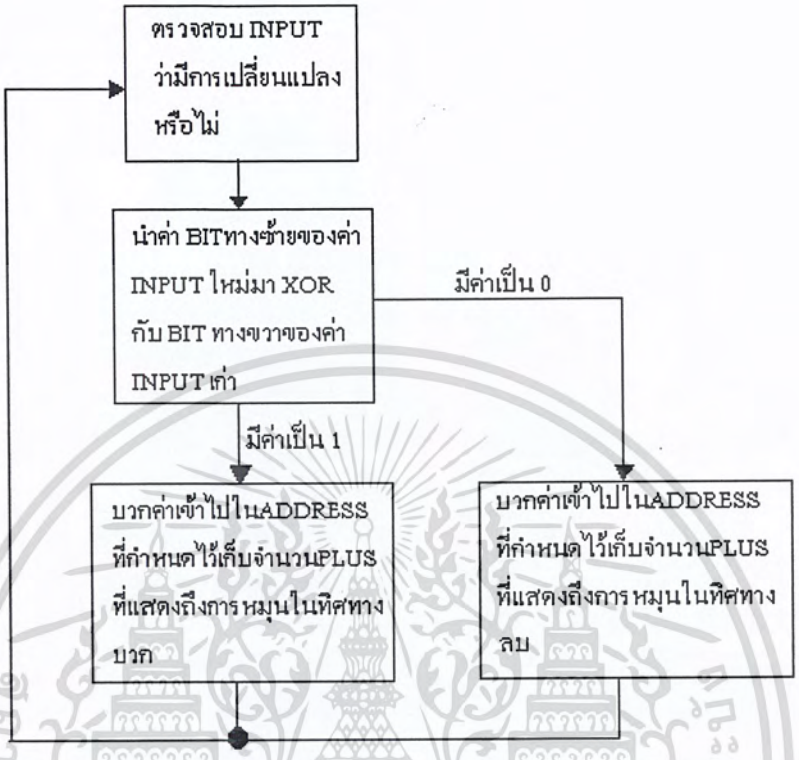


รูปที่ 2.15 แสดงการนับพัลซ์ด้วยไมโครโปรเซสเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

42685

สำหรับ FLOW CHART แสดงลำดับการทำงานของไมโครโปรเซสเซอร์ ที่ทำการตรวจสอบการหมุนของมอเตอร์ นั้นได้แสดงไว้ดังไดอะแกรมข้างล่างดังรูปที่ 2.16



รูปที่ 2.16 แสดง FLOW CHART ลำดับการทำงานของไมโครโปรเซสเซอร์ ที่ทำการตรวจสอบการหมุนของมอเตอร์

2.2.2อุปกรณ์ควบคุมไมโครคอนโทรลเลอร์ตระกูล MCS-51

ไมโครคอนโทรลเลอร์ในตระกูล MCS-51 เป็นไมโครคอนโทรลเลอร์ขนาด 8 บิท ที่มีอุปกรณ์สนับสนุนอยู่ภายในหลายอย่าง ได้แก่ หน่วยความจำสำหรับเก็บข้อมูล หน่วยความจำสำหรับเก็บโปรแกรม ตัวตั้งเวลา/ตัวนับ อุปกรณ์รับส่งข้อมูลแบบอนุกรม เนื่องจากโครงสร้างของไมโครคอนโทรลเลอร์มีอุปกรณ์สนับสนุนอยู่ภายในนี้เอง ทำให้การใช้งานง่ายขึ้นและมีประสิทธิภาพมากขึ้น โดยไม่ต้องมีการเชื่อมต่ออุปกรณ์ภายนอกเพิ่มเติมมากเหมือนกับตัวไมโครโปรเซสเซอร์ทั่วไป นอกจากนี้หากเราต้องการใช้งานไมโครคอนโทรลเลอร์ร่วมกับอุปกรณ์อื่นเพิ่มเติม เช่นไอซี 8255 หรือหน่วยความจำภายนอก เรายังสามารถนำมาเชื่อมต่อเพิ่มเติมเข้ากับไมโครคอนโทรลเลอร์ได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 มีให้เลือกหลายเบอร์ดังแสดงในตารางที่ 2.2

DEVICE NAME	EPROM	ROM BYTES	RAM BYTES	16 BIT TIMER/COUNTER	INTERRUPT
8031	-	-	128*8	2	5
8031 AH	-	-	128*8	2	5
8031 BH	-	-	128*8	2	5
8032 AH	-	-	256*8	3	6
8051	-	4K*8	128*8	2	5
8051 AH	-	4K*8	128*8	2	5
8051 BH	-	4K*8	128*8	2	5
8052 AH	-	8K*8	256*8	3	6
8751 H	4K*8	-	128*8	2	5
8752 H	8K*8	-	256*8	3	6

ตารางที่ 2.2 ไมโครคอนโทรลเลอร์ตระกูล MCS-51

2.2.2.1 โครงสร้างภายในของไมโครคอนโทรลเลอร์ตระกูล MCS-51

โครงสร้างภายในพื้นฐานของไมโครคอนโทรลเลอร์ตระกูล MCS-51 เบอร์ 8051 แสดงในรูปที่ 2.17 ประกอบด้วยอุปกรณ์ต่างๆดังนี้

-หน่วยความจำภายในสำหรับเก็บข้อมูลขนาด 128 ไบท์ (Internal Data Memory 128 Bytes)

-หน่วยความจำภายในสำหรับเก็บโปรแกรมขนาด 4 กิโลไบท์ (Internal Program Memory 4 K Bytes)

-อุปกรณ์ควบคุมการอินเตอร์รัพต์ (Interrupt Control Unit)

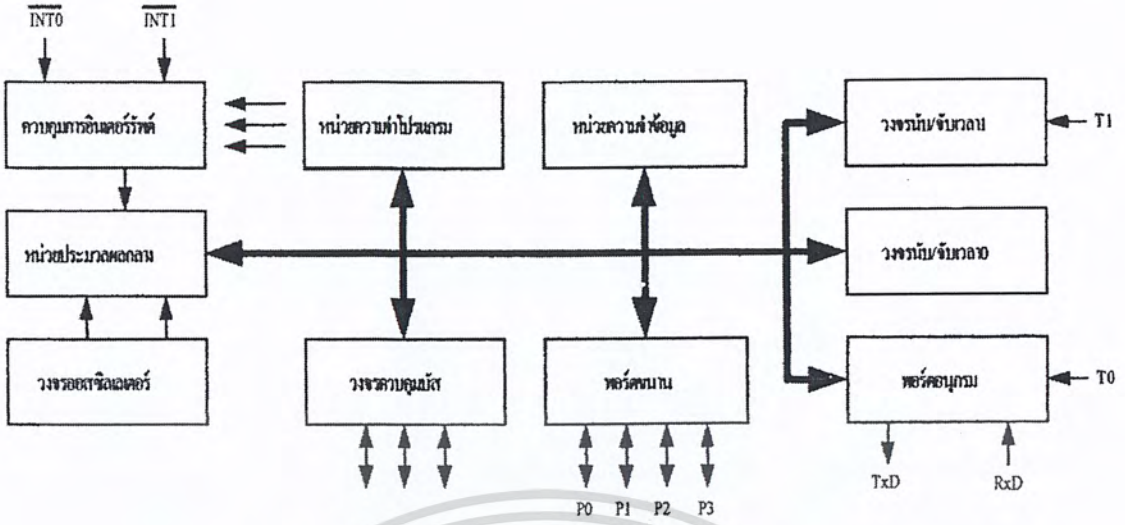
-ตัวตั้งเวลาและตัวนับขนาด 16 บิท 2 ชุด (Timer/Counter0 And Timer/Counter1)

-พอร์ตควบคุมการสื่อสารอนุกรมแบบ Full Duplex ซึ่งสามารถรับและส่งข้อมูลพร้อมกันได้

-พอร์ตขนานสำหรับติดต่อกับอุปกรณ์ภายนอกจำนวน 4 พอร์ต พอร์ตละ 8บิท

-วงจรผลิตสัญญาณนาฬิกาภายใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.17 แสดงโครงสร้างภายในของไมโครคอนโทรลเลอร์ตระกูล MCS-51

2.2.2.2 หน่วยความจำสำหรับเก็บโปรแกรม (Internal Program Memory)

หน่วยความจำสำหรับเก็บโปรแกรมทำหน้าที่เก็บ โปรแกรมที่ผู้ใช้เขียนขึ้น เพื่อควบคุมการทำงานของไมโครคอนโทรลเลอร์โดยหน่วยความจำจะเป็นแบบ Rom มีความจุ 4K Bytes (ตำแหน่ง 0000H-0FFFFH) ในการใช้งาน เราสามารถกำหนดให้ไมโครคอนโทรลเลอร์เลือกใช้โปรแกรมที่เก็บอยู่ภายในตัวไมโครคอนโทรลเลอร์ หรือ โปรแกรมที่เก็บอยู่ในหน่วยความจำ (EPROM) ที่อยู่ภายนอกก็ได้ การเลือกการติดต่อทำได้โดยป้อนสัญญาณควบคุมให้ที่ขา EA (External Access) ถ้าต้องการให้ไมโครคอนโทลเลอร์ติดต่อกับโปรแกรมที่อยู่ในหน่วยความจำภายในตัวไมโครคอนโทลเลอร์ จะต่อขานี้กับลอจิก 1 หากต้องการให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำที่อยู่ภายนอกจะต่อขานี้กับลอจิก 0 การติดต่อกับหน่วยความจำโปรแกรมภายนอกจะติดต่อได้ทั้งหมด 64K Bytes (ตำแหน่ง 0000H-0FFFFH)

ในกรณีที่กำหนดให้ไมโครคอนโทรลเลอร์ติดต่อกับ โปรแกรมที่อยู่ในหน่วยความจำภายในตัวไมโครคอนโทรลเลอร์จะติดต่อได้ 4K Bytes (สำหรับเบอร์ 8051) หากตำแหน่งของโปรแกรมมีค่าเกินกว่าตำแหน่งของหน่วยความจำภายใน (โปรแกรมยาวเกินกว่า 4K Bytes) ตัวไมโครคอนโทรลเลอร์จะติดต่อกับหน่วยความจำที่อยู่ภายนอกโดยอัตโนมัติ

2.2.2.3 หน่วยความจำข้อมูลภายใน (Internal Data Memory)

หน่วยความจำข้อมูลภายในทำหน้าที่เก็บข้อมูลทั่วไป และทำหน้าที่เป็นแสตก (Stack) บางส่วน หน่วยความจำข้อมูลภายในของเบอร์ 8051 มีอยู่ 128 ไบท์ โดยอยู่ที่ตำแหน่ง 00H-7FH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2.4 อุปกรณ์อุปกรณ์ควบคุมการอินเตอร์รัพต์ (Interrupt Control Unit)

เป็นส่วนที่ทำหน้าที่ควบคุมการอินเตอร์รัพต์ของไมโครคอนโทรลเลอร์ ซึ่งมีการร้องขออินเตอร์รัพต์ได้จาก 6 แหล่งกำเนิด คือ สัญญาณจากภายนอก 2 สัญญาณจากตัว Timer0,Timer1 และ Timer2 (เบอร์ 8051 มี Timer เพียง 2 ตัวดังนั้นจะมีแหล่งกำเนิดสัญญาณอินเตอร์รัพต์ 5 แหล่ง) และจากอุปกรณ์รับส่งข้อมูลแบบอนุกรม 1 สัญญาณ สัญญาณอินเตอร์รัพต์ที่เกิดขึ้นเราสามารถควบคุมให้ไมโครคอนโทรลเลอร์ตอบรับหรือไม่ตอบรับก็ได้ นอกจากนี้เรายังสามารถจัดลำดับความสำคัญของการอินเตอร์รัพต์ (Interrupt Priority) จากสัญญาณต่างๆ ได้เป็น 2 ระดับแตกต่างกัน

2.2.2.5ตัวตั้งเวลาและตัวนับ (Timer/Counter)

ในไมโครคอนโทรลเลอร์เบอร์ 8051 ประกอบด้วยรีจิสเตอร์ตัวตั้งเวลาตัวนับขนาด 16 บิต จำนวน 2 ชุด คือ Timer0 และ Timer1 สำหรับไมโครคอนโทรลเลอร์เบอร์ 8052 จะมี Timer2เพิ่มขึ้นอีก 1 ตัวโดย Timer ทั้งหมดสามารถกำหนดให้ทำงานในลักษณะของตัวนับ หรือตัวจับเวลาก็ได้

การทำงานในโหมดของตัวตั้งเวลา ค่าในรีจิสเตอร์จะเพิ่มขึ้นทุกๆ แมชชีนไซเกิลโดย 1 แมชชีนไซเกิลประกอบด้วยสัญญาณนาฬิกา 12 ลูก ดังนั้นอัตราการจับเวลาจะเป็น 1/12 เท่าของความถี่สัญญาณนาฬิกาของระบบ ค่าสูงสุดที่ตั้งได้ คือ 2^{16}

การทำงานในโหมดการนับ ค่าของการนับจะเพิ่มขึ้นเมื่อมีสัญญาณเข้ามาที่ขา T0 หรือ T1 เปลี่ยนจาก 1 เป็น 0 ความเร็วในการนับสูงสุดคือ 1/24 เท่าของสัญญาณนาฬิกา โดยสัญญาณที่เข้ามาที่ขา T0 หรือ T1 จะมี Duty Cycle เท่าใดก็ได้

2.2.2.6พอร์ตอินพุทเอาต์พุทในไมโครคอนโทรลเลอร์ MCS-51

ประกอบด้วยพอร์ตอินพุทเอาต์พุทแบบ 2 ทิศทาง (BIDIRECTIONAL) จำนวน 4 พอร์ต แต่ละพอร์ตมีอุปกรณ์แลทช์ข้อมูลและเอาต์พุทไดรเวอร์ประกอบอยู่ด้านเอาต์พุท และทางด้านอินพุทจะมีบัฟเฟอร์(พอร์ตทั้ง 4 เป็นรีจิสเตอร์พิเศษชื่อ P0,P1,P2 และ P3) เราสามารถใช้งานแต่ละพอร์ตเป็นอินพุทหรือเอาท์พุทได้ตามต้องการ แต่ละบิทของพอร์ตสามารถเชื่อมต่อกับสัญญาณ TTL ได้โดยตรง

ในการติดต่อกับหน่วยความจำภายนอกพอร์ต P0 และ P2 จะใช้สำหรับกำหนดตำแหน่งของหน่วยความจำภายนอก โดยพอร์ต P0 จะทำงานในลักษณะของมัลติเพล็กซ์คือเป็นทั้งพอร์ตตำแหน่งและพอร์ตข้อมูล โดย P0 จะเป็นตำแหน่งของหน่วยความจำด้านต่ำ (LOW BYTE) และ P2 จะเป็นตำแหน่งของหน่วยความจำด้านสูง(HIGH BYTE)

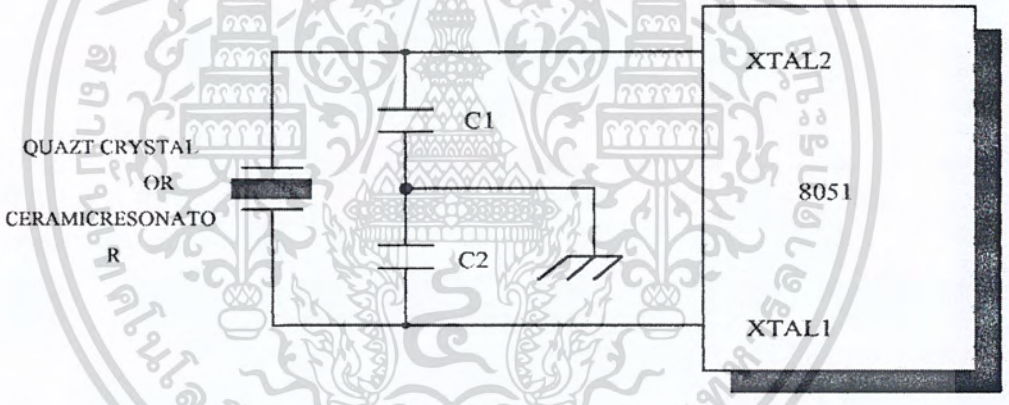
2.2.2.7 สัญญาณต่าง ๆ ของไมโครคอนโทรลเลอร์ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไมโครคอนโทรลเลอร์ในตระกูล MCS-51 เป็นไอซีขนาด 40 ขาซึ่งมีสัญญาณต่างๆ สัญญาณต่างๆ ของไมโครคอนโทรลเลอร์ MCS-51 สามารถจำแนกตามการทำงานเป็น 3 กลุ่ม คือ

- กลุ่มสัญญาณตำแหน่ง เป็นตัวกำหนดตำแหน่งของหน่วยความจำ
- กลุ่มสัญญาณควบคุมเป็นสัญญาณควบคุมการทำงานของไมโครคอนโทรลเลอร์
- กลุ่มสัญญาณข้อมูลเป็นทางผ่านของข้อมูลระหว่าง ไมโครคอนโทรลเลอร์กับหน่วย ความจำ หน้าที่และการใช้งานของสัญญาณต่างๆ เป็นดังนี้
- VCC สำหรับต่อกับไปเลี้ยง 5 โวลท์
- VSS สำหรับต่อกับกราวน์
- XTAL1 เป็นอินพุทของภาคขยายสัญญาณแบบอินเวิรส์ของวงจรผลิตสัญญาณนาฬิกา
- XTAL2 เป็นเอาท์พุทของภาคขยายสัญญาณแบบอินเวิรส์ของวงจรผลิตสัญญาณนาฬิกา

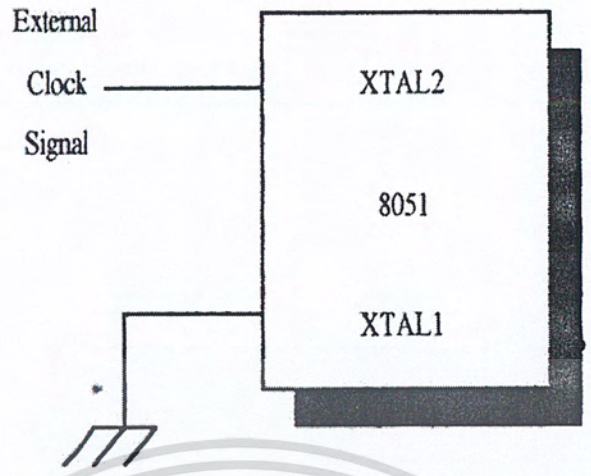
การต่อใช้งานของขา XTAL1 และ XTAL2 เพื่อสร้างวงจรผลิตสัญญาณให้กับไมโครคอนโทรลเลอร์โดยใช้คริสตอล สามารถทำได้ดังรูปที่ 2.18



รูปที่ 2.18 การต่อสัญญาณนาฬิกาจากภายนอกจะต่อ XTAL1 ลงกราวน์และต่อสัญญาณ

หากต้องการใช้สัญญาณนาฬิกาจากภายนอกจะต่อ XTAL1 ลงกราวน์และต่อสัญญาณนาฬิกาจากภายนอกเข้าที่ขา XTAL2 ดังรูปที่ 2.19

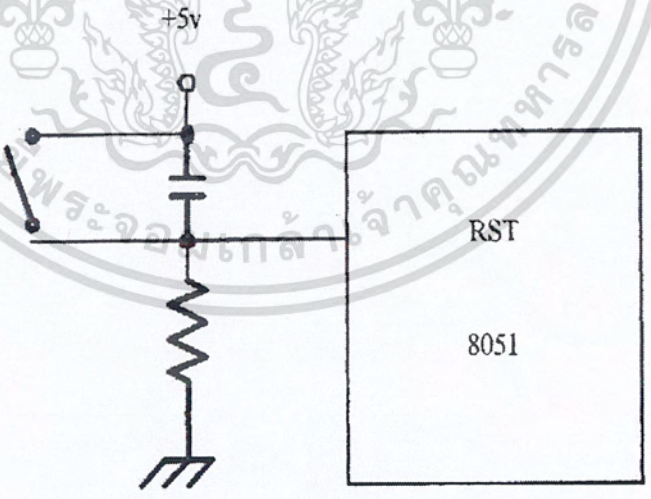
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.19 การใช้สัญญาณนาฬิกาจากภายนอก

- RST

สัญญาณรีเซตไมโครคอนโทรลเลอร์จะถูกรีเซตเมื่อสัญญาณที่ขานี้มีค่าเป็น ลอจิก 1 นานไม่ต่ำกว่า 2 แมชชีนไซเกิล การต่อขารีเซตจะเป็นดังรูป 2.20



รูปที่ 2.20 การต่อสัญญาณรีเซต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ALE/PROG : (Address Latch Enable)

เป็นเอาต์พุทซึ่งไมโครคอนโทรลเลอร์จะส่งออก ไปเป็นพัลส์เพื่อแลทช์ค่าตำแหน่งไบท์ ต่ำที่อยู่ที่พอร์ต P0 ในขณะติดต่อกับหน่วยความจำภายสัญญาณนี้จะถูกส่งออกไปด้วยอัตราคงที่คือ 1/6 เท่าของความถี่สัญญาณนาฬิกาซึ่งสามารถนำไปใช้เป็นสัญญาณนาฬิกา ให้กับอุปกรณ์ภายนอกได้สัญญาณพัลส์นี้จะถูกข้ามไปพัลส์เมื่อมีการติดต่อกับหน่วยความจำข้อมูลภายนอก (External Data Memory) และสัญญาณนี้จะใช้เป็นอินพุทเพื่อควบคุมโปรแกรม PROM ภายในตัวไมโครคอนโทรลเลอร์ด้วย

- PSEN (Program Store Enable)

เป็นเอาต์พุทสำหรับส่งสัญญาณ สโตรป(พัลส์ต่ำ) เพื่ออ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอก (External Data Memory) เมื่อซีพียูอ่านรหัสคำสั่งจากหน่วยความจำภายนอก จะส่งสัญญาณสโตรปออกมา 2 ครั้งใน 1 แมชชีนไซเกิลแต่สัญญาณสโตรปทั้ง 2 ครั้งจะถูกข้ามไปหากเป็นช่วงที่ซีพียูติดต่อกับ External Data Memory

-EA (External Access)

เป็นสัญญาณอินพุท ใช้สำหรับควบคุมให้ไมโครคอนโทรลเลอร์เลือกติดต่อโปรแกรมที่เก็บอยู่ในหน่วยความจำโปรแกรมภายใน หรือโปรแกรมที่เก็บอยู่ในหน่วยความจำภายนอกตัวไมโครคอนโทรลเลอร์ หากให้ค่าลอจิก 1 ที่ขานี้จะเป็นการเลือกใช้หน่วยความจำโปรแกรมภายในตัวไมโครคอนโทรลเลอร์ หากต้องการให้ซีพียูติดต่อกับหน่วยความจำโปรแกรมภายนอกต้องต่อสัญญาณเข้ากับลอจิก 0 หรือ VSS ถึงแม้ว่าเบอร์ 8031 ไม่มี EPROM ภายในต้องต่อขานี้ลงกราวด์ด้วย ในกรณีของการโปรแกรม ROM ภายในไมโครคอนโทรลเลอร์จะต่อขานี้เข้ากับไฟ 21 V ถ้าเป็น 8751 AH แต่หากเป็น 8751 BH ต้องต่อกับ 12.75 V

- PORT 0

เป็นพอร์ตอินพุทเอาต์พุท 2 ทางแบบ OPEN DRAIN ขนาด 8 บิท P0.1-P0.7 เมื่อใช้เป็นเอาต์พุทสามารถต่อกับไอซี TTL ตระกูล LS ได้ 8 ตัว เมื่อต้องการใช้งานเป็นอินพุทต้องส่งค่าลอจิก 1 ออกไปที่พอร์ตก่อนเพื่อทำให้ลอยซึ่งจะเป็นอิมพีแดนซ์สูงพอร์ต P0 จะทำงานอีกหน้าที่หนึ่งคือ เป็นมัลติเพล็กซ์ของสัญญาณตำแหน่งด้านต่ำ และสัญญาณข้อมูลในการติดต่อกับหน่วยความจำภายนอก การทำงานในลักษณะนี้จะใช้การพุลอัพ (Pull Up) จากภายในที่สามารถจ่ายกระแสให้กับอินพุทของ TTL ได้ 8 ตัว นอกจาก 2 หน้าที่ดังกล่าวแล้ว พอร์ต P0 ยังใช้เป็นตัวรับข้อมูลในช่วงการโปรแกรม EPROM ซึ่งจะต้องใช้พูลอัพจากภายนอกในขณะทำการตรวจสอบโปรแกรมการสร้างสัญญาณตำแหน่ง A0-A7 และสัญญาณข้อมูล D0-D7 ทำโดยใช้อุปกรณ์แลทช์ข้อมูลอุปกรณ์ที่ที่นำมาใช้แลทช์ตำแหน่ง A0-A7 ที่ออกมาจากพอร์ต P0 คือ ไอซี 74LS373 ซึ่งเป็น 8 บิตแลทช์เราสามารถนำมาต่อใช้งานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- PORT 1

เป็นพอร์ตอินพุท/เอาต์พุท แบบ 2 ทาง ขนาด 8 บิท ที่มีพูลอัพอยู่ภายใน ในกรณีเอาต์พุทจะต่อกับอินพุทของ TTL ตระกูล LS ได้ 4 ตัว เมื่อต้องการใช้เป็นอินพุทต้องเริ่มต้นด้วยการส่งค่าลอจิกออกไปที่พอร์ตนี้ก่อน เพื่อทำให้เกิดพูลอัพภายใน เมื่อสัญญาณอินพุทเป็น 0 เข้ามาจะทำให้ พอร์ตจ่ายกระแสแกเนื่องจากมีการพูลอัพภายใน นอกจากนี้พอร์ต P1 ยังทำหน้าที่รับตำแหน่งด้านต่ำในช่วงของการโปรแกรม EPROM และในช่วงการตรวจสอบโปรแกรมใน ROM หรือ EPROM อีกด้วย สำหรับเบอร์ 8032 AH ขา P1.0 และ P1.1 จะทำหน้าที่เป็น T2 และ T2EX อีกหนึ่งหน้าที่

- PORT 2

เป็นพอร์ตอินพุท/เอาท์พุทแบบ 2 ทาง ขนาด 8 บิต ที่มีพูลอัพอยู่ภายใน ในกรณีเอาท์พุทจะต่อกับอินพุทของ TTL ตระกูล LS ได้ 4 ตัว เมื่อต้องการทำเป็นอินพุท ต้องเริ่มต้นด้วยการส่งค่าลอจิกออกไปที่พอร์ตก่อนเพื่อทำให้เกิดการพูลอัพภายใน เมื่อสัญญาณอินพุทเข้ามาเป็น 0 จะทำให้พอร์ต P2 จ่ายกระแสออกเนื่องจากการพูลอัพภายใน ในระหว่างการติดต่อกับโปรแกรมภายนอกหรือการติดต่อกับหน่วยความจำข้อมูลภายนอกที่มีการอ้างตำแหน่งแบบ 16 บิต พอร์ต P2 จะส่งตำแหน่งไบต์สูงออกไป ซึ่งการทำงานในลักษณะนี้จะมีการพูลอัพภายในอยู่ ในช่วงของการติดต่อกับหน่วยความจำข้อมูลภายนอกที่มีการพูลอัพภายในอยู่ ในช่วงของการติดต่อกับหน่วยความจำข้อมูลภายนอกที่ใช้การอ้างตำแหน่งแบบ 8 บิต (คำสั่ง MOVX @Ri) สัญญาณที่ขาของพอร์ต P2 จะมีค่าเท่ากับรีจิสเตอร์ P2 ที่อยู่ใน SFR นอกจากนี้พอร์ต P2 ยังทำหน้าที่รับตำแหน่งไบต์สูง ในช่วงของการโปรแกรม EPROM และการตรวจสอบโปรแกรมใน ROM และ EPROM อีกด้วย

- PORT 3

เป็นพอร์ตอินพุท/เอาท์พุท 2 ทาง ขนาด 8 บิต ที่มีพูลอัพอยู่ภายในในกรณีเอาท์พุทสามารถต่อกับอินพุทของ TTL ตระกูล LS ได้ 4 ตัว เมื่อต้องการทำเป็นอินพุทต้องเริ่มต้นด้วยการส่งค่าลอจิก 1 ออกไปที่พอร์ตก่อนเพื่อทำให้เกิดการพูลอัพภายใน เมื่อสัญญาณอินพุทเข้ามาเป็น 0 จะทำให้พอร์ต P3 จ่ายกระแสออกเนื่องจากการพูลอัพภายใน นอกจากนี้ พอร์ต P3 ยังทำหน้าที่เป็นสัญญาณอื่นๆ อีกดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พอร์ต	สัญญาณ	หน้าที่
P3.0	RXD	อินพุทของพอร์ตอนุกรม
P3.1	TXD	เอาท์พุทของพอร์ตอนุกรม
P3.2	INTO	สัญญาณอินเตอร์รัพต์จากภายนอกตัวที่ 0
P3.3	INT	สัญญาณอินเตอร์รัพต์จากภายนอกตัวที่ 1
P3.4	T0	อินพุทจากภายนอกของตัวตั้งเวลา 0
P3.5	T1	อินพุทจากภายนอกของตัวตั้งเวลา 1
P3.6	WR	สัญญาณการเขียนข้อมูลออกไปภายนอก
P3.7	RD	สัญญาณการอ่านข้อมูลจากภายนอกเข้ามา

ตารางที่ 2.3 สัญญาณต่าง ๆ ของพอร์ต P3

เมื่อต้องการใช้งานพอร์ต P3 ให้ทำหน้าที่เป็นสัญญาณต่าง ๆ จะต้องเริ่มด้วยการส่งค่าลอจิก 1 ออกไปแลทช์ที่พอร์ต P3 ก่อนเพื่อให้เกิดการพูลอัพภายใน หากเรากำหนดให้มีค่าลอจิก 0 จะทำให้สัญญาณที่ขาต่าง ๆ มีค่าเป็น 0 ตลอดเวลา

2.2.3 รีเลย์ (Relay)

รีเลย์ เป็นสวิตช์ที่ทำงานเปิดปิด วงจรโดยใช้แม่เหล็ก แทนที่จะใช้เปิดปิดวงจรแบบสวิตช์ปกติซึ่งใช้สปริงค์หน้าคอนแท็คจะกินกระแสเพียงเล็กน้อยเป็นมิลลิแอมป์ โดยจะทำงานเพื่อดึงหน้าคอนแท็คให้เปิด หรือ ปิด วงจร โดยทำหน้าที่เปิดสวิทช์ยอมให้กระแสผ่านถึง 10 แอมป์

ข้อดีของรีเลย์ตรงที่การปิดหน้าของคอนแท็คจะเป็นไปอย่างรวดเร็ว ก็เนื่องจากแรงดึงของคอยล์ โดยอำนาจแม่เหล็กจะทำให้ลดประกายไฟลงที่หน้าคอนแท็คได้มาก รีเลย์ มีขนาดตั้งแต่เล็กๆ ใช้ในงานอิเล็กทรอนิกส์ ไปจนถึงงานขนาดกระแสเป็น 200 แอมป์ ในงานไฟฟ้ากำลัง เป็นทั้งตัวป้องกันการลัดวงจรโดยยกสวิตช์ ใช้บังคับการทำงานของมอเตอร์ ใช้เป็นตัวเปิดปิดไฟทั้งวงจร นอกจากนี้รีเลย์ยังใช้ในเครื่องใช้สำนักงาน เช่น เครื่องถ่ายเอกสาร คอมพิวเตอร์ เครื่องควบคุมต่างๆมากมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่3
การดำเนินการวิจัย

ในบทที่ 3 นี้เราจะกล่าวถึงขั้นตอนในการดำเนินการวิจัย ซึ่งเราจะแบ่งขั้นตอนในการดำเนินการวิจัยเป็น 2 ส่วนใหญ่ๆ คือ การออกแบบทางฮาร์ดแวร์ และการออกแบบทางซอร์ฟแวร์โดยการออกแบบทางฮาร์ดแวร์จะเป็นเรื่องของโครงสร้างของหุ่นยนต์ และวงจรที่ใช้ ส่วนการออกแบบทางซอร์ฟแวร์ เราจะกล่าวถึงโปรแกรมที่ใช้เป็นคำสั่งในไมโครคอนโทรลเลอร์ MCS-51

3.1 การออกแบบทางฮาร์ดแวร์

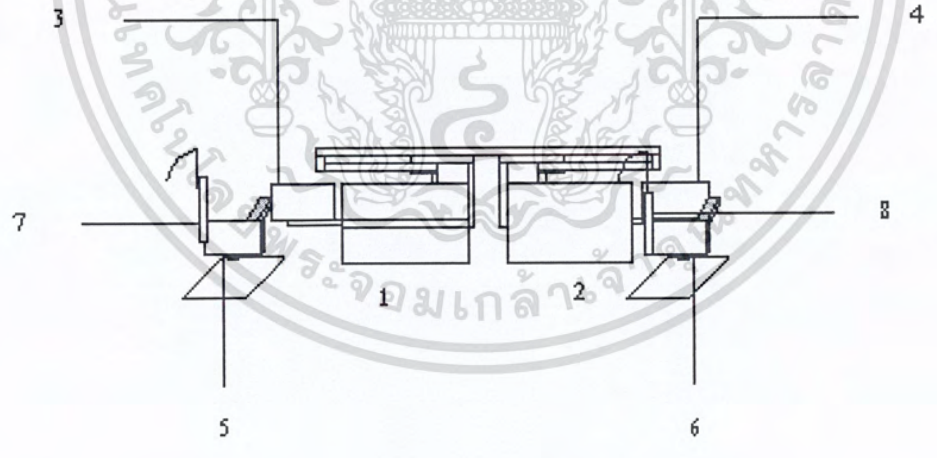
3.1.1 การออกแบบโครงสร้างของหุ่นยนต์

หุ่นยนต์ไต่กำแพงนี้จะเป็นหุ่นยนต์ที่มีลักษณะมือจับ 2ข้าง โดยสามารถสลับกันจับตาข่ายได้ โดยการทำงานจะถูกควบคุมด้วยอุปกรณ์ไมโครคอนโทรลเลอร์MCS-51และอุปกรณ์ทางอิเล็กทรอนิกส์ โดยใช้เซอร์โวมอเตอร์เป็นกำลังหลักในการขับเคลื่อน

หุ่นยนต์มีขนาด กว้าง 30 ซม. ยาว 50 ซม. สูง 20 ซม.

หุ่นยนต์มีน้ำหนัก 2.2 กก.

โครงหุ่นยนต์ทำจากพลาสติกภายในมีอุปกรณ์ต่างๆเชื่อมโยงกันดังนี้



รูปที่3.1 แสดงโครงสร้างของหุ่นยนต์เมื่อมองจากด้านหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

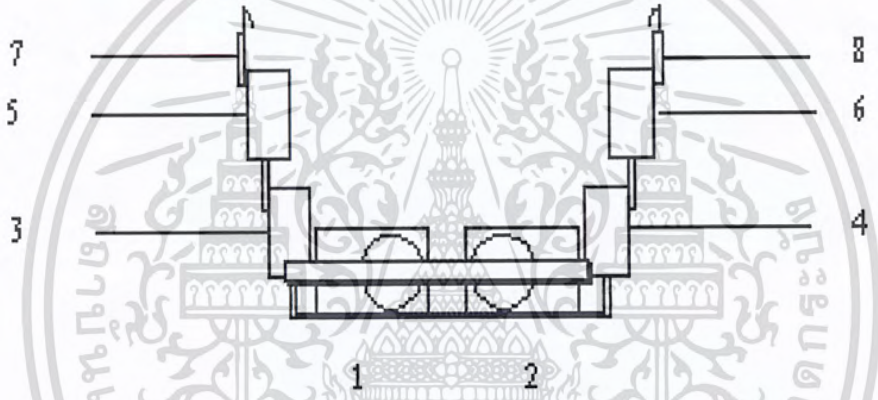
จากรูปที่ 3.1

หมายเลข 1และ 2 เป็นเซอร์โวมอเตอร์ขนาด 21 กิโลกรัม

หมายเลข 3,4,5 และ 6 เป็นเซอร์โวมอเตอร์ขนาด 7กิโลกรัม

หมายเลข 7และ 8 เป็นนิ้วของหุ่นยนต์ใช้สำหรับเกี่ยวเส้นลวด

จากรูปเราจะเห็นได้ว่าเราจะใช้เซอร์โวมอเตอร์หมายเลข 1และ 2 เป็นตัวยกหุ่นยนต์ในแนวตั้งฉากกับพื้นโลกให้เคลื่อนที่ไปในทิศทางที่กำหนดได้ สำหรับเซอร์โวมอเตอร์ 3และ4 จะใช้ยกแขนซึ่งมีเซอร์โวมอเตอร์ 5และ6 วางอยู่ ให้สามารถยกขึ้นและวางได้เพื่อเอาไว้ใช้สำหรับขั้นตอนการยกแขนเพื่อปล่อยหรือยกแขนเพื่อวางจับ ส่วนเซอร์โวมอเตอร์ 7กับ8 เอาไว้สำหรับ ให้นิ้วลวดหมุนงอเข้าไปแหย่รูตระแกรง เพื่อยึดเกี่ยวเอาไว้ขณะที่ทำการจับกับตาข่ายลวด สามารถมองภาพจากด้านบนได้จากรูปที่ 3.2



รูปที่ 3.2 แสดงโครงสร้างหุ่นยนต์เมื่อมองจากด้านบน

หลังจากที่เราได้ออกแบบส่วนโครงสร้างของหุ่นยนต์แล้วต่อไปเราก็จะกำหนดลักษณะการเดินของหุ่นยนต์โดยเราจะกำหนดการเดินของหุ่นยนต์ในแต่ละขั้นตอน ว่าส่วนใดของหุ่นยนต์จะต้องทำงานในขณะที่เราสั่งให่หุ่นยนต์ทำการเดิน สำหรับขั้นตอนการเดินของหุ่นยนต์เราจะกำหนดได้ดังนี้

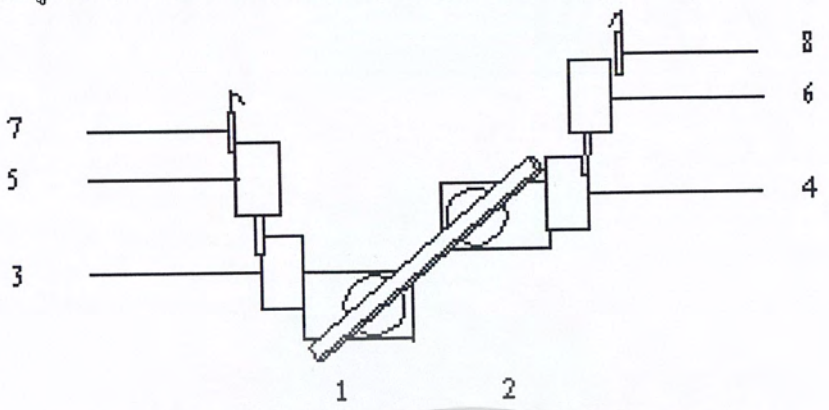
3.1.2 การออกแบบท่าเดินของหุ่นยนต์ไต่กำแพง

3.1.2.1 การเดินหน้า

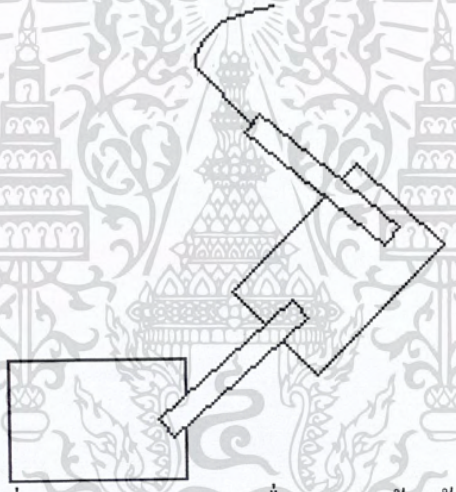
สำหรับขั้นตอนการเดินหน้านั้น ขั้นแรกเราจะเซตหุ่นยนต์ให้อยู่ในท่าที่เตรียมพร้อมก่อนดังรูปที่ 3.3 เมื่อหุ่นยนต์ถูกเซตให้เตรียมพร้อมที่จะเดินแล้ว ในการไต่ขึ้นข้างบน (จากส่วนล่างไต่ขึ้นไปส่วนบน) เราจะกำหนดให้ส่วนของนิ้วทางด้านแขนขวาปล่อยตาข่ายลวดเสียก่อนโดยการสั่งให้เซอร์โวมอเตอร์ ตัวที่ 7 หมุนในทิศทวนเข็มนาฬิกามา 90 องศา จากนั้นแขนทางด้านขวามือก็จะยกขึ้นทำมุมประมาณ 45 องศากับตาข่ายลวด โดยมอเตอร์ตัวที่ 5 จะหมุนไปในทิศทางทวนเข็มนาฬิกา(เมื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มองจากทางด้านหน้าของมอเตอร์)ประมาณ 45 องศาเป็นผลให้ ขณะนี้ ทั้งนิ้วและแขนของหุ่นยนต์ไม่มีส่วนใดยึดติดอยู่กับตาข่ายเลย สำหรับมอเตอร์ตัวที่ 4,6,8และ2ยังคงเดิมดังรูป



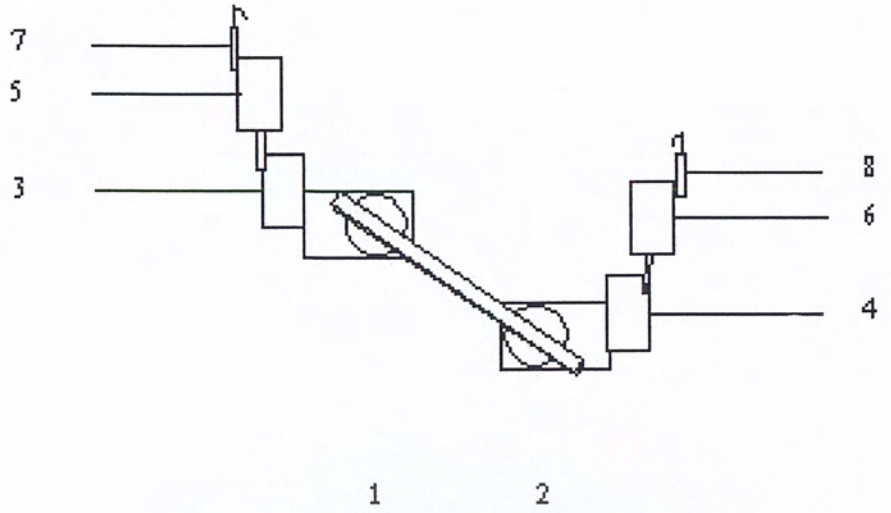
รูปที่ 3.3 หุ่นยนต์ขณะอยู่ในสภาวะเริ่มต้นก่อนทำการเดิน



รูปที่ 3.4แสดงการยกแขนเมื่อมองจากด้านข้าง

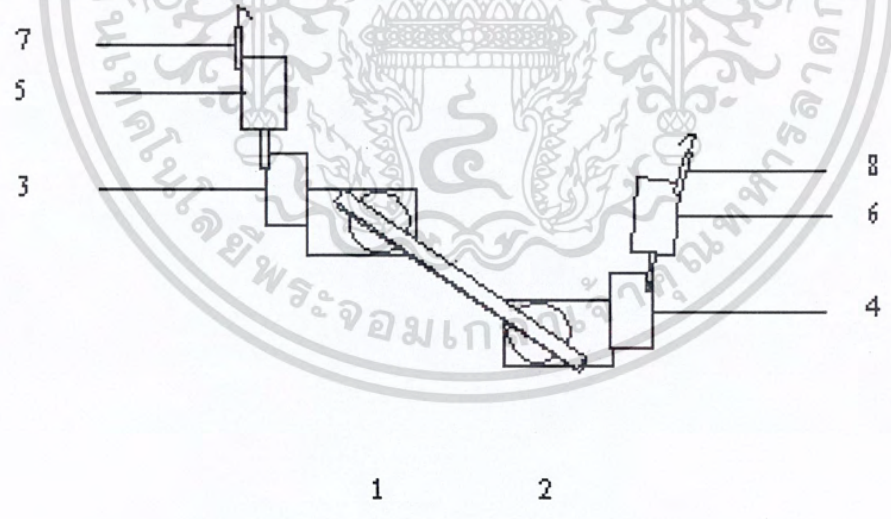
หลังจากที่หุ่นยนต์ได้ทำการยกแขนซ้ายแล้วหุ่นยนต์ก็พร้อมที่จะทำการไต่ขึ้นด้านบน โดยการให้เซอร์โวมอเตอร์ตัวที่ 2 ทำการหมุนในทิศทางตามเข็มนาฬิกา ไปประมาณ 90 องศา และขณะเดียวกันเซอร์โวมอเตอร์ตัวที่ 1ก็จะหมุนในทิศทางตามเข็มนาฬิกาด้วย ประมาณ 90 องศา สามารถแสดงได้ดังรูปที่ 3.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 ขณะหุ่นยนต์เริ่มทำการไต่ขึ้นในจังหวะแรก

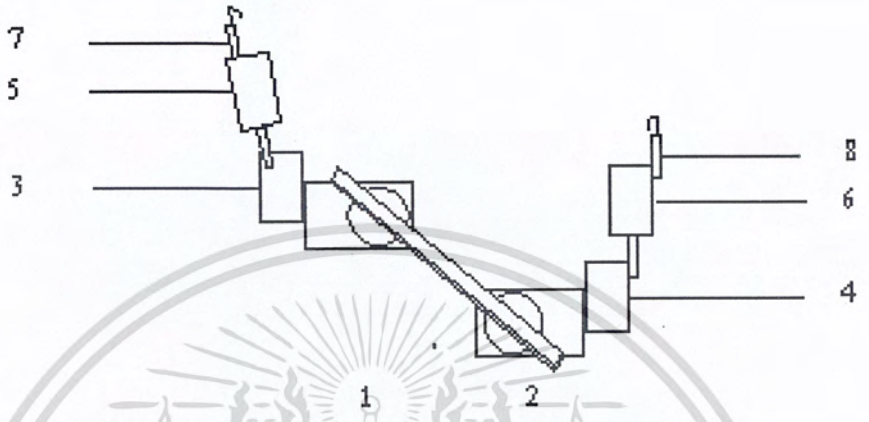
จากนั้น เซอร์โวมอเตอร์ตัวที่3จะถูกสั่งให้หมุนไปในทิศตามเข็มนาฬิกาเป็นมุม 45 องศา ส่งผลให้แขนในด้านซ้ายวางตัวลงและเซอร์โวมอเตอร์ตัวที่ 5 จะถูกสั่งให้หมุนไปในทิศตามเข็มนาฬิกาเป็นมุม 90 องศา นิ้วหมายเลข 7 ก็จะยึดเกี่ยวตาข่ายลวดเอาไว้ จากนั้นเซอร์โวมอเตอร์ตัวที่ 6จะหมุนทวนเข็มนาฬิกาเป็นมุม 90 องศา และเซอร์โวมอเตอร์ตัวที่4 ก็หมุนทวนเข็มนาฬิกา เป็นมุม 45 องศา ส่งผลให้ขาขวายกจากตาข่ายลวดและนิ้วหมายเลข 8 ก็ยกจากรูตาข่ายลวดด้วยดังรูปที่ 3.6



รูปที่ 3.6 ขณะหุ่นยนต์ใช้ขาขวาจับลวดและขาซ้ายปล่อยลวด

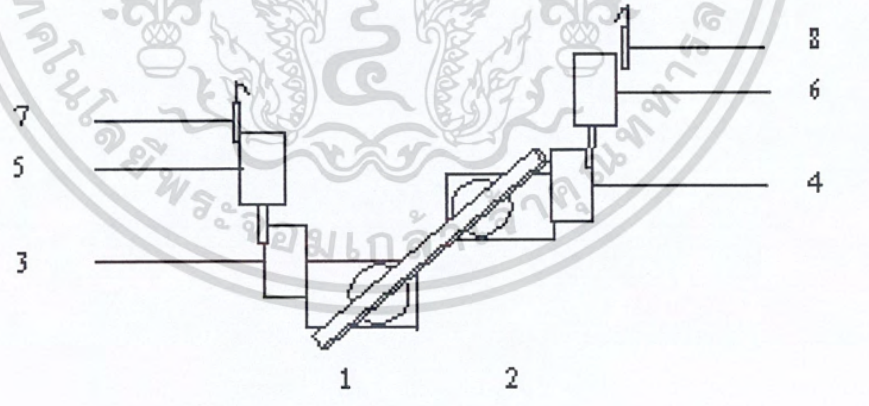
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในจังหวะต่อมา เซอร์โวมอเตอร์ตัวที่ 1และตัวที่ 2 จะทำการหมุนไปในทิศตามเข็มนาฬิกาเป็นมุมประมาณ 90 องศา พร้อมกันนั้น เซอร์โวมอเตอร์ก็จะทำการหมุนไปในทิศตามเข็มนาฬิกาเป็นมุม 90 องศาเช่นกันทำให้ได้ระยะทางเพิ่มขึ้นอีก ดังรูปที่ 3.7



รูปที่ 3.7 แสดงขณะหุ่นยนต์ก้าวขาเพื่อเพิ่มระยะทาง

หุ่นยนต์จะกระทำการก้าวขาไปเรื่อยๆ จนกระทั่งเมื่อใดก็ตาม ที่ไม่ได้รับสัญญาณจากเครื่องส่ง โปรแกรมจะสั่งให้หุ่นยนต์กระทำการเดินต่อไปจนกระทั่งกลับเข้าสู่สภาวะเริ่มต้น ดังรูปที่ 3.8

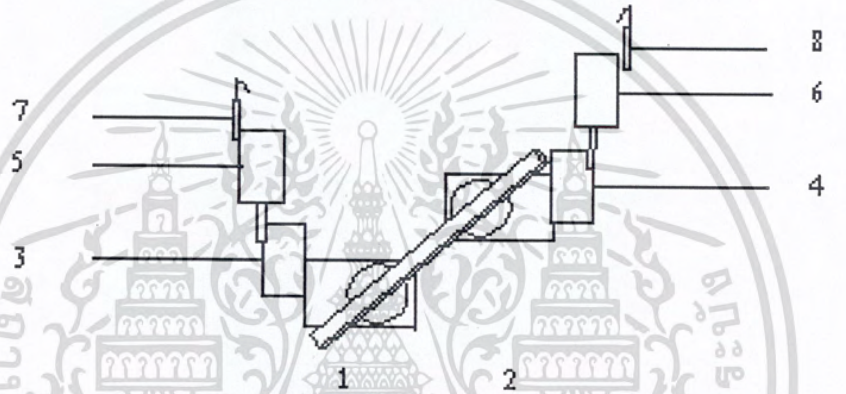


รูปที่ 3.8 แสดงสภาวะสิ้นสุดหลังจากเดินไปข้างหน้า

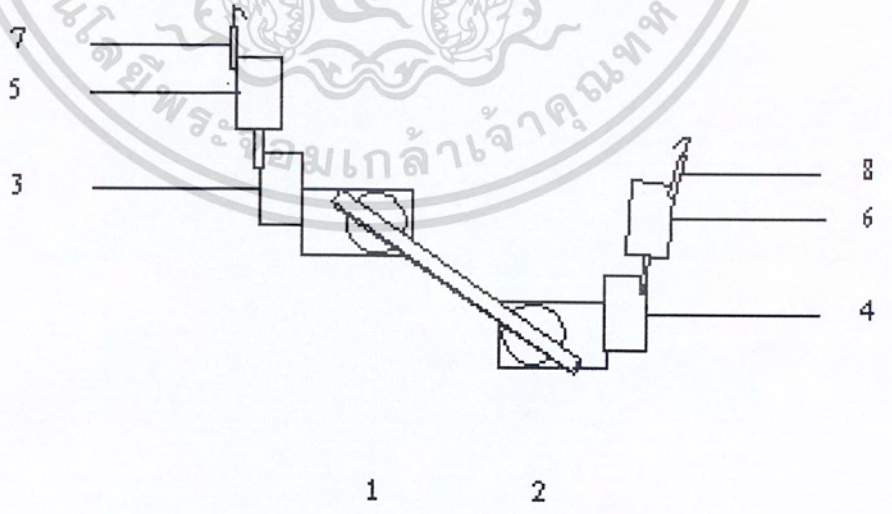
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2.2 การถอยหลัง

เมื่อเราต้องการให้หุ่นยนต์ถอยหลัง ขั้นแรกหุ่นยนต์ต้องอยู่ในสภาวะเริ่มต้นก่อน ดังรูปที่ 3.9 จากนั้นหุ่นยนต์จะปล่อยนิ้วที่เกาะกับตาข่ายลวดทางด้านขวาออก และยกขาขวาขึ้นจากนั้นเซอร์โวมอเตอร์ตัวที่ 1และ 2 จะหมุนตามเข็มนาฬิกาเป็นมุม 45 องศาดังรูปที่ 3.10



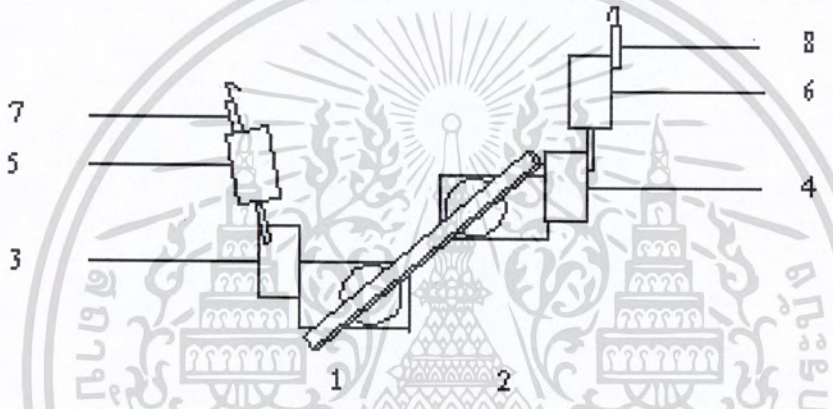
รูปที่ 3.9 หุ่นยนต์อยู่ในสภาวะเริ่มต้นเตรียมที่จะถอยหลัง



รูปที่ 3.10 การเดินถอยหลังในจังหวะแรก

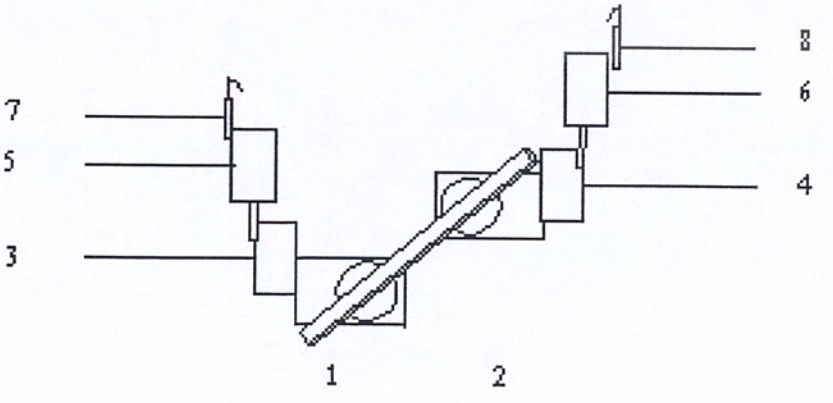
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นแขนทางด้านขวา ก็จะทำการวางลงและแขนทางนิ้วมือหมายเลข 8 ก็จะเกี่ยวเข้ากับตาข่ายลวด ในเวลาต่อมา นิ้วมือหมายเลข 7 ก็จะปล่อยตาข่ายลวดและแขนทางข้างซ้ายก็จะยกขึ้น หลังจากนั้นเซอร์โวมอเตอร์ตัวที่ 1และ 2ก็จะหมุนไปในทิศทางทวนเข็มนาฬิกา ตัวละ 90 องศาเพื่อที่จะเดินให้ได้ระยะทางจะได้ลักษณะเหมือนดังรูปที่ 3.11



รูปที่ 3.11 แสดงการเดินในจังหวะต่อไปเพื่อเพิ่มระยะทาง

และจะทำการเดินต่อไปเรื่อยๆ จนกระทั่ง เครื่องส่งหยุดส่งสัญญาณ โปรแกรมก็จะทำให้หุ่นยนต์ทำงานต่อไปจนกระทั่งกลับสู่สภาวะเริ่มต้นอีกครั้งเพื่อรอรับคำสั่งต่อไป ดังรูปที่ 3.12

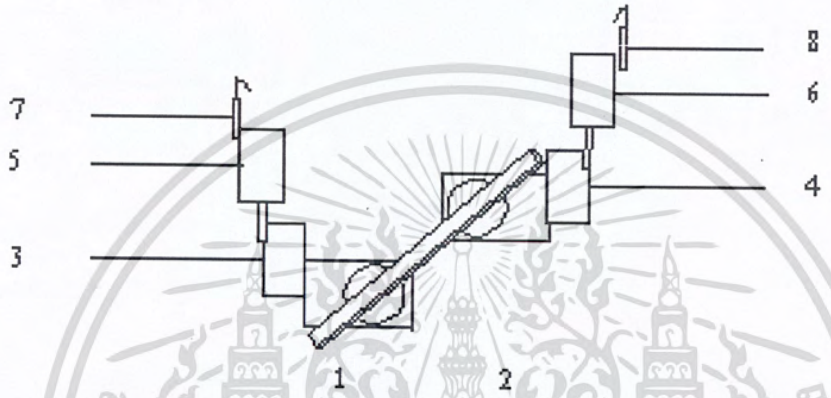


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.12 แสดงสภาวะสิ้นสุดจากทำการเดินถอยหลัง

3.1.2.3 การเลี้ยวขวา

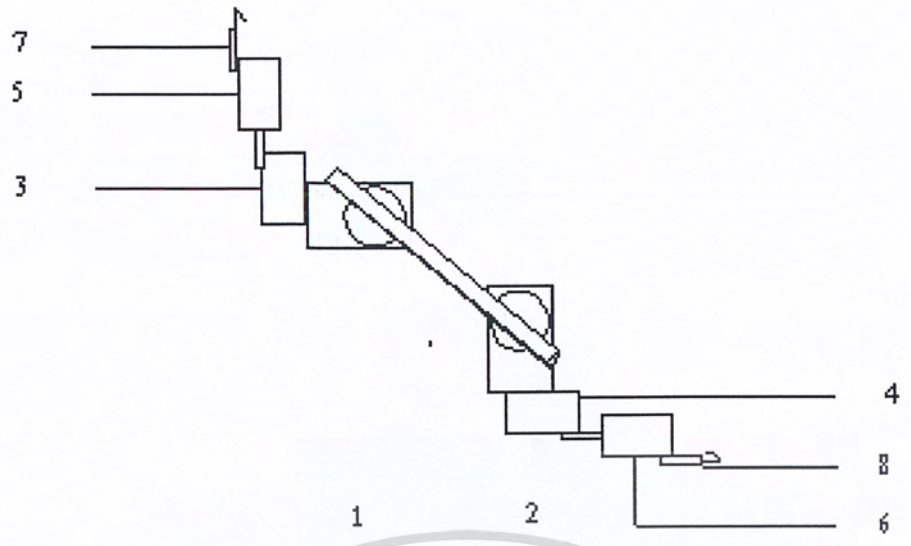
สำหรับขั้นตอนการเลี้ยวขวานั้น ขั้นแรกหุ่นยนต์ก็จะอยู่ในสภาวะเริ่มต้นดังรูปที่ 3.13 หลังจากนั้นแล้ว โปรแกรมจะสั่งให้หุ่นยนต์ทำการยกนิ้วหมายเลข 8 โดยหมุนเซอร์โวมอเตอร์ตัวที่ 6 ไปในทิศทางทวนเข็มนาฬิกาจากนั้นแล้ว ก็ยกแขนด้านขวาขึ้นโดยการหมุนเซอร์โวมอเตอร์ หมายเลข 4 ไป 45 องศา



รูปที่ 3.13 แสดงสภาวะเริ่มต้นก่อนทำการเลี้ยวขวา

เมื่อแขนขวาทำการยกขึ้นแล้วเซอร์โวมอเตอร์ตัวที่ 1 ก็จะทำการหมุนไปในทิศทางตามเข็มนาฬิกา เป็นมุม 90 องศาเมื่อทำเสร็จแล้วโปรแกรมก็จะสั่งให้เซอร์โวมอเตอร์ตัวที่ 2 ก็จะหมุนไปในทิศทางทวนเข็มนาฬิกา เป็นมุม 45 องศาจากนั้นทำการวางแขนด้านขวาลงและนิ้วของหุ่นยนต์หมายเลข 8 ก็จะทำการเกี่ยวกับตาข่ายลวดเอาไว้ ดังรูปที่ 3.14

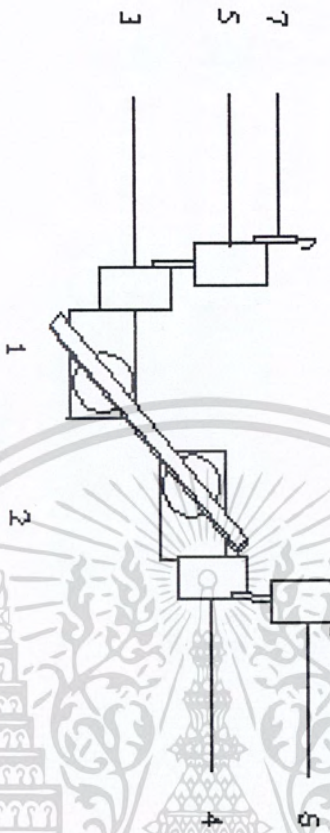
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.14 ขณะทำการเลี้ยวขวาเมื่อสิ้นสุดจังหวะแรก

ในจังหวะต่อมา เราจะทำการยกนิ้วหมายเลข 7 และแขนทางด้านซ้ายจากนั้นกำหนดให้เซอร์โวมอเตอร์หมายเลข 2 ทำการหมุนตามเข็มนาฬิกา 45 องศาเมื่อหมุนเสร็จแล้วก็สั่งให้ เซอร์โวมอเตอร์หมายเลข 1 หมุนในทิศทางทวนเข็มนาฬิกา 45 องศา จะทำให้หุ่นยนต์สามารถ อยู่ในสภาวะเริ่มต้นได้เพียงแต่ตัวหุ่นยนต์ ได้หันหน้าไปทางทิศขวาแล้ว ดังรูปที่ 3.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

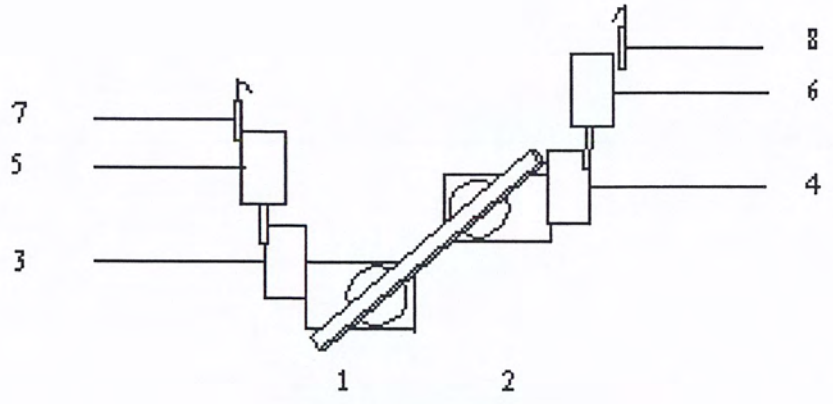


รูปที่ 3.15 หุ่นยนต์ในสภาวะที่ทำการเลี้ยวขวาเสร็จสิ้นแล้ว

3.1.2.4 การเลี้ยวซ้าย

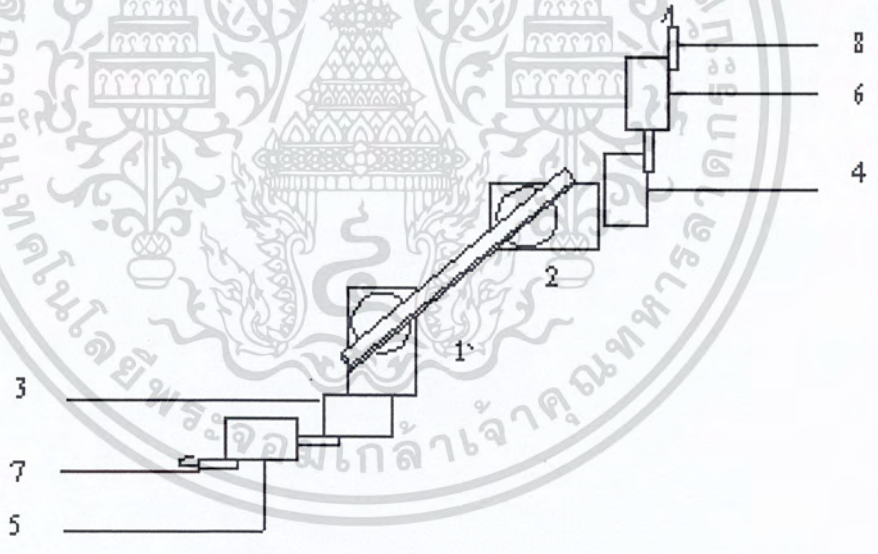
สำหรับการเลี้ยวซ้ายนั้นก็มีวิธีการเช่นเดียวกับการเลี้ยวไปทางขวาโดยต้องกำหนดให้หุ่นยนต์อยู่ในสภาวะเริ่มต้นเสียก่อน ดังรูปที่ 3.16 หลังจากนั้นแล้ว โปรแกรมจะสั่งให้หุ่นยนต์ทำการยกนิ้วหมายเลข 7 โดยหมุนเซอร์โวมอเตอร์ตัวที่ 5 ไปในทิศทางทวนเข็มนาฬิกาจากนั้นแล้ว ก็ยกแขนด้านซ้าย ขึ้นโดยการหมุนเซอร์โวมอเตอร์ หมายเลข 3ไป 45 องศา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.16 แสดงสภาวะเริ่มต้นก่อนทาการเลี้ยวซ้าย

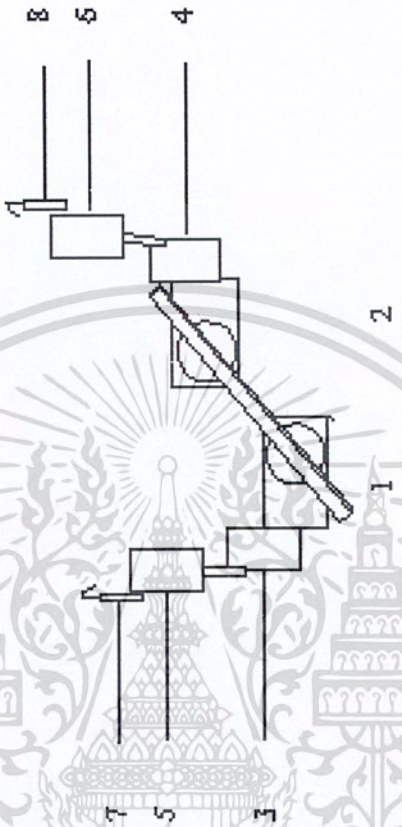
เมื่อแขนซ้ายทำการยกขึ้นแล้วเซอร์โวมอเตอร์ตัวที่ 2 ก็จะทำการหมุนไปในทิศทางทวนเข็มนาฬิกา เป็นมุม 45 องศาเมื่อทำเสร็จแล้วโปรแกรมก็จะสั่งให้เซอร์โวมอเตอร์ตัวที่ 2 ก็จะหมุนไปในทิศทางตามเข็มนาฬิกา เป็นมุม 45 องศาจากนั้นทำการวางแขนด้านขวาลงและนิ้วของหุ่นยนต์หมายเลข7 ก็จะทำการเกี่ยวกับตาข่ายลวดเอาไว้ ดังรูปที่ 3.17



รูปที่ 3.17 ขณะทำการเลี้ยวซ้ายเมื่อสิ้นสุดจังหวะแรก

ในจังหวะต่อมา เราจะทำการยกนิ้วหมายเลข 8 และแขนทางด้านขวาจากนั้นกำหนดให้เซอร์โวมอเตอร์หมายเลข 1 ทำการหมุนทวนเข็มนาฬิกา 90 องศาเมื่อหมุนเสร็จแล้วก็สั่งให้ เซอร์โวมอเตอร์หมายเลข 2 หมุนในทิศทางตามเข็มนาฬิกา 45 องศา จะทำให้หุ่นยนต์สามารถ อยู่ในสภาวะเริ่มต้นได้เพียงแต่ตัวหุ่นยนต์ ได้หันหน้าไปทางทิศซ้ายแล้ว ดังรูปที่ 3.18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.18 หุ่นยนต์ในสภาวะที่ทำการเลี้ยวซ้ายเสร็จสิ้นแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

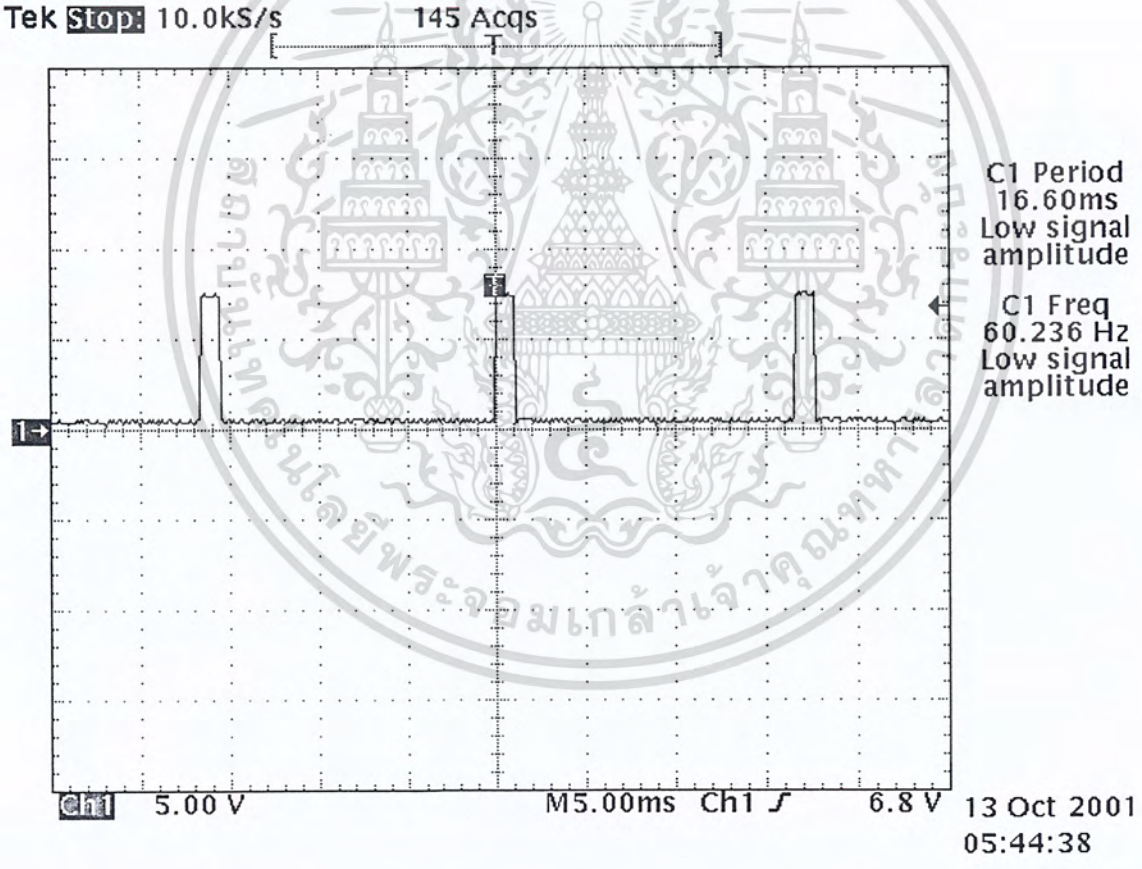
บทที่4

การทดลองและผลการทดลอง

ในบทนี้เราจะกล่าวถึง การทดลองในการเขียนรูปคลื่นพัลซ์ที่ต้องการ ให้มีขนาดตามที่เรากำหนด และทำการวัดค่าจาก ออสซิลโลสโคป (OSCILLO SCOPE) ว่าได้ค่าตามที่เราต้องการหรือใกล้เคียงหรือไม่ และเรายังทำการต่อสัญญาณที่เราสร้างขึ้นมา นำมาต่อเข้ากับเซอร์โวมอเตอร์แล้วสังเกตุทิศทางการหมุนของเซอร์โวมอเตอร์อีกด้วย และยังทดลองการทำงานระหว่างโปรแกรมกับหุ่นยนต์อีกด้วยว่าสามารถทำงานได้ดีแค่ไหน

4.1 การทดลองจับรูปคลื่นด้วยออสซิลโลสโคป

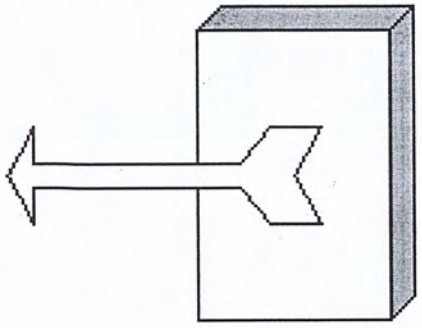
4.1.1ทำการเขียนโปรแกรมสร้างพัลซ์ 1ms (ส่วนของโปรแกรมสามารถดูได้จากภาคผนวก) ลงในไมโครคอนโทรลเลอร์ จากนั้นใช้ออสซิลโลสโคป จับค่าของสัญญาณพัลซ์ที่ เอาท์พุทของไมโครคอนโทรลเลอร์ว่า มีค่าเท่าใดดังรูปที่ 4.1



รูปที่ 4.1 แสดงรูปคลื่นพัลซ์ขนาด 1 ms

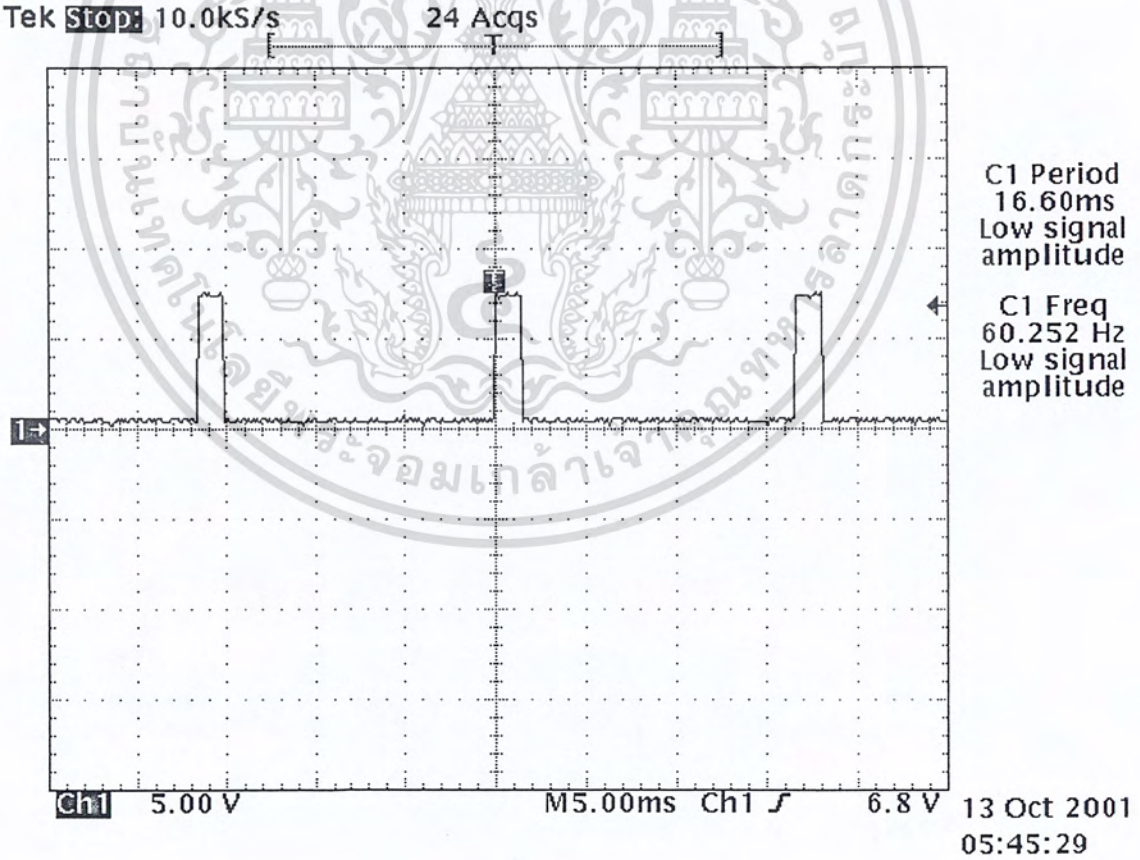
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อนำสัญญาณพัลซ์ ที่มีค่า 1 ms มาควบคุมมอเตอร์ ทิศทางในการหมุนของมอเตอร์จะเป็นดังรูปที่ 4.4



รูปที่ 4.2 แสดงการทิศการหมุนของเซอร์โวมอเตอร์ขณะสร้าง พัลซ์ 1 ms

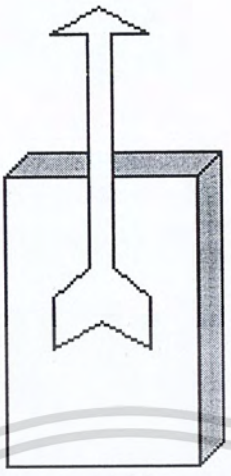
4.1.2 ทำการเขียนโปรแกรมสร้างพัลซ์ 1.5 ms (ส่วนของโปรแกรมสามารถดูได้จากภาคผนวก) ลงในไมโครคอนโทรลเลอร์ จากนั้นใช้ออสซิลโลสโคป จับค่าของสัญญาณพัลซ์ที่ เอาท์พุทของไมโครคอนโทรลเลอร์ว่า มีค่าเท่าใด ซึ่งได้ผลดังรูปที่ 4.3



รูปที่ 4.3 แสดงรูปคลื่นพัลซ์ขนาด 1.5 ms

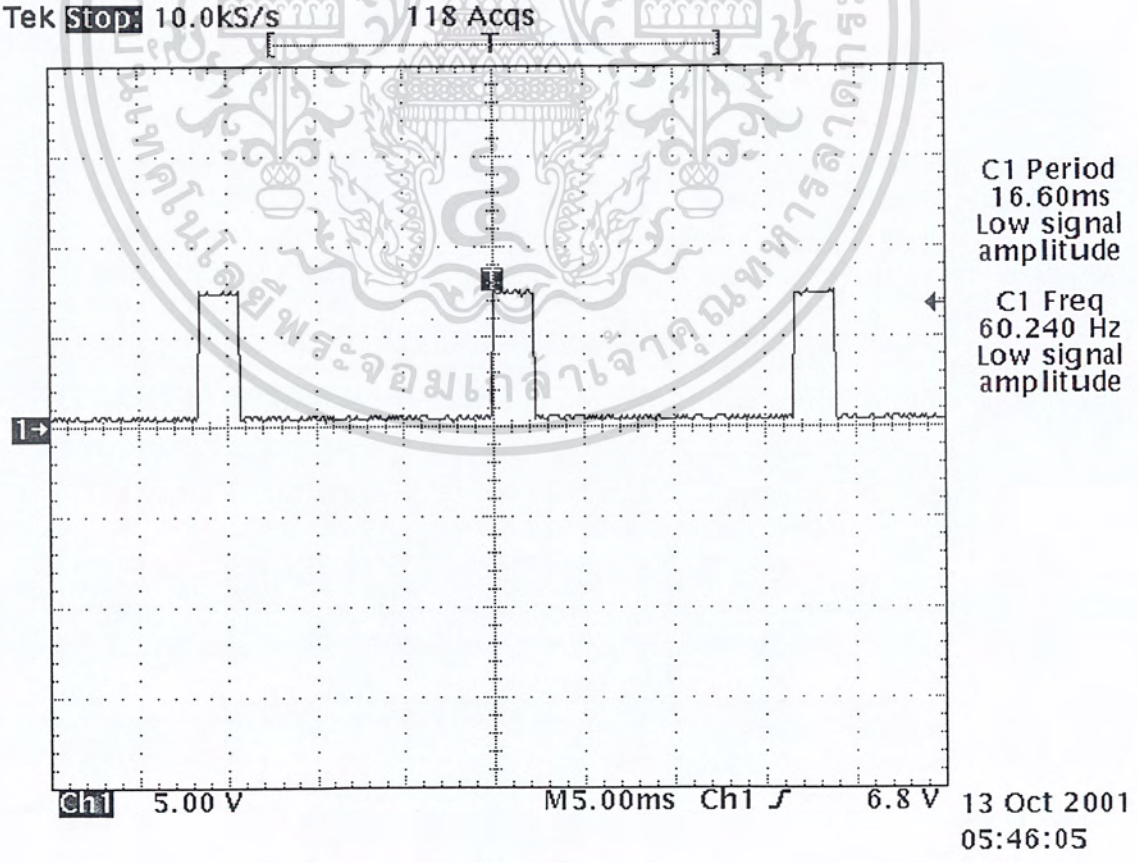
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อนำสัญญาณพัลซ์ ที่มีค่า 1.5 ms มาควบคุมมอเตอร์ ทิศทางในการหมุนของมอเตอร์จะเป็นดังรูปที่ 4.4



รูปที่ 4.4 แสดงการทิศการหมุนของเซอร์โวมอเตอร์ขณะสร้าง พัลซ์ 1.5 ms

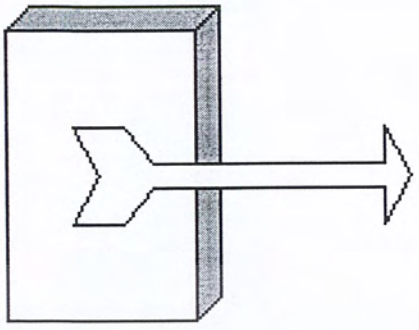
4.1.3 ทำการเขียนโปรแกรมสร้างพัลซ์ 2 ms (ส่วนของโปรแกรมสามารถดูได้จากภาคผนวก) ลงในไมโครคอนโทรลเลอร์ จากนั้นใช้ออสซิลโลสโคป จับค่าของสัญญาณพัลซ์ที่ เอาท์พุทของไมโครคอนโทรลเลอร์ว่า มีค่าเท่าใด ซึ่งได้ผลดังรูปที่ 4.5



รูปที่ 4.5แสดงรูปคลื่นพัลซ์ขนาด 2 ms

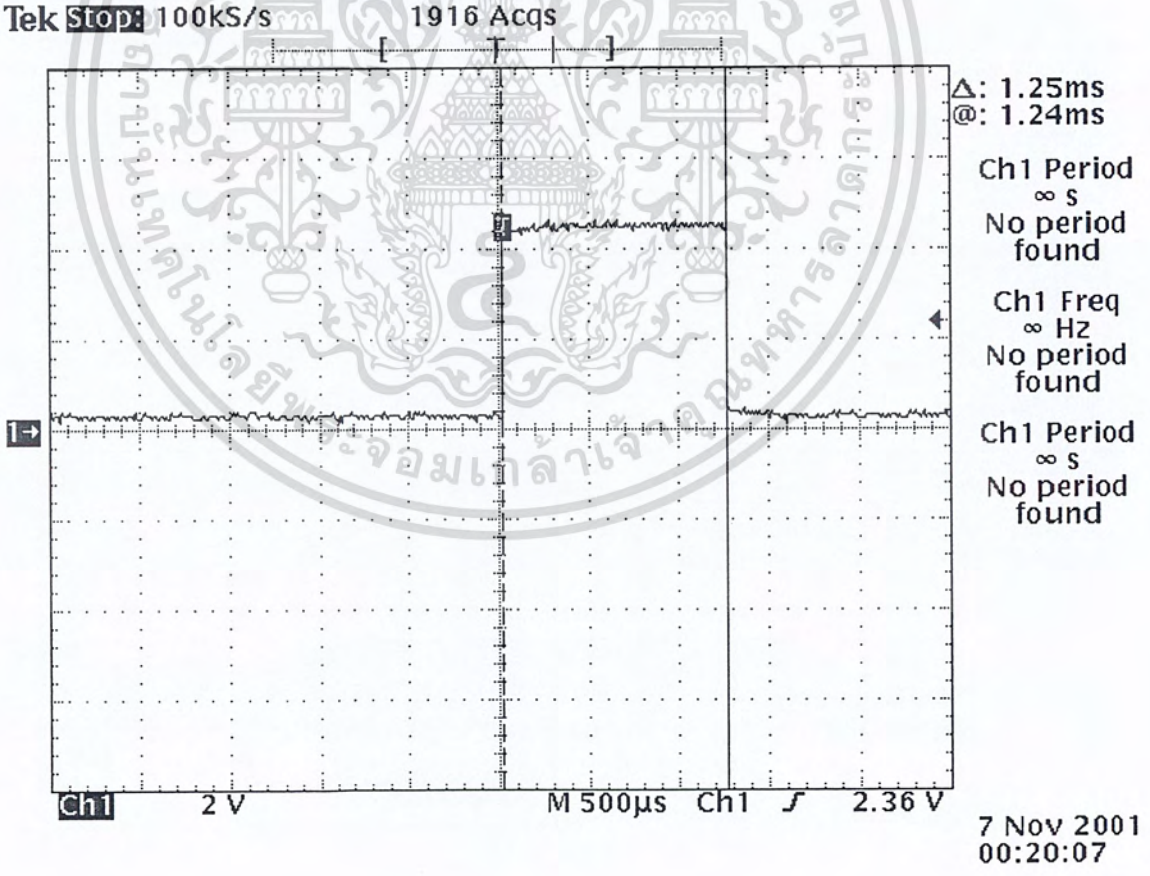
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อนำสัญญาณพัลซ์ ที่มีค่า 2 ms มาควบคุมมอเตอร์ ทิศทางในการหมุนของมอเตอร์จะเป็นดังรูปที่ 4.6



รูปที่ 4.6 แสดงการทิศการหมุนของเซอร์โวมอเตอร์ขณะ พัลซ์ 2 ms

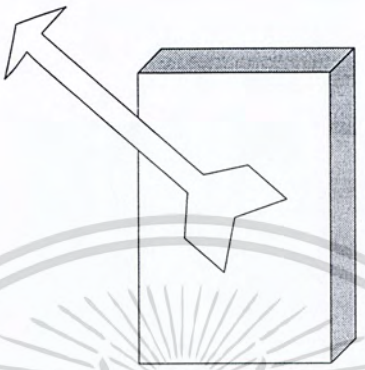
4.1.4 ทำการเขียนโปรแกรมสร้างพัลซ์ 1.25 ms (ส่วนของโปรแกรมสามารถดูได้จากภาคผนวก) ลงในไมโครคอนโทรลเลอร์ จากนั้นใช้ออสซิลโลสโคป จับค่าของสัญญาณพัลซ์ที่ เอาท์พุทของไมโครคอนโทรลเลอร์ว่ามีค่าเท่าใดซึ่งได้ผลดังรูปที่ 4.7



รูปที่ 4.7แสดงรูปคลื่นพัลซ์ขนาด 1.25 ms

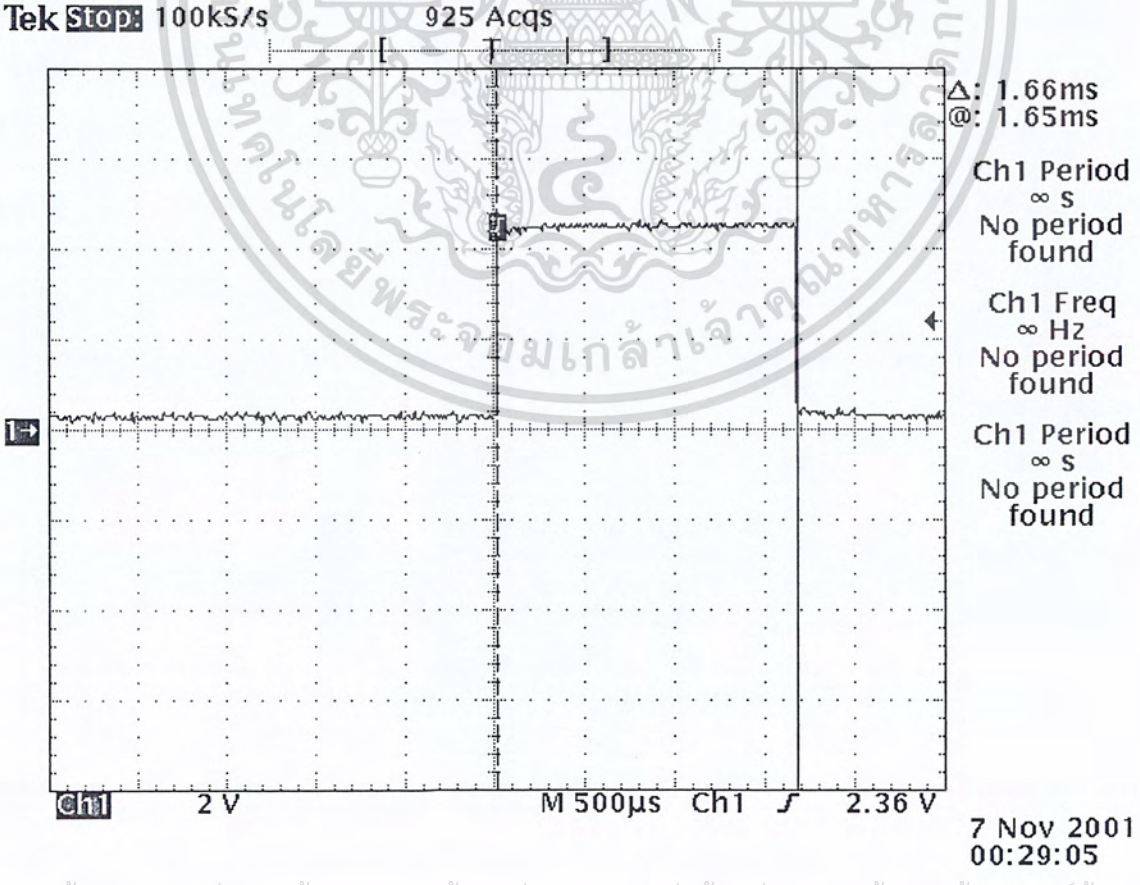
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อนำสัญญาณพัลซ์ ที่มีค่า 1.5 ms มาควบคุมมอเตอร์ ทิศทางในการหมุนของมอเตอร์จะเป็นดังรูปที่ 4.8



รูปที่ 4.8แสดงการทิศการหมุนของเซอร์โวมอเตอร์ขณะ พัลซ์ 1.25 ms

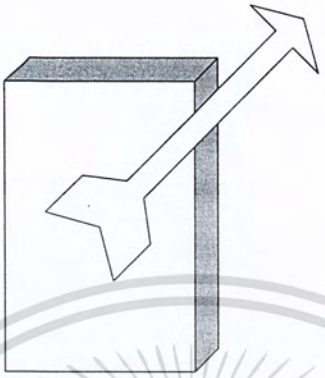
4.1.5 ทำการเขียนโปรแกรมสร้างพัลซ์ 1.65 ms (ส่วนของโปรแกรมสามารถดูได้จากภาคผนวก) ลงในไมโครคอนโทรลเลอร์ จากนั้นใช้ออสซิลโลสโคป จับค่าของสัญญาณพัลซ์ที่ เอาท์พุทของไมโครคอนโทรลเลอร์ว่ามีค่าเท่าใดซึ่งได้ผลดังรูปที่ 4.9



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.9แสดงรูปคลื่นพัลซ์ขนาด 1.65 ms

เมื่อนำสัญญาณพัลซ์ ที่มีค่า 1.5 ms มาควบคุมมอเตอร์ ทิศทางในการหมุนของมอเตอร์จะเป็นดังรูปที่ 4.10



รูปที่ 4.10 แสดงการทิศการหมุนของเซอร์โวมอเตอร์ขณะ พัลซ์ 1.65ms

4.2 บันทึกผลการทดลองที่ได้ลงในตาราง

สร้างโปรแกรมที่ใช้พัลซ์ขนาด	ทิศทางาการหมุนของเซอร์โวมอเตอร์	องศาที่หมุนไปโดยประมาณ
1ms	ทวนเข็มนาฬิกา	90
1.25ms	ทวนเข็มนาฬิกา	45
1.5ms	อยู่ตรงกลาง	0
1.65ms	ตามเข็มนาฬิกา	45
2ms	ตามเข็มนาฬิกา	90

ตารางที่ 4.1 แสดงการทิศทางการหมุนของมอเตอร์และองศาที่เปลี่ยนไป

4.3 สรุปผลการทดลอง

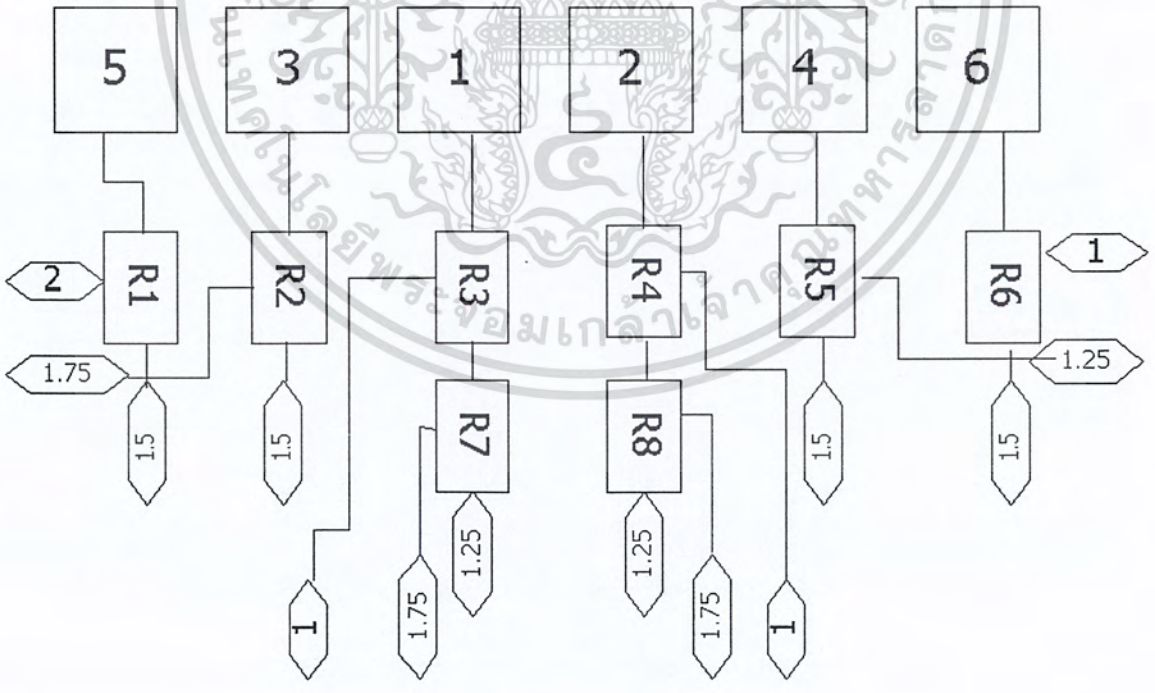
จากการทดลอง เราสามารถสรุปได้ว่า ในการควบคุมการหมุนของเซอร์โวมอเตอร์นั้นเราสามารถใช้พัลซ์เข้ามาเป็นตัวควบคุมโดยหากเราต้องการให้หมุนทวนเข็มนาฬิกาเราก็ให้ค่าพัลซ์มีค่าตั้งแต่ 1ms ไปจนถึง 1.5 ms หากเราต้องการให้เซอร์โวมอเตอร์อยู่ที่ 90 องศาเราก็ทำได้โดยการป้อนพัลซ์ที่มีค่า 1.5 ms ให้แก่เซอร์โวมอเตอร์ และหากเราต้องการให้เซอร์โวมอเตอร์หมุนไปในทิศทางตามเข็มนาฬิกาเราก็ควรให้พัลซ์ ที่มีค่าตั้งแต่ 1.5 ms ไปจนถึง 2ms โดยประมาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
การวิเคราะห์

การวิเคราะห์ในบทนี้นั้น เราจะนำทฤษฎีในบทที่ผ่านมา มาทำการวิเคราะห์ว่าสามารถจะทำอย่างไรให้หุ่นยนต์สามารถเดินได้ ส่วนใหญ่บทนี้จะเน้นขั้นตอนการเดิน เพราะเราจะต้องวิเคราะห์ว่าหุ่นยนต์ควรเดินอย่างไร เพื่อที่จะได้นำไปใช้เป็นรูปแบบในการเขียนโปรแกรมควบคุมได้ จากบทที่ผ่านมาเราสามารถสรุปได้โดยคร่าวว่า ในการที่จะควบคุมเซอร์โวมอเตอร์นั้นเราสามารถทำได้โดยการกำหนดขนาดของสัญญาณพัลซ์ ดังนั้นในการที่เราจะทำการควบคุมหุ่นยนต์ให้เดินไปในทางใด เราก็ทำการควบคุม การหมุนของเซอร์โวมอเตอร์ให้เป็นไปในทิศทางที่จะสามารถทำให้ส่วนประกอบส่วนใดของหุ่นยนต์ หมุนไปในทิศทางที่เราต้องการได้

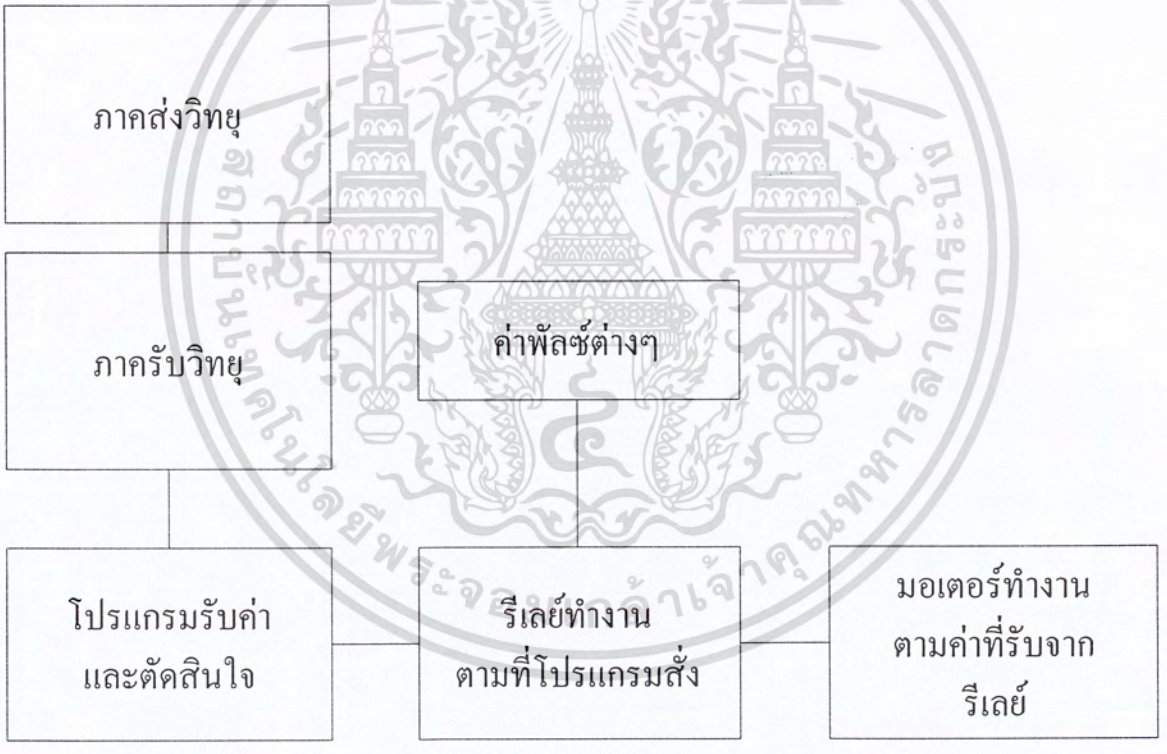
ในการกำหนดให้ เซอร์โวมอเตอร์ตัวใดทำงาน เราได้ใช้รีเลย์ เป็นทางผ่านของสัญญาณ พัลซ์ที่เราต้องการ โดยเราได้สร้างพัลซ์รอเอาไว้แล้ว จากนั้นก็ให้โปรแกรมทำงานว่า ต้องการให้เซอร์โวมอเตอร์ตัวใดทำงานก็จะ ไปทริกรีเลย์ให้จ่ายพัลซ์ที่ต้องการไปยังเซอร์โวมอเตอร์แต่ละตัว



รูปที่ 5.1 การต่อรีเลย์ให้กับมอเตอร์แต่ละตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อมีสัญญาณมาจากเครื่องส่ง สัญญาณนั้นจะถูกรับและเก็บค่าพร้อมกับเช็คดูว่า เป็นสัญญาณที่จะส่งไปให้โปรแกรมใดบ้าง จากนั้นตัวโปรแกรมก็จะทำการ เรียกโปรแกรมต่างๆที่เราได้ทำการ เขียนเป็นโปรแกรมย่อยเอาไว้แล้ว และสั่งงานให้โปรแกรมย่อยนั้นไปควบคุม การทำงานของรีเลย์ อีกทีหนึ่ง โดยที่ รีเรย์ได้รับค่าของพัลซ์ รอเอาไว้อยู่แล้วขึ้นอยู่กับว่าจะให้หน้าสัมผัสของรีเลย์ เชื่อมต่อกับรูปคลื่นใด หลังจากนั้น ก็ต่อเข้ากับมอเตอร์แต่ละตัว เพื่อให้ค่าของพัลซ์เหล่านั้นไป ส่งค่าให้เซอร์โวมอเตอร์ทำงานอีกทีหนึ่ง ดังรูปแสดงการทำงาน



รูปที่ 5.2 แสดงบล๊อกการทำงานของมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
สรุปโครงงานและปัญหาการทำโครงงาน

6.1 สรุปโครงงาน

ในการสร้างหุ่นยนต์ไต่กำแพงนั้นมีส่วนที่สำคัญๆได้แก่ ภาครับ-ส่ง วิทยุ เพื่อใช้ในการเป็นสัญญาณกระตุ้นให้เกิดการทำงานในระบบ ส่วนที่สำคัญที่ควรทราบก็คือการทำงาน ของเซอร์โวมอเตอร์ที่จะทำการหมุนไปในทิศทางทวนเข็มนาฬิกา ที่ ช่วงของพัลซ์ 1-1.5 ms และจะหยุดอยู่ในช่วงกึ่งกลางของการหมุนที่พัลซ์มีขนาด 1.5 ms และจะมีทิศทางของการหมุนไปในทิศทางตามเข็มนาฬิกาที่ช่วงของพัลซ์ 1.5-2.0 ms ทำให้เราสามารถที่จะกำหนดได้ว่า เซอร์โวมอเตอร์ตัวที่เราต้องการให้หมุนไปในทิศทางใดๆก็ตาม เราควรจะใช้พัลซ์ช่วงใด ซึ่งสัญญาณพัลซ์ที่เราจะใช้นั้นเราก็สามารถสร้างขึ้นมาได้โดยการใช้ไมโครคอนโทรเลอร์

อุปกรณ์ไมโครคอนโทรลเลอร์นั้น นอกจากใช้ผลิตพัลซ์แล้วยังเป็นเหมือนกับสมองสั่งการโดยจะทำการคิดประมวลผลตามโปรแกรมที่เราได้ทำการเขียนคำสั่งเอาไว้แล้ว ซึ่งหลังจากที่ไมโครคอนโทรลเลอร์ประมวลผลเสร็จแล้วก็จะให้ ลอจิกออกมาเป็นสัญญาณมาควบคุมอุปกรณ์ต่างๆโดยผ่านภาคขยายเพื่อให้มีกำลังพอที่จะไปควบคุมอุปกรณ์ต่างๆได้

ในการเขียนโปรแกรม เราจะแบ่งเป็นโปรแกรมย่อยๆหลายๆโปรแกรมแล้วนำโปรแกรมย่อยเหล่านั้นมารวมกันอยู่ในโปรแกรมหลัก เพื่อที่โปรแกรมหลักจะเป็นตัวสั่งให้โปรแกรมย่อยเหล่านั้นทำงานให้ถูกต้องตามหน้าที่

6.2 สรุปปัญหาการทำโครงงาน

6.2.1 ปัญหาเรื่องการรับน้ำหนัก ของเซอร์โวมอเตอร์ เนื่องจาก เซอร์โวมอเตอร์ที่ใช้ ไม่ได้มาตรฐานเพียงพอ ทำให้น้ำหนักที่ใช้ได้จริงกับน้ำหนัก ตามเสปคของเซอร์โวมอเตอร์ แตกต่างกันและยังต้องรวมไปถึงค่าของทอร์คภายในเซอร์โวมอเตอร์ด้วย จึงทำให้ เซอร์โวมอเตอร์เสียหาย

6.2.2 ปัญหาความไม่เสถียรของ มือหุ่นยนต์ที่ใช้จับตาข่ายลวด เนื่องจากว่าในการจับตาข่ายลวดนั้น ส่วนที่เป็นมือของหุ่นยนต์ นั้นมักจะเกิดการชนกันกับเส้นลวด ทำให้มือไม่สามารถที่จะทำการจับ ตาข่ายลวดได้อย่างมั่นคง ตรงนี้เองทำให้เกิดปัญหาที่ว่า หุ่นยนต์ โคลงเคลง หรือจับไม่ได้ ก็จะทำให้หุ่นยนต์หลุดวงโคจรในการจับไป

6.2.3 การออกแบบตัวหุ่นยนต์ ผิดพลาด ทำให้ต้องออกแบบบ่อยๆ เพราะว่ายังไม่มีต้นแบบที่ทำมาก่อน เราจึงต้องออกแบบขึ้นมาเองก่อน โดยคาดว่าน่าจะสำเร็จ แต่หลักการ กับการปฏิบัติจริงนั้นให้ผลที่ต่างกัน ทำให้หลักการที่ได้คิดไว้ ไม่สามารถทำได้จริง เนื่องจากมีองค์ประกอบอื่นๆ เข้ามาเกี่ยวข้อง ทำให้หุ่นยนต์ที่ได้รับการออกแบบนั้น ไม่สามารถเดินได้ตามความคิดแต่แรก จึงต้องทำการเปลี่ยน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แปลงแบบอยู่หลายครั้ง และเมื่อต้องทำการเปลี่ยนแปลงแบบที่เราใช้อยู่ บ่อยๆ ก็ทำให้เราต้องเปลี่ยน วงจรต่างๆ บ่อยยิ่งขึ้น และโปรแกรมที่ใช้ก็บ่อยขึ้นมากเช่นกัน

6.2.4 ปัญหาการขาดต้นแบบในการศึกษา เนื่องจากไม่มีต้นแบบที่จะใช้ในการศึกษาเลย ดังนั้น ระยะเวลาส่วนใหญ่ จะหมดไปกับการคิดคำนวณต้นแบบ ซึ่งทำให้เสียเวลา และ ค่าใช้จ่าย เป็นจำนวนมาก

6.2.5 ปัญหาด้านการหาอุปกรณ์ เนื่องจากอุปกรณ์ที่ใช้ มีราคาสูง หากเกิดการเสียหายขึ้นมา จำเป็นต้องรอเป็นระยะ เวลานาน เพื่อที่จะหาทุนไปซื้ออุปกรณ์ ตัวใหม่ ทำให้เสียเวลาในการรอมากกว่าครึ่งหนึ่งของระยะเวลา การทำงานทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

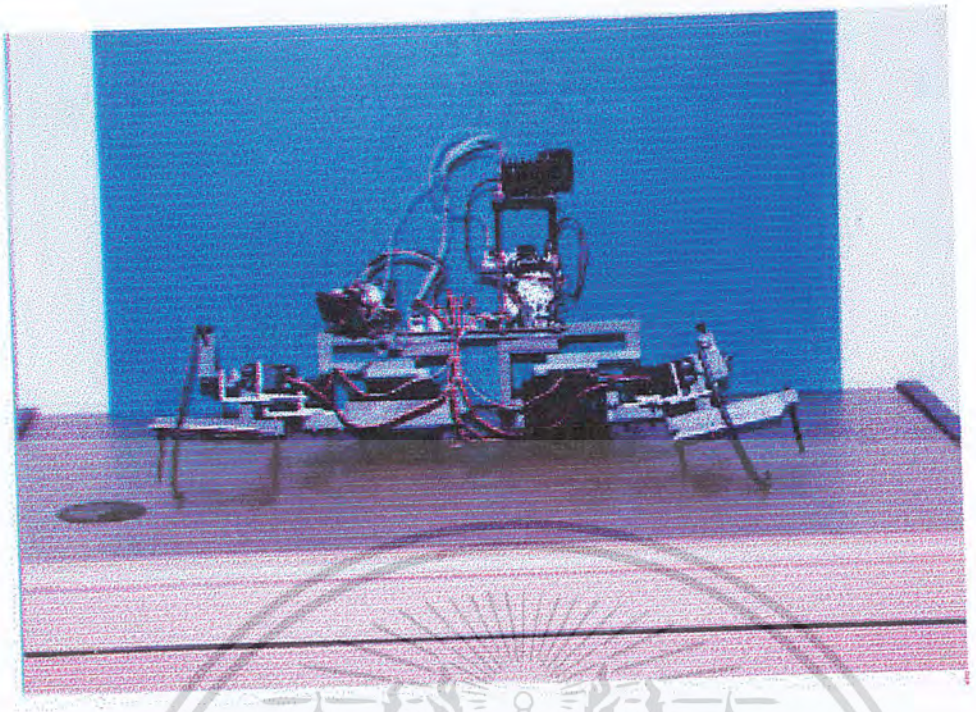
ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

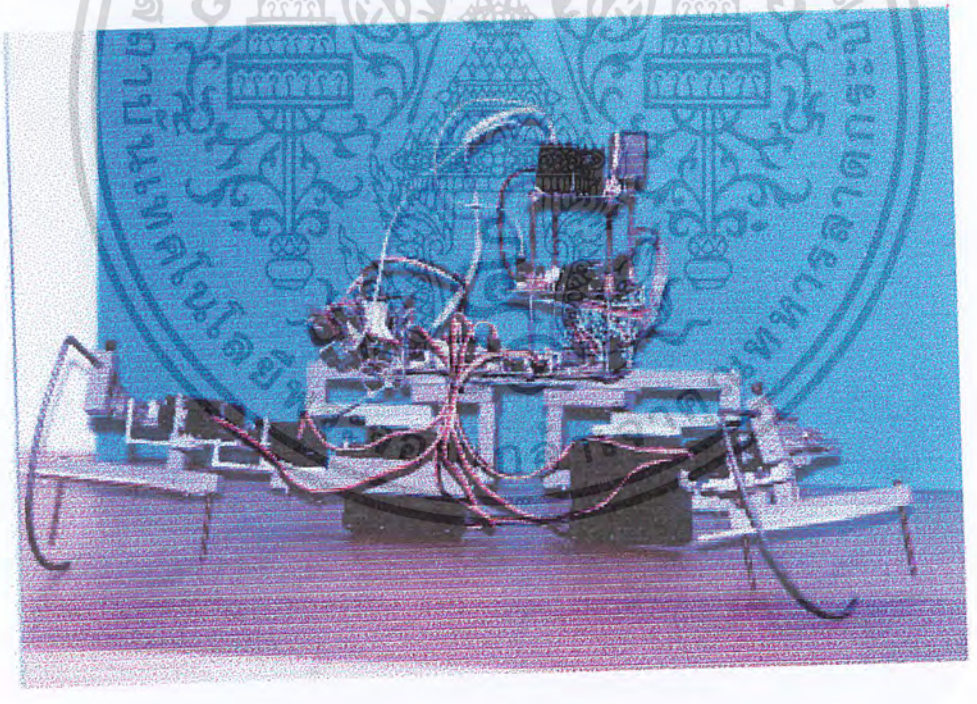
ภาคผนวก ก.

รูปโครงสร้างหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

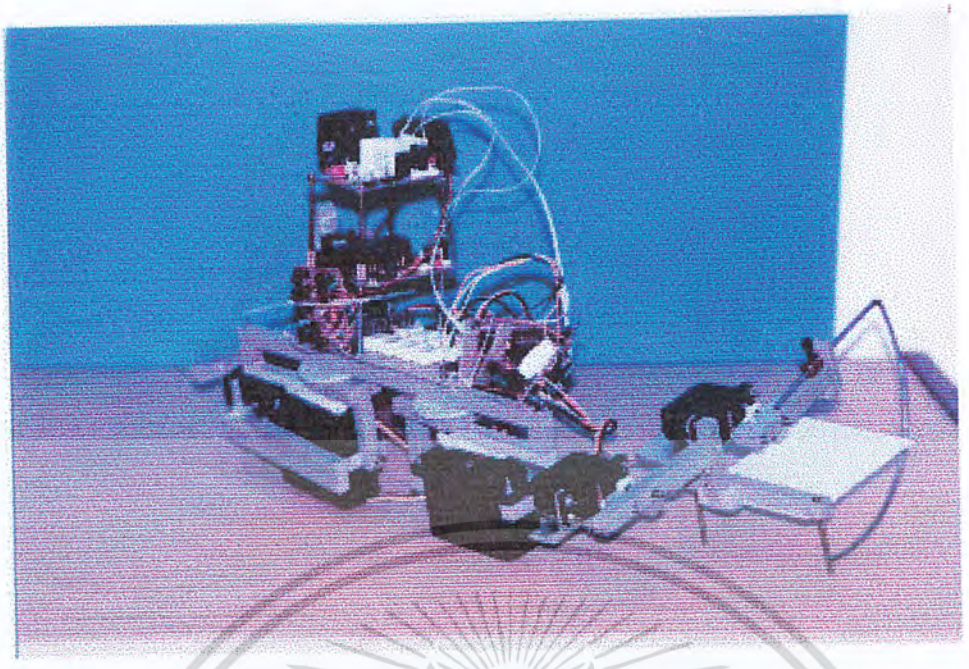


รูปที่ ก.1 ภาพถ่ายจากด้านหน้าของหุ่นยนต์

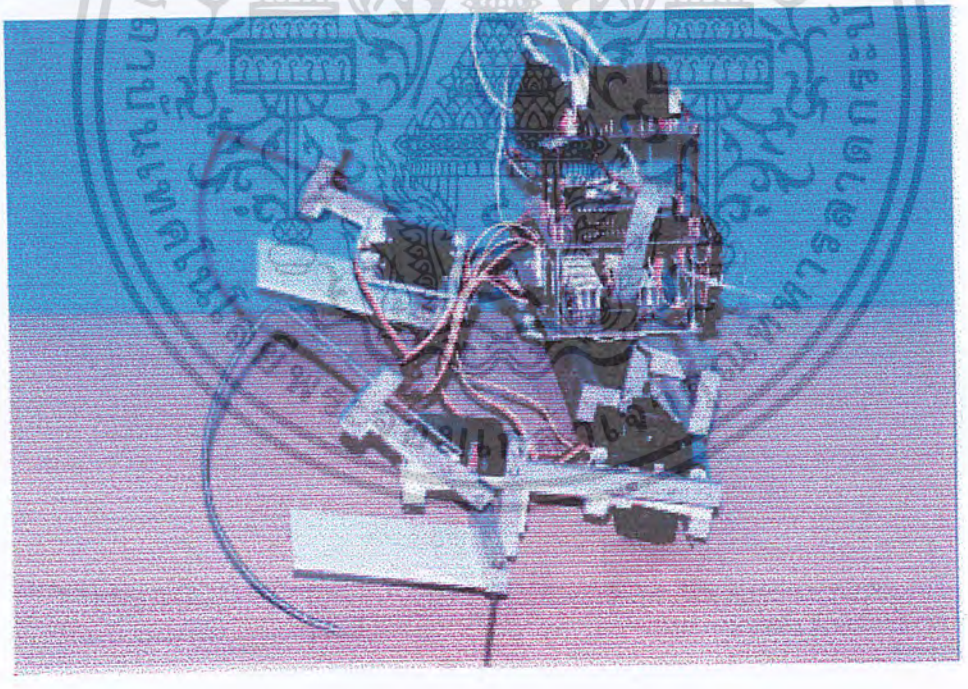


รูปที่ ก.2 ภาพถ่ายมุมเฉียงของหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

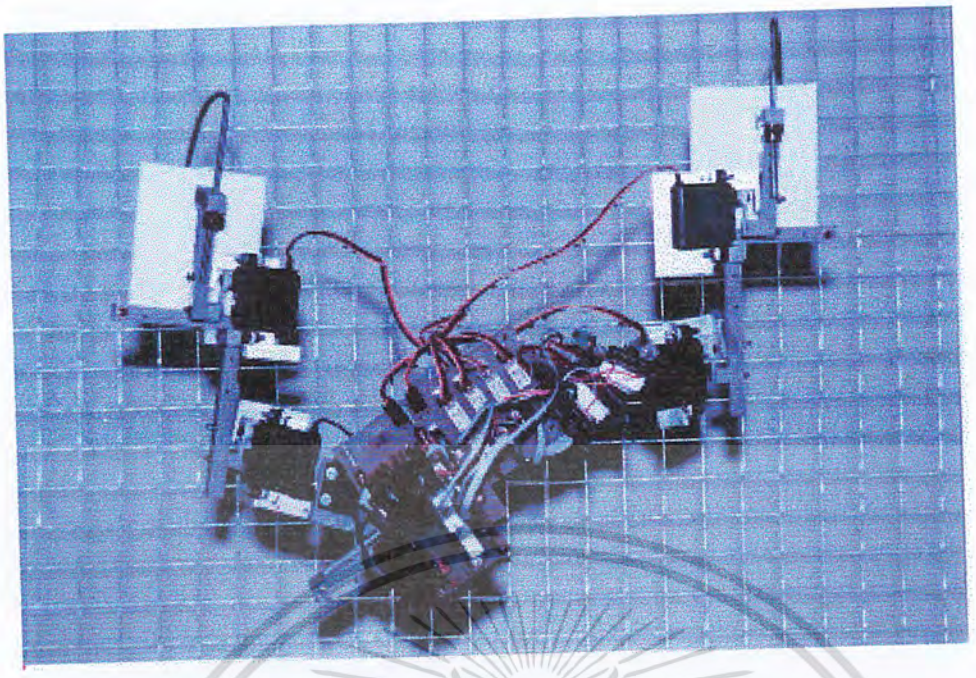


รูปที่ ก.3 ภาพถ่ายเฉียงด้านหลังของหุ่นยนต์

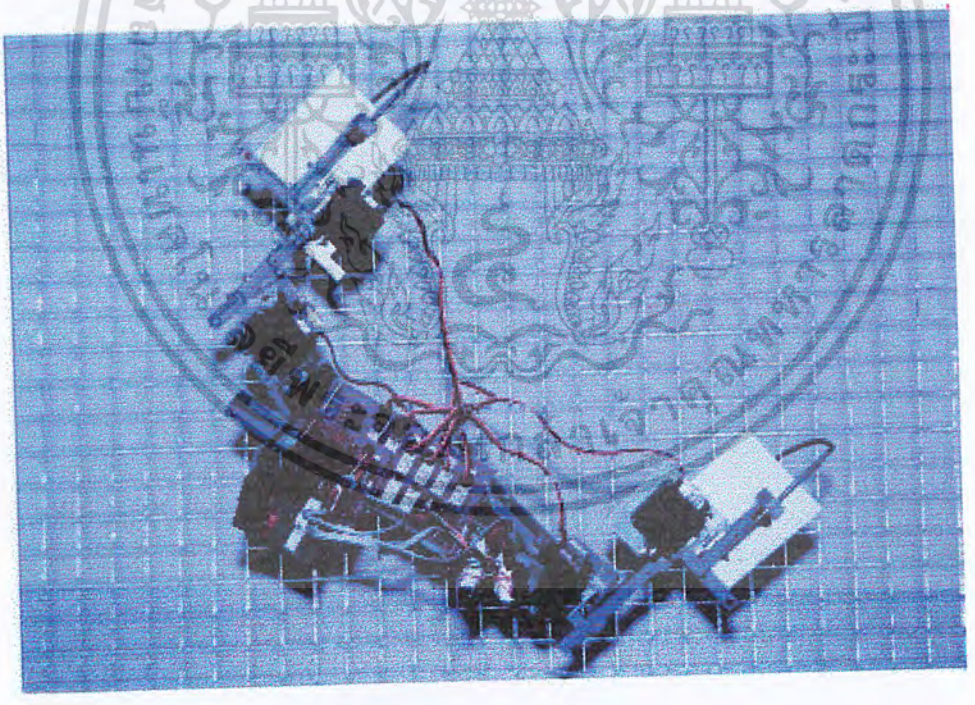


รูปที่ ก.4 ภาพถ่ายด้านข้างของหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.5 ภาพถ่ายขณะหุ่นยนต์กำลังไต่กำแพงตาข่าย



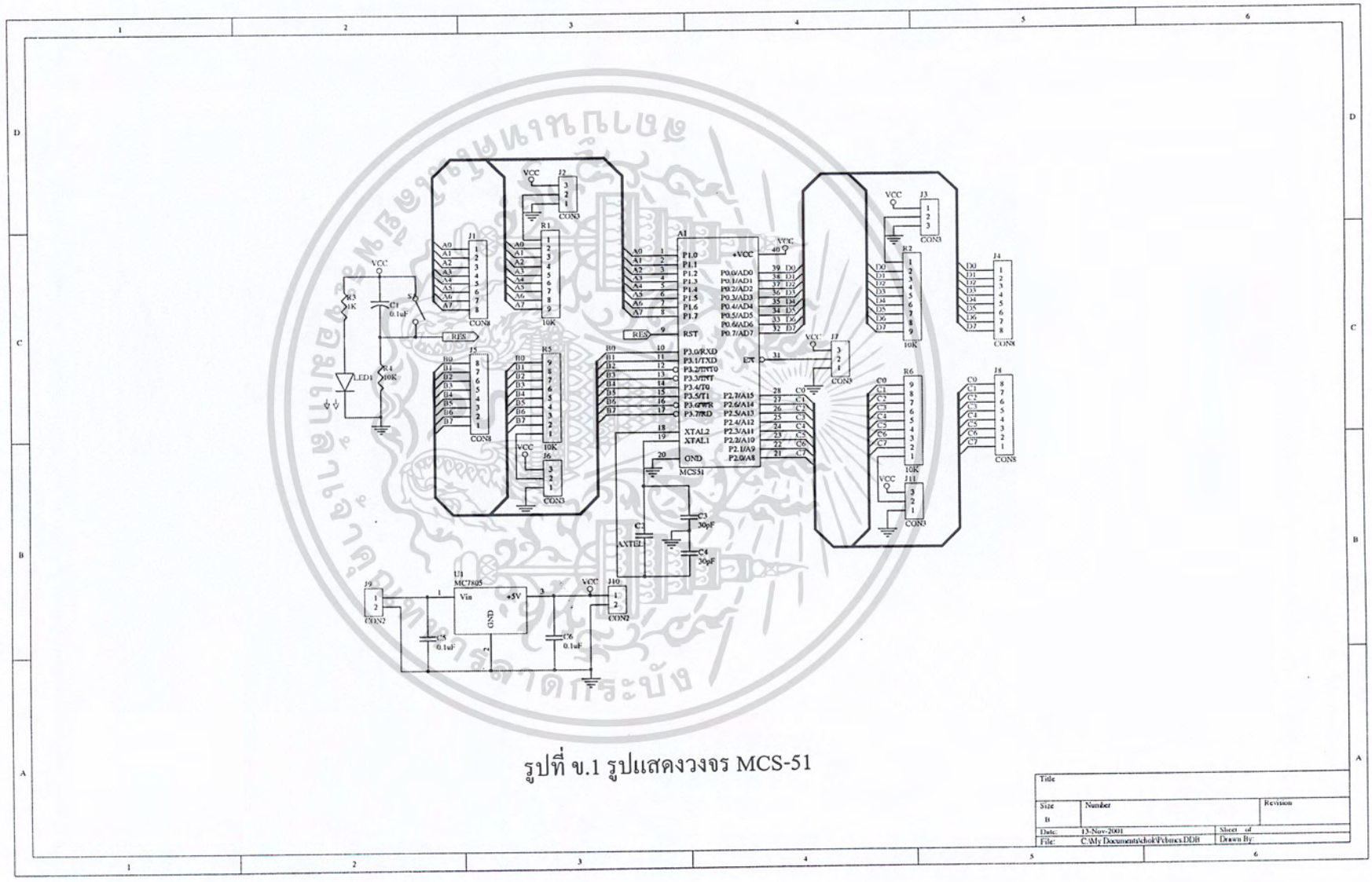
รูปที่ ก.6 ขณะหุ่นยนต์กำลังทำก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

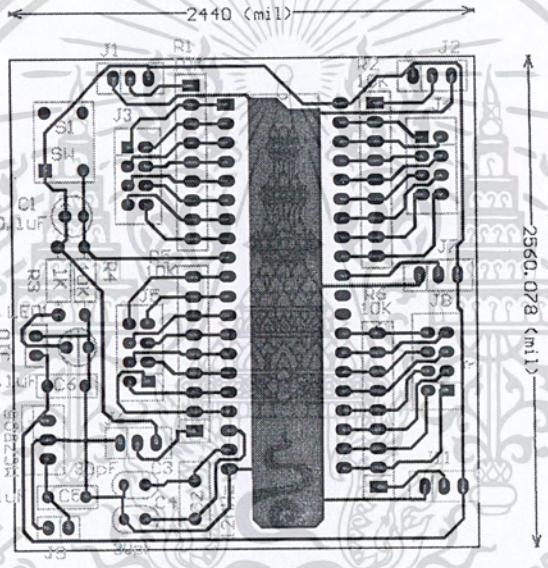
ภาคผนวก ข.

วงจรอิเล็กทรอนิกส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

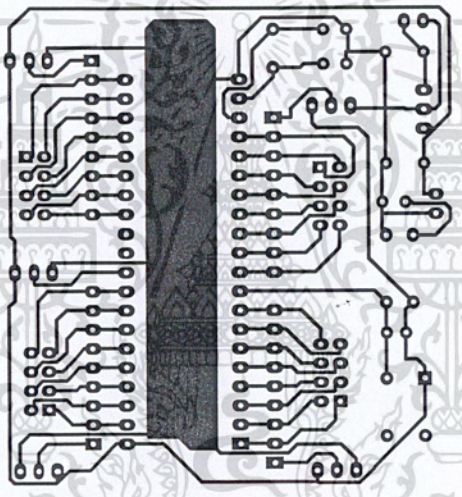


รูปที่ ข.1 รูปแสดงวงจร MCS-51



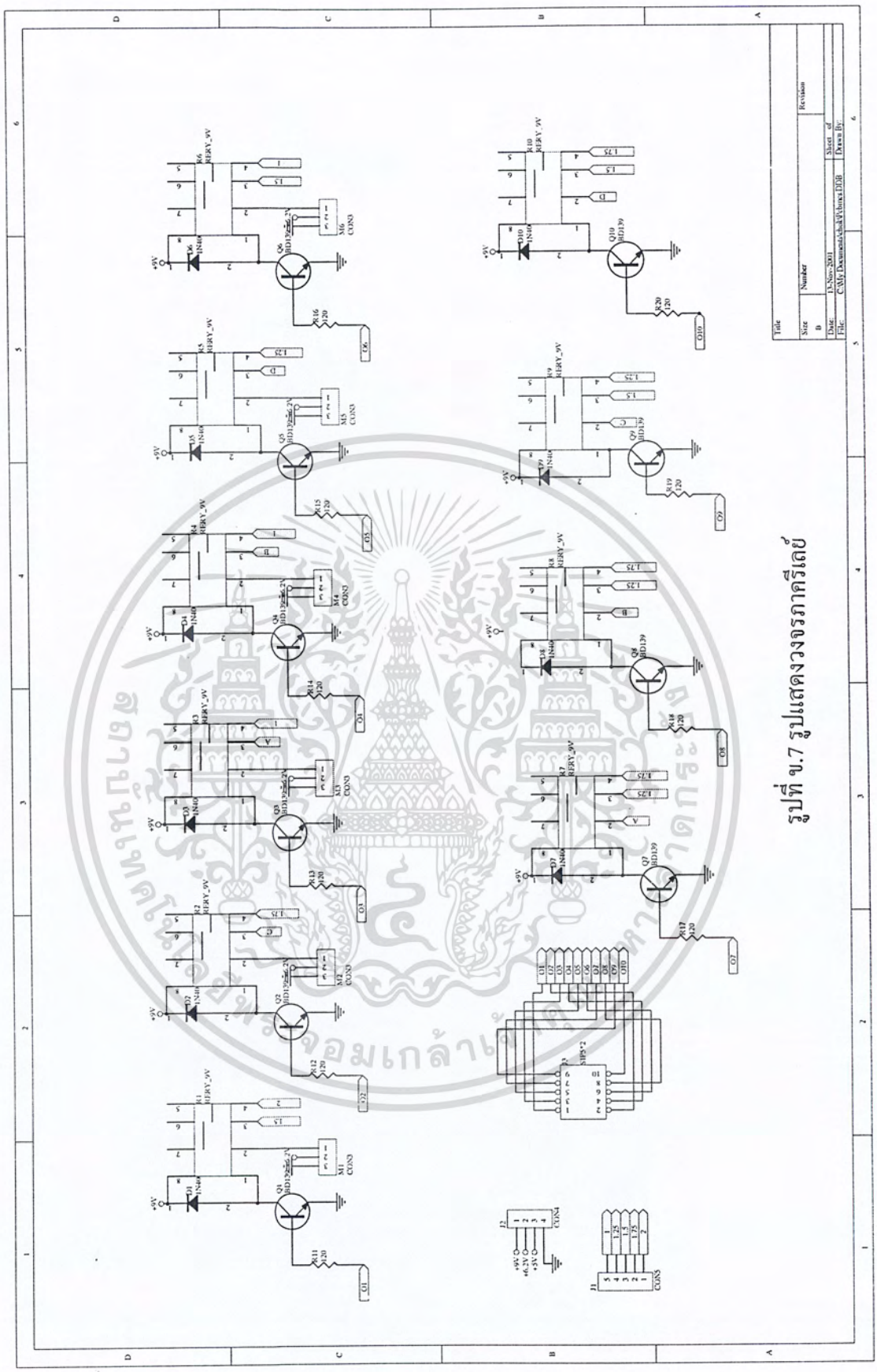
รูปที่ ข.2 รูปแสดงอุปกรณ์ด้านบนลายทองแดง MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



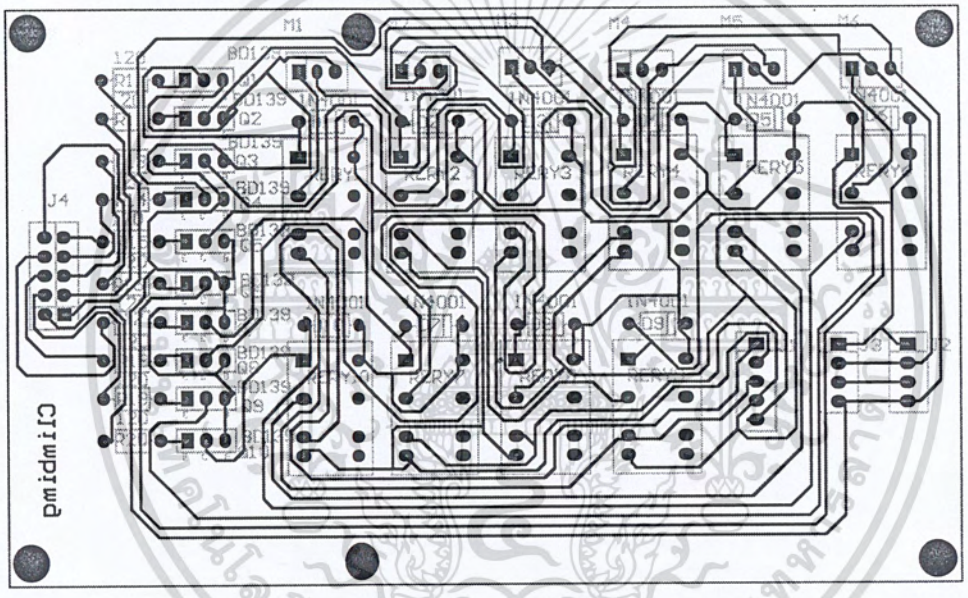
รูปที่ ข.3 รูปแสดงลายทองแดง MCS-51

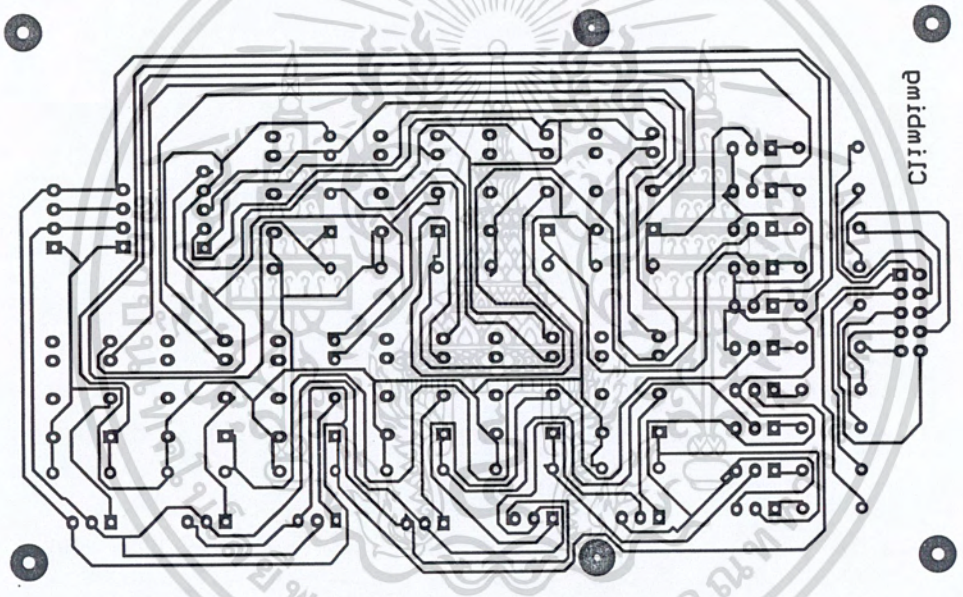
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.7 รูปแสดงวงจรภาครีเลย์

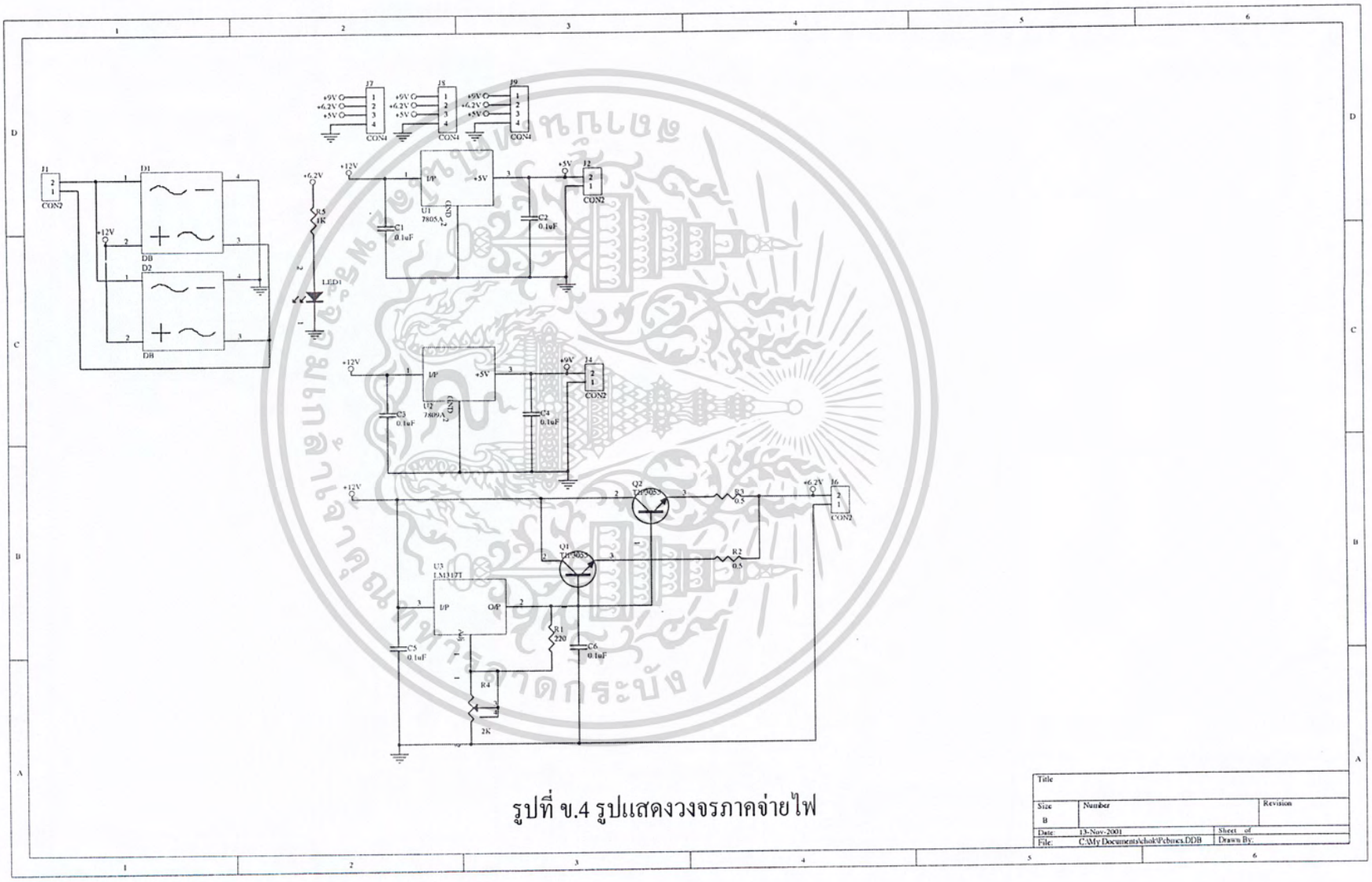
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


M1
M4
M5
M6
J4
D9
D10
D11
D12
D13
D14
D15
D16
D17
D18
D19
D20
D21
D22
D23
D24
D25
D26
D27
D28
D29
D30
D31
D32
D33
D34
D35
D36
D37
D38
D39
D40
D41
D42
D43
D44
D45
D46
D47
D48



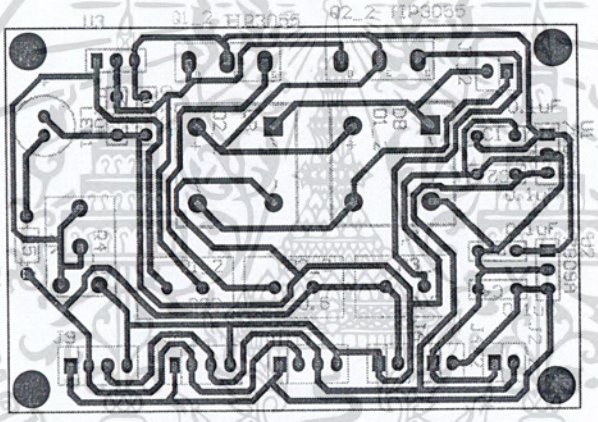
รูปที่ ข.9 รูปแสดงลายทองแดงภาครีเลย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



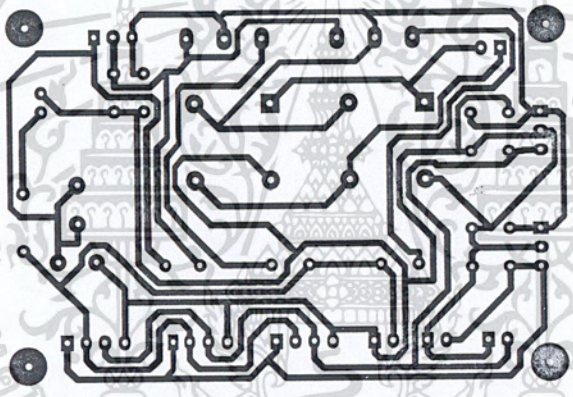
รูปที่ ข.4 รูปแสดงวงจรภาคจ่ายไฟ

ลิขสิทธิ์ของมหาวิทยาลัยราชภัฏราชนครินทร์ ไม่อนุญาตให้ทำซ้ำ ดัดแปลง เผยแพร่ หรือกระทำการใดๆ ในรูปแบบอื่นใดแก่งานนี้ นอกจากจะได้รับอนุญาตจากมหาวิทยาลัยราชภัฏราชนครินทร์ก่อนเท่านั้น



รูปที่ ข.5รูปแสดงอุปกรณ์ด้านบนลายทองแดงภาคจ่ายไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.6 รูปแสดงลายทองแดงภาคจ่ายไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค.

โปรแกรมควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        ORG         0000H

                        AJMP        MAIN

            DELAY       EQU         4
;           DATA DELAY
                        ORG         000BH
                        LJMP        T0_INT

MAIN:                   MOV         DPTR,#07FFH
;           POWER ON DELAY
POR:                    INC         DPTR
                        MOV         A,DPH
                        JNZ         POR
;
                        MOV         P1,#0
;           INITIAL     PORT
                        MOV         SP,#100
                        ;
                        MOV         TMOD,#022H
;           INITIAL     TIMER
                        MOV         TH0,#-230
                        MOV         TL0,#-230
                        ;
                        MOV         R2,#200
;           INITIAL     DELAY TIME
                        MOV         R1,#20
                        MOV         R0,#DELAY
LL:                     CPL         P1.0
                        ;SJMP       LL
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                        ;
                                        SETB        EA
        ;       INITIAL     INTERRUPT
                                        SETB        ET0
                                        SETB        TR0
                                        ;

                                        MOV         P1,#0FFH
CHK:                                    MOV         A,P1
                                        CPL         A
                                        CJNE        A,#1,C1
                                        AJMP        FORWARD
C1:                                     CJNE        A,#2,C2
                                        AJMP        REWARD
C2:                                     CJNE        A,#4,C3
                                        AJMP        T_LEFT
C3:                                     CJNE        A,#8,C4
                                        AJMP        T_RIGHT
C4:                                     AJMP        CHK

;*************************************************************

FORWARD:                                ACALL       INITIAL_TIME
        ;p0     p2
                                        SETB        TR0

                                        MOV             p0,#0
                                        MOV             P2,#0
                                        CLR             TF0
                                        JNB             TF0,$
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SETB    P0.0
CLR     TF0
JNB     TF0,$

SETB    P0.1
CLR     TF0
JNB     TF0,$

MOV     P0,#11000011B
CLR     TF0
JNB     TF0,$

CLR     P0.1
SETB    P2.0
CLR     TF0
JNB     TF0,$

CLR     P0.0
CLR     TF0
JNB     TF0,$

CLR     P2.0
CLR     TF0
JNB     TF0,$

SETB    P0.5
CLR     TF0
JNB     TF0,$

SETB    P0.4
CLR     TF0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                JNB     TF0,$

                MOV     P0,#00001100B
                CLR     TF0
                JNB     TF0,$

                CLR     P0.4
                SETB    P2.1
                CLR     TF0
                JNB     TF0,$

                CLR     P0.5
                CLR     TF0
                JNB     TF0,$

                AJMP    CHK
;***************************************************************

REWARD:         ACALL   INITIAL_TIME

                MOV     P0,#0
                MOV     P2,#0
                CLR     TF0
                JNB     TF0,$

                SETB    P0.5
                CLR     TF0
                JNB     TF0,$

                SETB    P0.4
                CLR     TF0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
JNB     TF0,$

MOV     P1,#0FH
CLR     TF0
JNB     TF0,$

CLR     P1.4
SETB    P2.1
CLR     TF0
JNB     TF0,$

CLR     P1.5
CLR     TF0
JNB     TF0,$

CLR     P2.1
CLR     TF0
JNB     TF0,$

SETB    P1.0
CLR     TF0
JNB     TF0,$

SETB    P1.1
CLR     TF0
JNB     TF0,$

MOV     P0,#03H
CLR     TF0
JNB     TF0,$
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    CLR         P0.1
                    SETB        P2.0
                    CLR         TF0
                    JNB         TF0,$

                    CLR         P0.0
                    CLR         TF0
                    JNB         TF0,$

                    AJMP        CHK

;*************************************************************

T_RIGHT:            ACALL       INITIAL_TIME

                    MOV         P0,#0
                    MOV         P2,#0
                    CLR         TF0
                    JNB         TF0,$

                    SETB        P0.5
                    CLR         TF0
                    JNB         TF0,$

                    SETB        P0.4
                    CLR         TF0
                    JNB         TF0,$

                    MOV         P0,#0F0H
                    CLR         TF0
                    JNB         TF0,$
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CLR             P0.4
SETB    P2.1
CLR             TF0
JNB             TF0,$

CLR             P0.5
CLR             TF0
JNB             TF0,$

CLR             P2.1
CLR             TF0
JNB             TF0,$

SETB    P0.0
CLR             TF0
JNB             TF0,$

SETB    P0.1
CLR             TF0
JNB             TF0,$

MOV             P1,#087H
CLR             TF0
JNB             TF0,$

CLR             P0.1
SETB    P2.0
CLR             TF0
JNB             TF0,$

CLR             P0.0
CLR             TF0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
JNB     TF0,$

CLR     P2.0
CLR     TF0
JNB     TF0,$

SETB    P0.5
CLR     TF0
JNB     TF0,$

SETB    P0.4
CLR     TF0
JNB     TF0,$

CLR     P0.7
CLR     TF0
JNB     TF0,$

CLR     P0.4
SETB    P2.1
CLR     TF0
JNB     TF0,$

CLR     P0.5
CLR     TF0
JNB     TF0,$

CLR     P0.1
CLR     TF0
JNB     TF0,$

SETB    P0.0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                CLR             TF0
                JNB             TF0,$

                SETB    P0.1
                CLR             TF0
                JNB             TF0,$

                CLR             P0.2
                CLR             TF0
                JNB             TF0,$

                CLR             P0.1
                SETB    P2.0
                CLR             TF0
                JNB             TF0,$

                CLR             P0.0
                CLR             TF0
                JNB             TF0,$

                AJMP    CHK
;*********************************************************

T_LEFT:                         ACALL   INITIAL_TIME

                MOV             P0,#0
                MOV             P2,#0
                CLR             TF0
                JNB             TF0,$
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SETB    P0.5
CLR     TF0
JNB     TF0,$

SETB    P0.4
CLR     TF0
JNB     TF0,$

SETB    P0.3
CLR     TF0
JNB     TF0,$

CLR     P0.4
SETB    P2.1
CLR     TF0
JNB     TF0,$

CLR     P0.5
CLR     TF0
JNB     TF0,$

CLR     P2.1
CLR     TF0
JNB     TF0,$

SETB    P0.0
CLR     TF0
JNB     TF0,$

SETB    P1.1
CLR     TF0
JNB     TF0,$
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SETB    P0.6
CLR     TF0
JNB     TF0,$

CLR     P0.1
SETB    P2.0
CLR     TF0
JNB     TF0,$

CLR     P0.0
CLR     TF0
JNB     TF0,$

CLR     P2.0
CLR     TF0
JNB     TF0,$

SETB    P0.5
CLR     TF0
JNB     TF0,$

SETB    P0.4
CLR     TF0
JNB     TF0,$

MOV     P0,#30H
CLR     TF0
JNB     TF0,$
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    CLR         P0.6
                    SETB        P2.1
                    CLR         TF0
                    JNB         TF0,$

                    CLR         P0.7
                    CLR         TF0
                    JNB         TF0,$

                    AJMP        CHK

;******************  TIMER 0 INTERRUPT  *****************

T0_INT:             CLR         TF0
                    DJNZ        R2,E_T0
                    MOV         R2,#200
                    DJNZ        R1,E_T0
                    MOV         R1,#20
                    CPL         P0.0
                    DJNZ        R0,E_T0
                    MOV         R0,#DELAY
                    SETB        F0
                    CPL         P0.1
E_T0:               RETI

;**********************  INITIAL  TIMER ******************

INITIAL_TIME:       MOV         R2,#200
                    MOV         R1,#20
                    MOV         R0,#DELAY
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RET

;**********

END

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง.

รายละเอียดของ MCS -52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Features

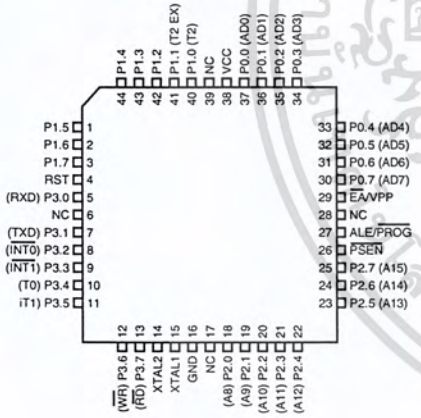
- Compatible with MCS-51™ Products
- 8K Bytes of In-System Reprogrammable Flash Memory
- Endurance: 1,000 Write/Erase Cycles
- Fully Static Operation: 0 Hz to 24 MHz
- Three-level Program Memory Lock
- 256 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Three 16-bit Timer/Counters
- Eight Interrupt Sources
- Programmable Serial Channel
- Low-power Idle and Power-down Modes

Description

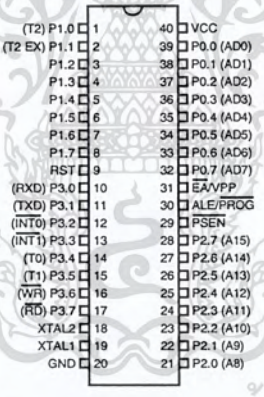
The AT89C52 is a low-power, high-performance CMOS 8-bit microcomputer with 8K bytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard 80C51 and 80C52 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C52 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications.

Pin Configurations

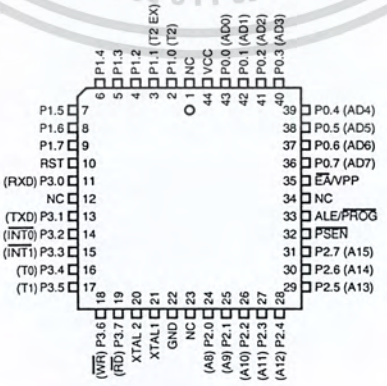
PQFP/TQFP



PDIP



PLCC



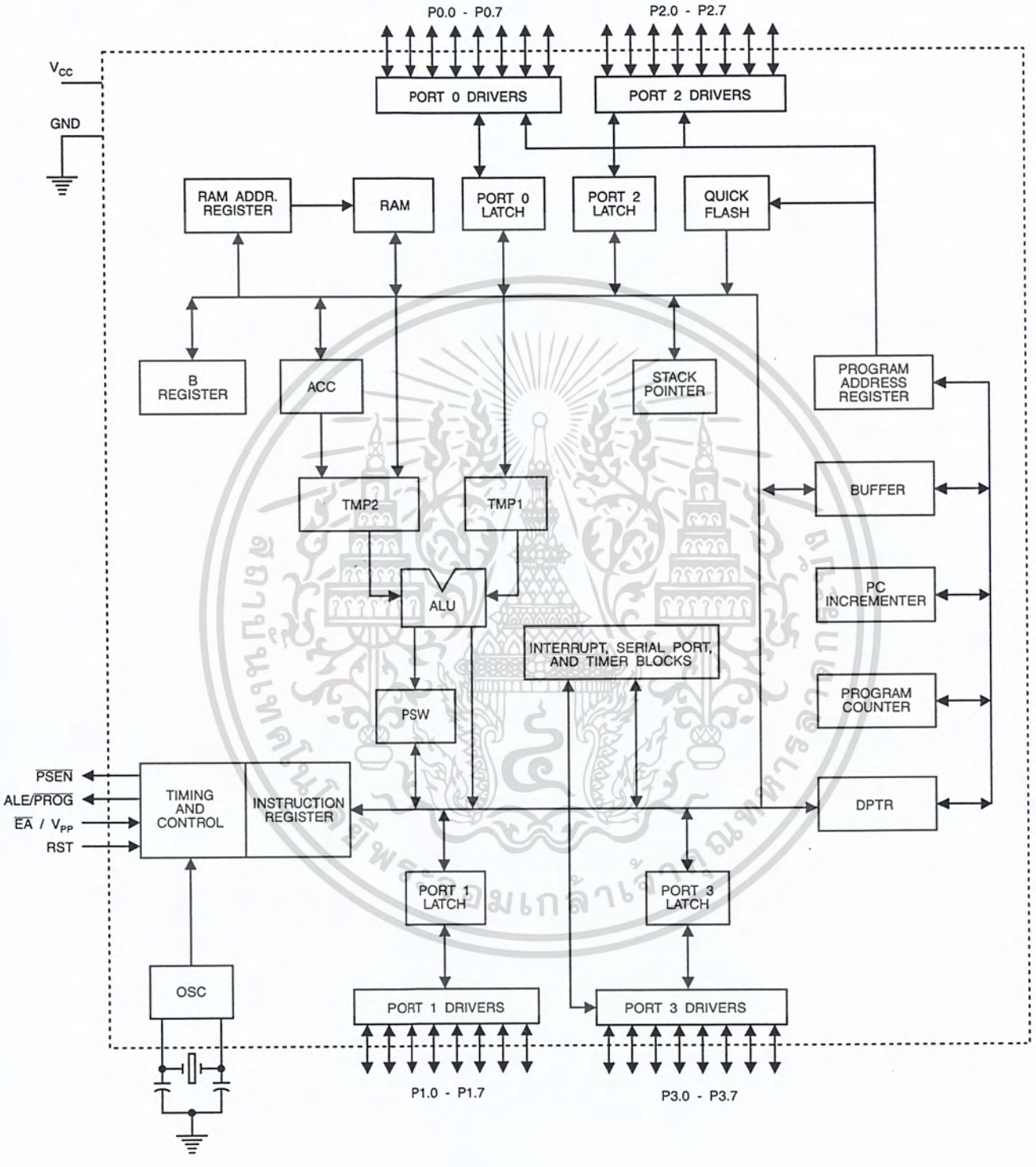
8-bit Microcontroller with 8K Bytes Flash

AT89C52

Rev. 0313H–02/00

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The AT89C52 provides the following standard features: 8K bytes of Flash, 256 bytes of RAM, 32 I/O lines, three 16-bit timer/counters, a six-vector two-level interrupt architecture, a full-duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89C52 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power-down mode saves the RAM contents but freezes the oscillator, disabling all other chip functions until the next hardware reset.

Pin Description

VCC
Supply voltage.

GND
Ground.

Port 0
Port 0 is an 8-bit open drain bi-directional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs.

Port 0 can also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode, P0 has internal pullups.

Port 0 also receives the code bytes during Flash programming and outputs the code bytes during program verification. External pullups are required during program verification.

Port 1
Port 1 is an 8-bit bi-directional I/O port with internal pullups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

In addition, P1.0 and P1.1 can be configured to be the timer/counter 2 external count input (P1.0/T2) and the timer/counter 2 trigger input (P1.1/T2EX), respectively, as shown in the following table.

Port 1 also receives the low-order address bytes during Flash programming and verification.

Port Pin	Alternate Functions
P1.0	T2 (external count input to Timer/Counter 2), clock-out
P1.1	T2EX (Timer/Counter 2 capture/reload trigger and direction control)

Port 2
Port 2 is an 8-bit bi-directional I/O port with internal pullups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application, Port 2 uses strong internal pullups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

Port 3
Port 3 is an 8-bit bi-directional I/O port with internal pullups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}) because of the pullups.

Port 3 also serves the functions of various special features of the AT89C51, as shown in the following table.

Port 3 also receives some control signals for Flash programming and verification.

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{INT0}$ (external interrupt 0)
P3.3	$\overline{INT1}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{WR}$ (external data memory write strobe)
P3.7	$\overline{RD}$ (external data memory read strobe)

RST
Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

ALE/$\overline{PROG}$
Address Latch Enable is an output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input ($\overline{PROG}$) during Flash programming.

In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency and may be used for external

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external data memory.

If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

$\overline{\text{PSEN}}$

Program Store Enable is the read strobe to external program memory.

When the AT89C52 is executing code from external program memory, $\overline{\text{PSEN}}$ is activated twice each machine cycle, except that two $\overline{\text{PSEN}}$ activations are skipped during each access to external data memory.

$\overline{\text{EA}}$/VPP

External Access Enable. $\overline{\text{EA}}$ must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, $\overline{\text{EA}}$ will be internally latched on reset.

$\overline{\text{EA}}$ should be strapped to V_{CC} for internal program executions.

This pin also receives the 12-volt programming enable voltage (V_{PP}) during Flash programming when 12-volt programming is selected.

XTAL1

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

XTAL2

Output from the inverting oscillator amplifier.

Table 1. AT89C52 SFR Map and Reset Values

0F8H								0FFH
0F0H	B 00000000							0F7H
0E8H								0EFH
0E0H	ACC 00000000							0E7H
0D8H								0DFH
0D0H	PSW 00000000							0D7H
0C8H	T2CON 00000000	T2MOD XXXXXX00	RCAP2L 00000000	RCAP2H 00000000	TL2 00000000	TH2 00000000		0CFH
0C0H								0C7H
0B8H	IP XX000000							0BFH
0B0H	P3 11111111							0B7H
0A8H	IE 0X000000							0AFH
0A0H	P2 11111111							0A7H
98H	SCON 00000000	SBUF XXXXXXXX						9FH
90H	P1 11111111							97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000		8FH
80H	P0 11111111	SP 00000111	DPL 00000000	DPH 00000000			PCON 0XXX0000	87H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Special Function Registers

A map of the on-chip memory area called the Special Function Register (SFR) space is shown in Table 1.

Note that not all of the addresses are occupied, and unoccupied addresses may not be implemented on the chip. Read accesses to these addresses will in general return random data, and write accesses will have an indeterminate effect.

User software should not write 1s to these unlisted locations, since they may be used in future products to invoke new features. In that case, the reset or inactive values of the new bits will always be 0.

Timer 2 Registers Control and status bits are contained in registers T2CON (shown in Table 2) and T2MOD (shown in Table 4) for Timer 2. The register pair (RCAP2H, RCAP2L) are the Capture/Reload registers for Timer 2 in 16-bit capture mode or 16-bit auto-reload mode.

Interrupt Registers The individual interrupt enable bits are in the IE register. Two priorities can be set for each of the six interrupt sources in the IP register.r

Table 2. T2CON – Timer/Counter 2 Control Registe

T2CON Address = 0C8H — Reset Value = 0000 0000B

Bit Addressable

Bit	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$
	7	6	5	4	3	2	1	0

Symbol	Function
TF2	Timer 2 overflow flag set by a Timer 2 overflow and must be cleared by software. TF2 will not be set when either RCLK = 1 or TCLK = 1.
EXF2	Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX and EXEN2 = 1. When Timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software. EXF2 does not cause an interrupt in up/down counter mode (DCEN = 1).
RCLK	Receive clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its receive clock in serial port Modes 1 and 3. RCLK = 0 causes Timer 1 overflow to be used for the receive clock.
TCLK	Transmit clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its transmit clock in serial port Modes 1 and 3. TCLK = 0 causes Timer 1 overflows to be used for the transmit clock.
EXEN2	Timer 2 external enable. When set, allows a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. EXEN2 = 0 causes Timer 2 to ignore events at T2EX.
TR2	Start/Stop control for Timer 2. TR2 = 1 starts the timer.
C/$\overline{T2}$	Timer or counter select for Timer 2. C/$\overline{T2}$ = 0 for timer function. C/$\overline{T2}$ = 1 for external event counter (falling edge triggered).
CP/$\overline{RL2}$	Capture/Reload select. CP/$\overline{RL2}$ = 1 causes captures to occur on negative transitions at T2EX if EXEN2 = 1. CP/$\overline{RL2}$ = 0 causes automatic reloads to occur when Timer 2 overflows or negative transitions occur at T2EX when EXEN2 = 1. When either RCLK or TCLK = 1, this bit is ignored and the timer is forced to auto-reload on Timer 2 overflow.

Data Memory

The AT89C52 implements 256 bytes of on-chip RAM. The upper 128 bytes occupy a parallel address space to the Special Function Registers. That means the upper 128 bytes have the same addresses as the SFR space but are physically separate from SFR space.

When an instruction accesses an internal location above address 7FH, the address mode used in the instruction specifies whether the CPU accesses the upper 128 bytes of RAM or the SFR space. Instructions that use direct addressing access SFR space.

For example, the following direct addressing instruction accesses the SFR at location 0A0H (which is P2).

```
MOV 0A0H, #data
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Instructions that use indirect addressing access the upper 128 bytes of RAM. For example, the following indirect addressing instruction, where R0 contains 0A0H, accesses the data byte at address 0A0H, rather than P2 (whose address is 0A0H).

```
MOV @R0, #data
```

Note that stack operations are examples of indirect addressing, so the upper 128 bytes of data RAM are available as stack space.

Timer 0 and 1

Timer 0 and Timer 1 in the AT89C52 operate the same way as Timer 0 and Timer 1 in the AT89C51.

Timer 2

Timer 2 is a 16-bit Timer/Counter that can operate as either a timer or an event counter. The type of operation is selected by bit C/$\overline{T2}$ in the SFR T2CON (shown in Table 2). Timer 2 has three operating modes: capture, auto-reload (up or down counting), and baud rate generator. The modes are selected by bits in T2CON, as shown in Table 3.

Timer 2 consists of two 8-bit registers, TH2 and TL2. In the Timer function, the TL2 register is incremented every machine cycle. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

Table 3. Timer 2 Operating Modes

RCLK +TCLK	CP/$\overline{RL2}$	TR2	MODE
0	0	1	16-bit Auto-reload
0	1	1	16-bit Capture
1	X	1	Baud Rate Generator
X	X	0	(Off)

In the Counter function, the register is incremented in response to a 1-to-0 transition at its corresponding external input pin, T2. In this function, the external input is sampled during S5P2 of every machine cycle. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register during S3P1 of the cycle following the one in which the transition was detected. Since two machine cycles (24 oscillator periods) are required to recognize a 1-to-0 transition, the maximum count rate is 1/24 of the oscillator frequency. To ensure that a given level is sampled at least once before it changes, the level should be held for at least one full machine cycle.

Capture Mode

In the capture mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 is a 16-bit timer or counter which upon overflow sets bit TF2 in T2CON. This bit can then be used to generate an interrupt. If EXEN2 = 1, Timer 2 performs the same operation, but a 1-to-0 transition at external input T2EX also causes the current value in TH2 and TL2 to be captured into RCAP2H and RCAP2L, respectively. In addition, the transition at T2EX causes bit EXF2 in T2CON to be set. The EXF2 bit, like TF2, can generate an interrupt. The capture mode is illustrated in Figure 1.

Auto-reload (Up or Down Counter)

Timer 2 can be programmed to count up or down when configured in its 16-bit auto-reload mode. This feature is invoked by the DCEN (Down Counter Enable) bit located in the SFR T2MOD (see Table 4). Upon reset, the DCEN bit is set to 0 so that timer 2 will default to count up. When DCEN is set, Timer 2 can count up or down, depending on the value of the T2EX pin.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 1. Timer in Capture Mode

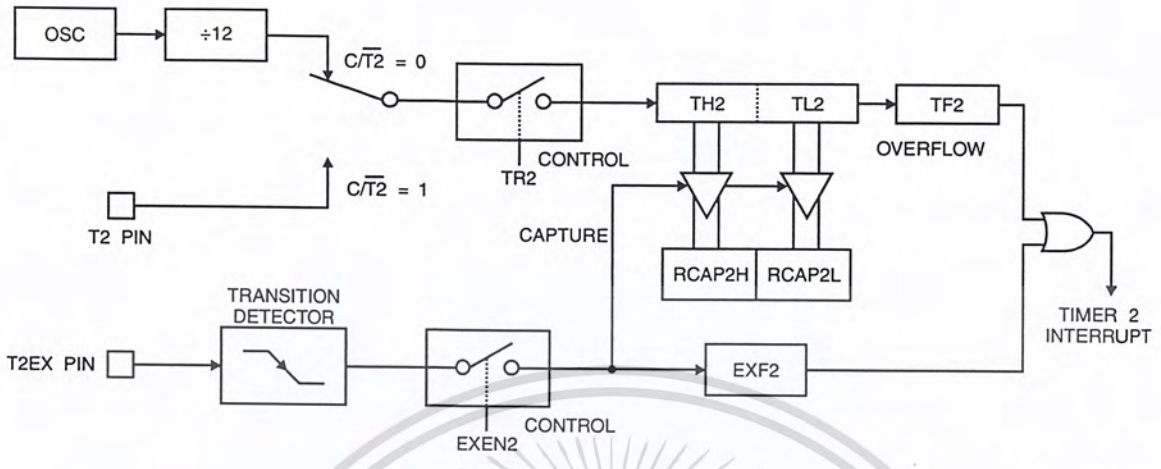


Figure 2 shows Timer 2 automatically counting up when DCEN = 0. In this mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 counts up to 0FFFFH and then sets the TF2 bit upon overflow. The overflow also causes the timer registers to be reloaded with the 16-bit value in RCAP2H and RCAP2L. The values in Timer in Capture ModeRCAP2H and RCAP2L are preset by software. If EXEN2 = 1, a 16-bit reload can be triggered either by an overflow or by a 1-to-0 transition at external input T2EX. This transition also sets the EXF2 bit. Both the TF2 and EXF2 bits can generate an interrupt if enabled.

Setting the DCEN bit enables Timer 2 to count up or down, as shown in Figure 3. In this mode, the T2EX pin controls the direction of the count. A logic 1 at T2EX makes Timer 2 count up. The timer will overflow at 0FFFFH and set the TF2 bit. This overflow also causes the 16-bit value in RCAP2H and RCAP2L to be reloaded into the timer registers, TH2 and TL2, respectively.

A logic 0 at T2EX makes Timer 2 count down. The timer underflows when TH2 and TL2 equal the values stored in RCAP2H and RCAP2L. The underflow sets the TF2 bit and causes 0FFFFH to be reloaded into the timer registers.

The EXF2 bit toggles whenever Timer 2 overflows or underflows and can be used as a 17th bit of resolution. In this operating mode, EXF2 does not flag an interrupt.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 2. Timer 2 Auto Reload Mode (DCEN = 0)

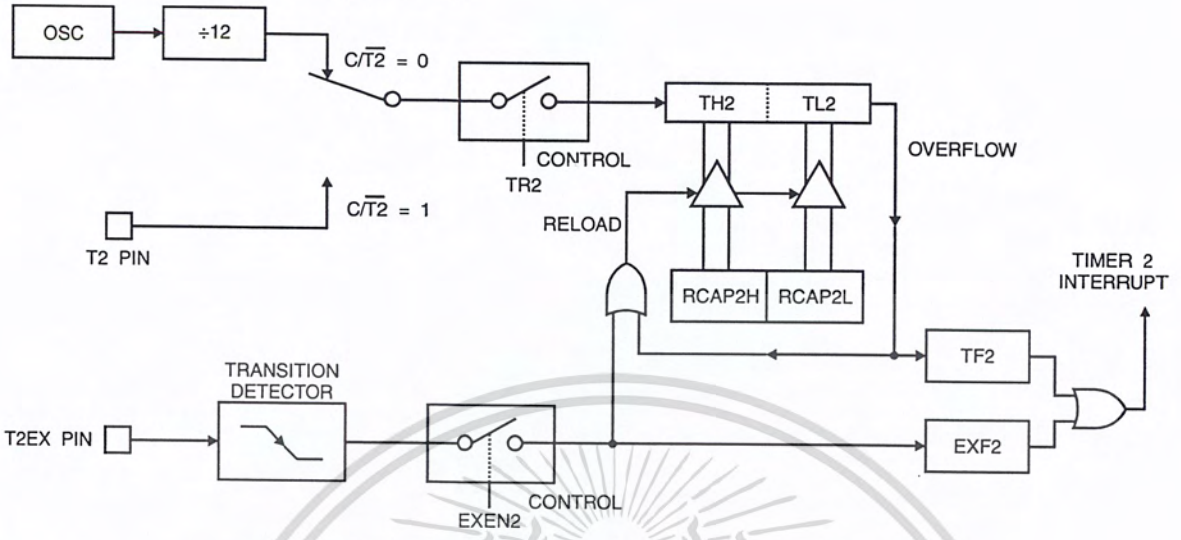


Table 4. T2MOD – Timer 2 Mode Control Register

T2MOD Address = 0C9H — Reset Value = XXXX XX00B

Not Bit Addressable

	–	–	–	–	–	–	T2OE	DCEN
Bit	7	6	5	4	3	2	1	0

Symbol	Function
–	Not implemented, reserved for future
T2OE	Timer 2 Output Enable bit.
DCEN	When set, this bit allows Timer 2 to be configured as an up/down counter.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 3. Timer 2 Auto Reload Mode (DCEN = 1)

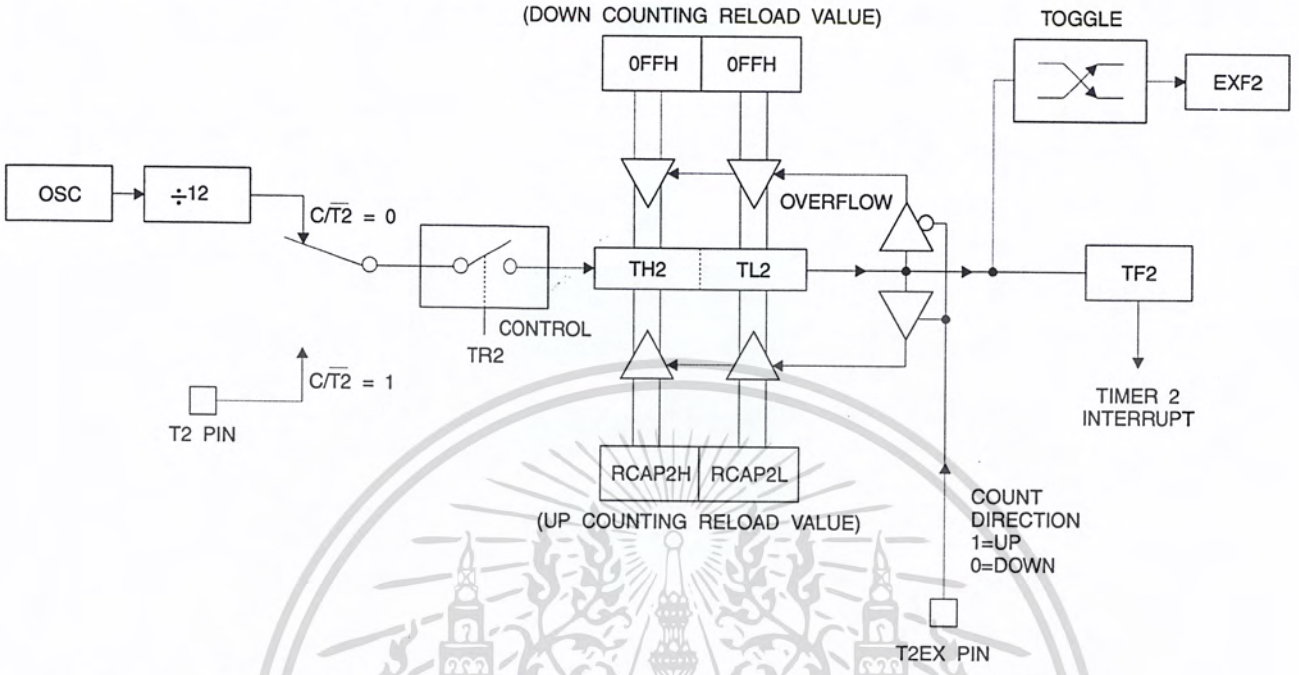
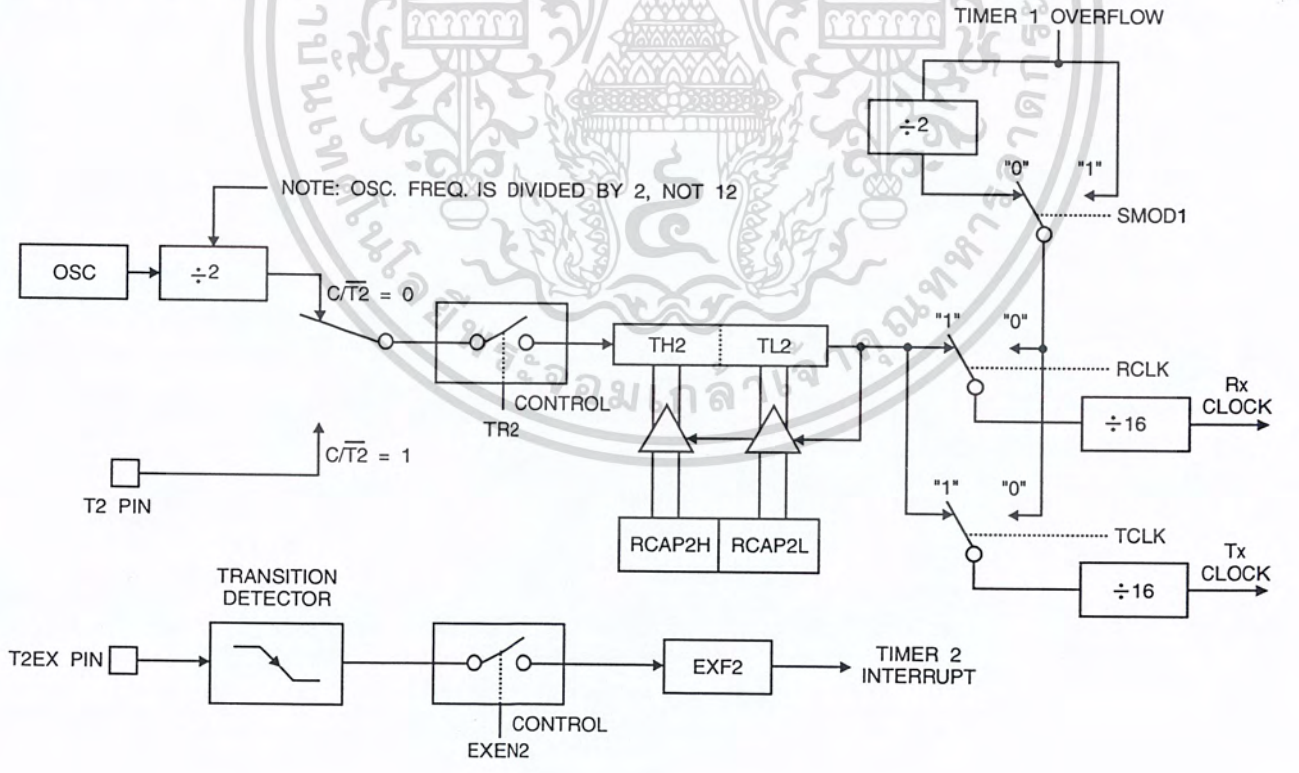


Figure 4. Timer 2 in Baud Rate Generator Mode



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Baud Rate Generator

Timer 2 is selected as the baud rate generator by setting TCLK and/or RCLK in T2CON (Table 2). Note that the baud rates for transmit and receive can be different if Timer 2 is used for the receiver or transmitter and Timer 1 is used for the other function. Setting RCLK and/or TCLK puts Timer 2 into its baud rate generator mode, as shown in Figure 4.

The baud rate generator mode is similar to the auto-reload mode, in that a rollover in TH2 causes the Timer 2 registers to be reloaded with the 16-bit value in registers RCAP2H and RCAP2L, which are preset by software.

The baud rates in Modes 1 and 3 are determined by Timer 2's overflow rate according to the following equation.

$$\text{Modes 1 and 3 Baud Rates} = \frac{\text{Timer 2 Overflow Rate}}{16}$$

The Timer can be configured for either timer or counter operation. In most applications, it is configured for timer operation ($C/\overline{T2} = 0$). The timer operation is different for Timer 2 when it is used as a baud rate generator. Normally, as a timer, it increments every machine cycle (at 1/12 the oscillator frequency). As a baud rate generator, however, it increments every state time (at 1/2 the oscillator frequency). The baud rate formula is given below.

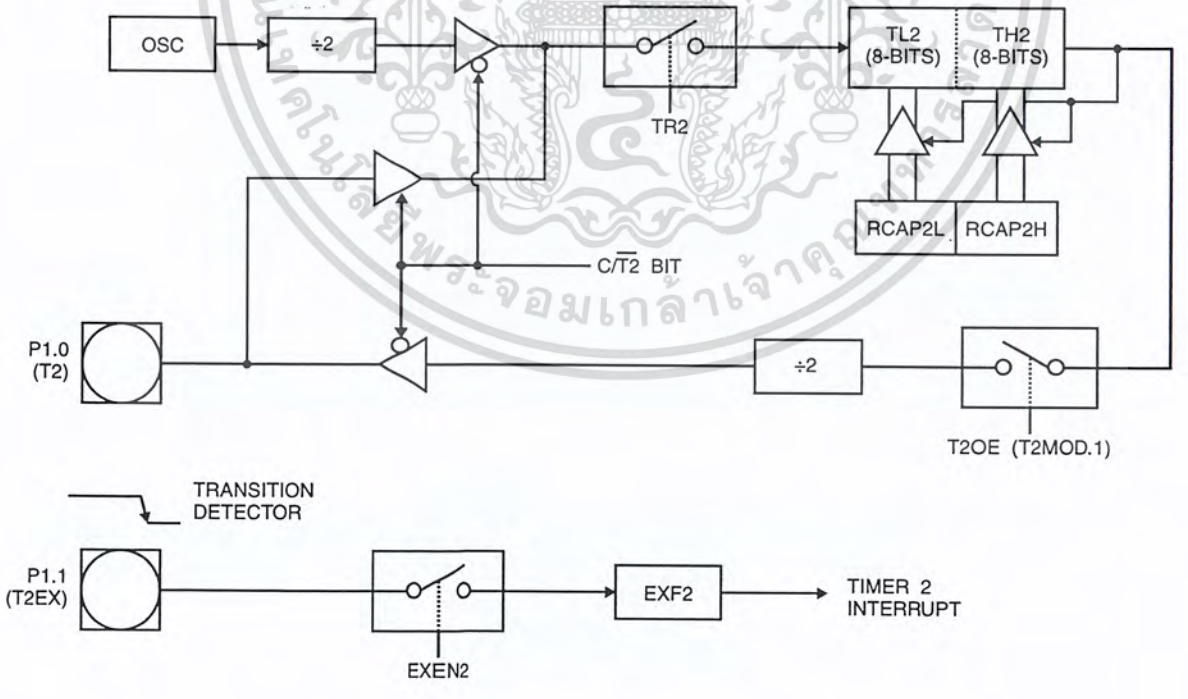
$$\frac{\text{Modes 1 and 3}}{\text{Baud Rate}} = \frac{\text{Oscillator Frequency}}{32 \times [65536 - (\text{RCAP2H,RCAP2L})]}$$

where (RCAP2H, RCAP2L) is the content of RCAP2H and RCAP2L taken as a 16-bit unsigned integer.

Timer 2 as a baud rate generator is shown in Figure 4. This figure is valid only if RCLK or TCLK = 1 in T2CON. Note that a rollover in TH2 does not set TF2 and will not generate an interrupt. Note too, that if EXEN2 is set, a 1-to-0 transition in T2EX will set EXF2 but will not cause a reload from (RCAP2H, RCAP2L) to (TH2, TL2). Thus when Timer 2 is in use as a baud rate generator, T2EX can be used as an extra external interrupt.

Note that when Timer 2 is running (TR2 = 1) as a timer in the baud rate generator mode, TH2 or TL2 should not be read from or written to. Under these conditions, the Timer is incremented every state time, and the results of a read or write may not be accurate. The RCAP2 registers may be read but should not be written to, because a write might overlap a reload and cause write and/or reload errors. The timer should be turned off (clear TR2) before accessing the Timer 2 or RCAP2 registers.

Figure 5. Timer 2 in Clock-out Mode



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Programmable Clock Out

A 50% duty cycle clock can be programmed to come out on P1.0, as shown in Figure 5. This pin, besides being a regular I/O pin, has two alternate functions. It can be programmed to input the external clock for Timer/Counter 2 or to output a 50% duty cycle clock ranging from 61 Hz to 4 MHz at a 16 MHz operating frequency.

To configure the Timer/Counter 2 as a clock generator, bit C/$\overline{T2}$ (T2CON.1) must be cleared and bit T2OE (T2MOD.1) must be set. Bit TR2 (T2CON.2) starts and stops the timer.

The clock-out frequency depends on the oscillator frequency and the reload value of Timer 2 capture registers (RCAP2H, RCAP2L), as shown in the following equation.

$$\text{Clock-Out Frequency} = \frac{\text{Oscillator Fequency}}{4 \times [65536 - (\text{RCAP2H,RCAP2L})]}$$

In the clock-out mode, Timer 2 roll-overs will not generate an interrupt. This behavior is similar to when Timer 2 is used as a baud-rate generator. It is possible to use Timer 2 as a baud-rate generator and a clock generator simultaneously. Note, however, that the baud-rate and clock-out frequencies cannot be determined independently from one another since they both use RCAP2H and RCAP2L.

UART

The UART in the AT89C52 operates the same way as the UART in the AT89C51.

Interrupts

The AT89C52 has a total of six interrupt vectors: two external interrupts ($\overline{INT0}$ and $\overline{INT1}$), three timer interrupts (Timers 0, 1, and 2), and the serial port interrupt. These interrupts are all shown in Figure 6.

Each of these interrupt sources can be individually enabled or disabled by setting or clearing a bit in Special Function Register IE. IE also contains a global disable bit, EA, which disables all interrupts at once.

Note that Table shows that bit position IE.6 is unimplemented. In the AT89C51, bit position IE.5 is also unimplemented. User software should not write 1s to these bit positions, since they may be used in future AT89 products.

Timer 2 interrupt is generated by the logical OR of bits TF2 and EXF2 in register T2CON. Neither of these flags is cleared by hardware when the service routine is vectored to. In fact, the service routine may have to determine whether it was TF2 or EXF2 that generated the interrupt, and that bit will have to be cleared in software.

The Timer 0 and Timer 1 flags, TF0 and TF1, are set at S5P2 of the cycle in which the timers overflow. The values are then polled by the circuitry in the next cycle. However, the Timer 2 flag, TF2, is set at S2P2 and is polled in the same cycle in which the timer overflows.

Table 5. Interrupt Enable (IE) Register

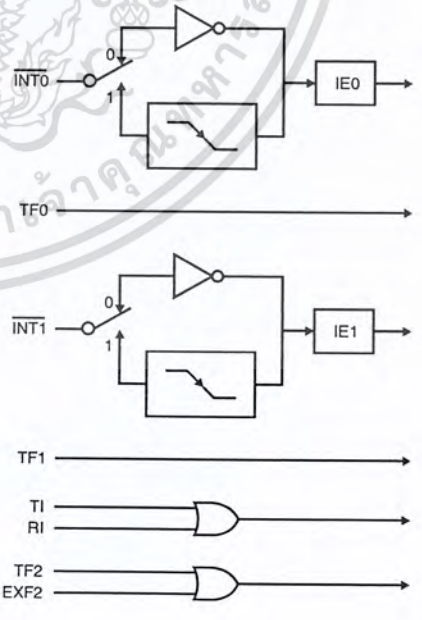
(MSB)							(LSB)
EA	–	ET2	ES	ET1	EX1	ET0	EX0

Enable Bit = 1 enables the interrupt.

Enable Bit = 0 disables the interrupt.

Symbol	Position	Function
EA	IE.7	Disables all interrupts. If EA = 0, no interrupt is acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
–	IE.6	Reserved.
ET2	IE.5	Timer 2 interrupt enable bit.
ES	IE.4	Serial Port interrupt enable bit.
ET1	IE.3	Timer 1 interrupt enable bit.
EX1	IE.2	External interrupt 1 enable bit.
ET0	IE.1	Timer 0 interrupt enable bit.
EX0	IE.0	External interrupt 0 enable bit.
User software should never write 1s to unimplemented bits, because they may be used in future AT89 products.		

Figure 6. Interrupt Sources



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Oscillator Characteristics

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier that can be configured for use as an on-chip oscillator, as shown in Figure 7. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven, as shown in Figure 8. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Idle Mode

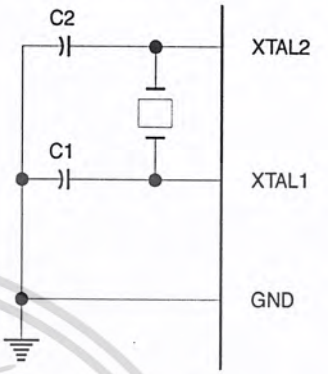
In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

Note that when idle mode is terminated by a hardware reset, the device normally resumes program execution from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when idle mode is terminated by a reset, the instruction following the one that invokes idle mode should not write to a port pin or to external memory.

Power-down Mode

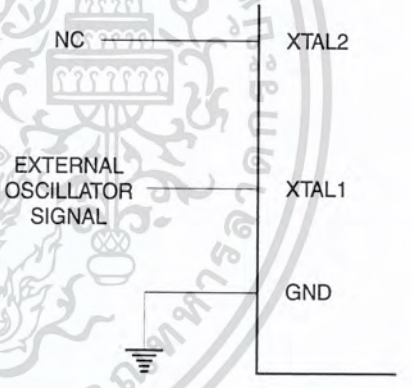
In the power-down mode, the oscillator is stopped, and the instruction that invokes power-down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power-down mode is terminated. The only exit from power-down is a hardware reset. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

Figure 7. Oscillator Connections



Note: C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

Figure 8. External Clock Drive Configuration



Status of External Pins During Idle and Powe-down Modes

Mode	Program Memory	ALE	$\overline{PSEN}$	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power-down	Internal	0	0	Data	Data	Data	Data
Power-down	External	0	0	Float	Data	Data	Data

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Program Memory Lock Bits

The AT89C52 has three lock bits that can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the following table.

Lock Bit Protection Modes

Program Lock Bits				
	LB1	LB2	LB3	Protection Type
1	U	U	U	No program lock features.
2	P	U	U	MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory, $\overline{EA}$ is sampled and latched on reset, and further programming of the Flash memory is disabled.
3	P	P	U	Same as mode 2, but verify is also disabled.
4	P	P	P	Same as mode 3, but external execution is also disabled.

When lock bit 1 is programmed, the logic level at the $\overline{EA}$ pin is sampled and latched during reset. If the device is powered up without a reset, the latch initializes to a random value and holds that value until reset is activated. The latched value of $\overline{EA}$ must agree with the current logic level at that pin in order for the device to function properly.

Programming the Flash

The AT89C52 is normally shipped with the on-chip Flash memory array in the erased state (that is, contents = FFH) and ready to be programmed. The programming interface accepts either a high-voltage (12-volt) or a low-voltage (V_{CC}) program enable signal. The Low-voltage programming mode provides a convenient way to program the AT89C52 inside the user's system, while the high-voltage programming mode is compatible with conventional third-party Flash or EPROM programmers.

The AT89C52 is shipped with either the high-voltage or low-voltage programming mode enabled. The respective top-side marking and device signature codes are listed in the following table.

	V_{PP} = 12V	V_{PP} = 5V
Top-side Mark	AT89C52 xxxx yyww	AT89C52 xxxx - 5 yyww
Signature	(030H) = 1EH (031H) = 52H (032H) = FFH	(030H) = 1EH (031H) = 52H (032H) = 05H

The AT89C52 code memory array is programmed byte-by-byte in either programming mode. *To program any non-blank byte in the on-chip Flash Memory, the entire memory must be erased using the Chip Erase Mode.*

Programming Algorithm Before programming the AT89C52, the address, data and control signals should be set up according to the Flash programming mode table and Figure 9 and Figure 10. To program the AT89C52, take the following steps.

1. Input the desired memory location on the address lines.
2. Input the appropriate data byte on the data lines.
3. Activate the correct combination of control signals.
4. Raise $\overline{EA}/V_{PP}$ to 12V for the high-voltage programming mode.
5. Pulse ALE/$\overline{PROG}$ once to program a byte in the Flash array or the lock bits. The byte-write cycle is self-timed and typically takes no more than 1.5 ms. Repeat steps 1 through 5, changing the address and data for the entire array or until the end of the object file is reached.

Data Polling The AT89C52 features $\overline{Data}$ Polling to indicate the end of a write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written data on PO.7. Once the write cycle has been completed, true data is valid on all outputs, and the next cycle may begin. $\overline{Data}$ Polling may begin any time after a write cycle has been initiated.

Ready/$\overline{Busy}$ The progress of byte programming can also be monitored by the RDY/$\overline{BSY}$ output signal. P3.4 is pulled low after ALE goes high during programming to indicate $\overline{BUSY}$. P3.4 is pulled high again when programming is done to indicate READY.

Program Verify If lock bits LB1 and LB2 have not been programmed, the programmed code data can be read back via the address and data lines for verification. The lock bits cannot be verified directly. Verification of the lock bits is achieved by observing that their features are enabled.

Chip Erase The entire Flash array is erased electrically by using the proper combination of control signals and by holding ALE/$\overline{PROG}$ low for 10 ms. The code array is written with all 1s. The chip erase operation must be executed before the code memory can be reprogrammed.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Reading the Signature Bytes The signature bytes are read by the same procedure as a normal verification of locations 030H, 031H, and 032H, except that P3.6 and P3.7 must be pulled to a logic low. The values returned are as follows.

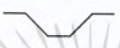
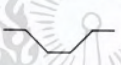
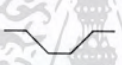
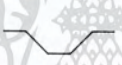
(030H) = 1EH indicates manufactured by Atmel
(031H) = 52H indicates 89C52
(032H) = FFH indicates 12V programming
(032H) = 05H indicates 5V programming

Programming Interface

Every code byte in the Flash array can be written, and the entire array can be erased, by using the appropriate combination of control signals. The write operation cycle is self-timed and once initiated, will automatically time itself to completion.

All major programming vendors offer worldwide support for the Atmel microcontroller series. Please contact your local programming vendor for the appropriate software revision.

Flash Programming Modes

Mode		RST	$\overline{PSEN}$	ALE/$\overline{PROG}$	$\overline{EA}/V_{PP}$	P2.6	P2.7	P3.6	P3.7
Write Code Data		H	L		H/12V	L	H	H	H
Read Code Data		H	L	H	H	L	L	H	H
Write Lock	Bit - 1	H	L		H/12V	H	H	H	H
	Bit - 2	H	L		H/12V	H	H	L	L
	Bit - 3	H	L		H/12V	H	L	H	L
Chip Erase		H	L	(1)	H/12V	H	L	L	L
Read Signature Byte		H	L	H	H	L	L	L	L

Note: 1. Chip Erase requires a 10 ms $\overline{PROG}$ pulse.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 9. Programming the Flash Memory

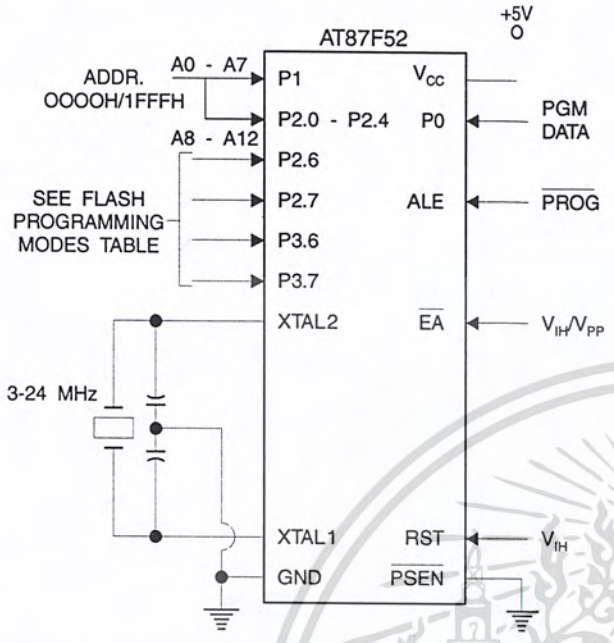
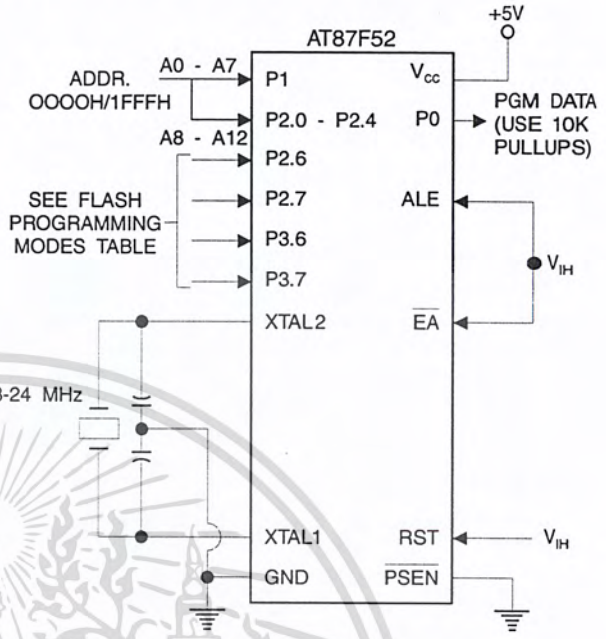


Figure 10. Verifying the Flash Memory



Flash Programming and Verification Characteristics

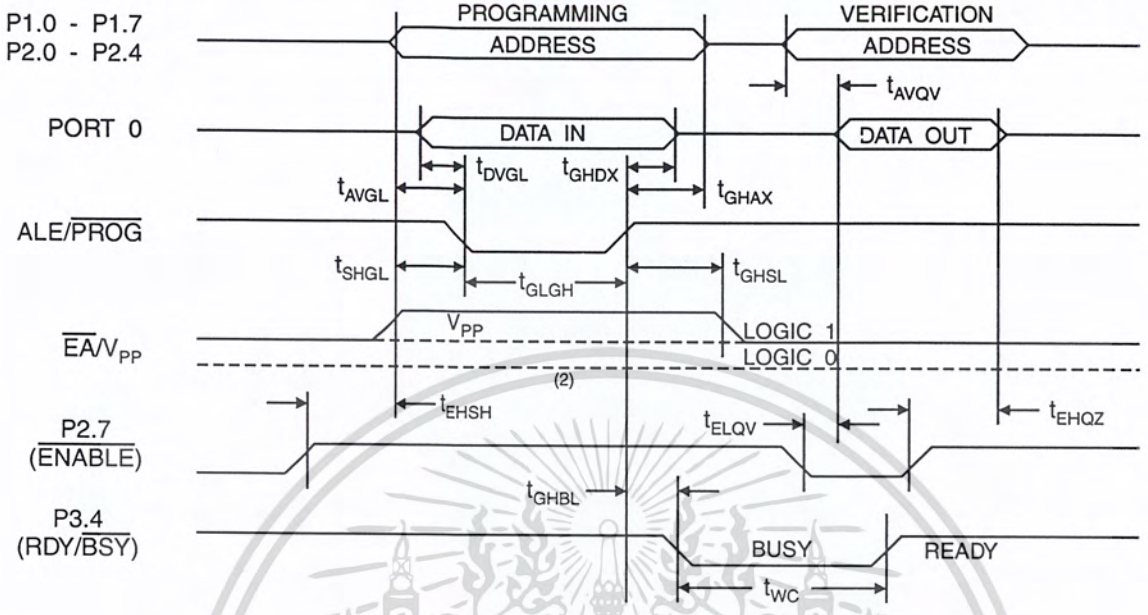
T_A = 0°C to 70°C, V_{CC} = 5.0 ± 10%

Symbol	Parameter	Min	Max	Units
V_{PP}(1)	Programming Enable Voltage	11.5	12.5	V
I_{PP}(1)	Programming Enable Current		1.0	mA
$1/t_{CLCL}$	Oscillator Frequency	3	24	MHz
t_{AVGL}	Address Setup to PROG Low	$48t_{CLCL}$		
t_{GHAX}	Address Hold after PROG	$48t_{CLCL}$		
t_{DVGL}	Data Setup to PROG Low	$48t_{CLCL}$		
t_{GHDX}	Data Hold After PROG	$48t_{CLCL}$		
t_{EHSH}	P2.7 (ENABLE) High to V_{PP}	$48t_{CLCL}$		
t_{SHGL}	V_{PP} Setup to PROG Low	10		µs
t_{GHSL}(1)	V_{PP} Hold after PROG	10		µs
t_{GLGH}	PROG Width	1	110	µs
t_{AVQV}	Address to Data Valid		$48t_{CLCL}$	
t_{ELQV}	ENABLE Low to Data Valid		$48t_{CLCL}$	
t_{EHQZ}	Data Float after ENABLE	0	$48t_{CLCL}$	
t_{GHBL}	PROG High to BUSY Low		1.0	µs
t_{WC}	Byte Write Cycle Time		2.0	ms

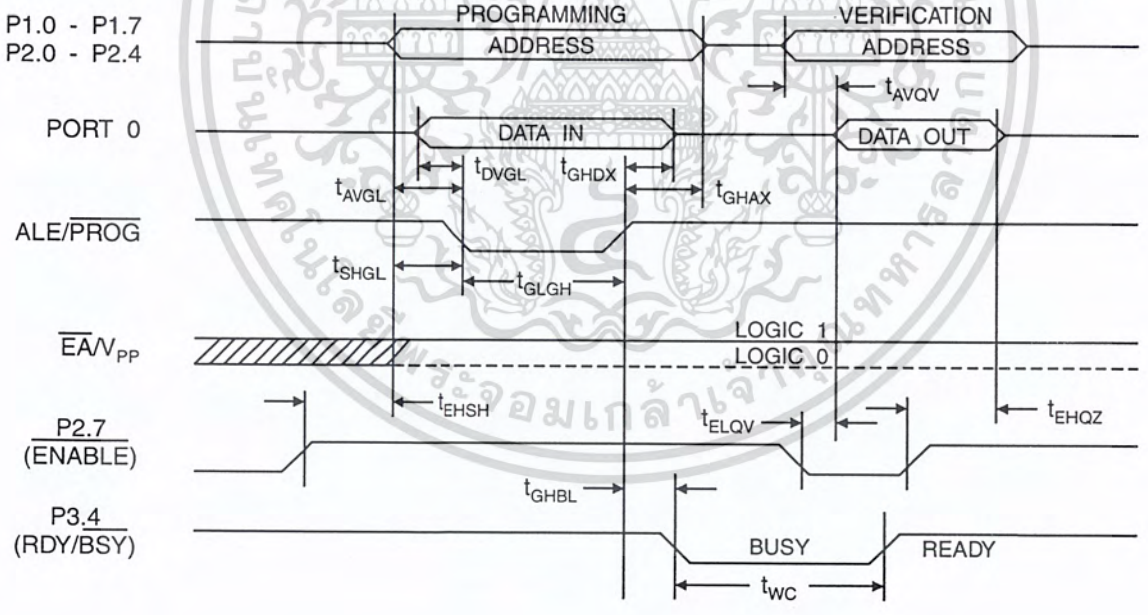
Note: 1. Only used in 12-volt programming mode.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flash Programming and Verification Waveforms - High-voltage Mode (V_{PP}=12V)



Flash Programming and Verification Waveforms - Low-voltage Mode (V_{PP}=5V)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on Any Pin with Respect to Ground	-1.0V to +7.0V
Maximum Operating Voltage	6.6V
DC Output Current	15.0 mA

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC Characteristics

The values shown in this table are valid for T_A = -40°C to 85°C and V_{CC} = 5.0V ± 20%, unless otherwise noted.

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low-voltage	(Except $\overline{EA}$)	-0.5	0.2 V_{CC}-0.1	V
V_{IL1}	Input Low-voltage ($\overline{EA}$)		-0.5	0.2 V_{CC}-0.3	V
V_{IH}	Input High-voltage	(Except XTAL1, RST)	0.2 V_{CC}+0.9	V_{CC}+0.5	V
V_{IH1}	Input High-voltage	(XTAL1, RST)	0.7 V_{CC}	V_{CC}+0.5	V
V_{OL}	Output Low-voltage[1] (Ports 1,2,3)	I_{OL} = 1.6 mA		0.45	V
V_{OL1}	Output Low-voltage[1] (Port 0, ALE, $\overline{PSEN}$)	I_{OL} = 3.2 mA		0.45	V
V_{OH}	Output High-voltage (Ports 1,2,3, ALE, $\overline{PSEN}$)	I_{OH} = -60 µA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -25 µA	0.75 V_{CC}		V
		I_{OH} = -10 µA	0.9 V_{CC}		V
V_{OH1}	Output High-voltage (Port 0 in External Bus Mode)	I_{OH} = -800 µA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -300 µA	0.75 V_{CC}		V
		I_{OH} = -80 µA	0.9 V_{CC}		V
I_{IL}	Logical 0 Input Current (Ports 1,2,3)	V_{IN} = 0.45V		-50	µA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1,2,3)	V_{IN} = 2V, V_{CC} = 5V ± 10%		-650	µA
I_{LI}	Input Leakage Current (Port 0, $\overline{EA}$)	0.45 < V_{IN} < V_{CC}		±10	µA
RRST	Reset Pulldown Resistor		50	300	KΩ
C_{IO}	Pin Capacitance	Test Freq. = 1 MHz, T_A = 25°C		10	pF
I_{CC}	Power Supply Current	Active Mode, 12 MHz		25	mA
		Idle Mode, 12 MHz		6.5	mA
	Power-down Mode[1]	V_{CC} = 6V		100	µA
		V_{CC} = 3V		40	µA

Notes: 1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
Maximum I_{OL} per port pin: 10 mA
Maximum I_{OL} per 8-bit port:
Port 0: 26 mA Ports 1, 2, 3: 15 mA
Maximum total I_{OL} for all output pins: 71 mA
If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
2. Minimum V_{CC} for Power-down is 2V.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC Characteristics

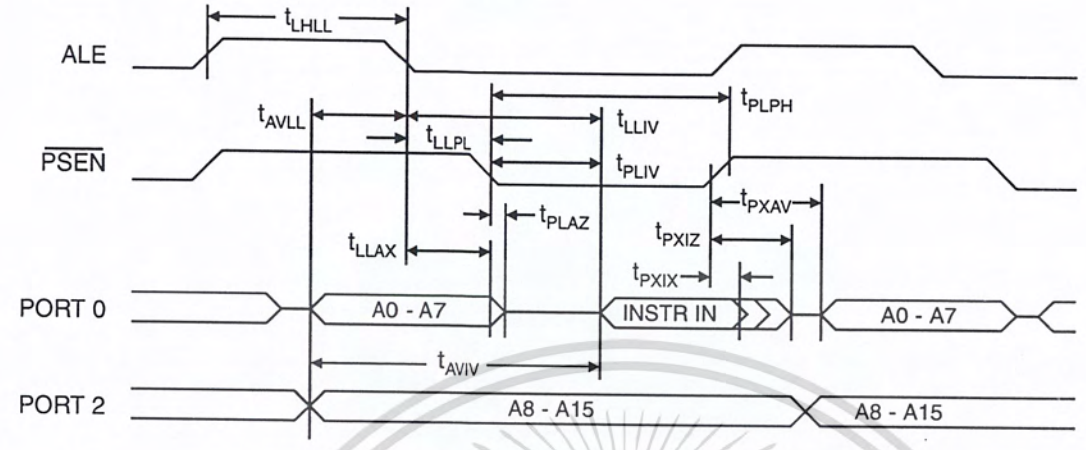
Under operating conditions, load capacitance for Port 0, ALE/$\overline{PROG}$, and $\overline{PSEN}$ = 100 pF; load capacitance for all other outputs = 80 pF.

External Program and Data Memory Characteristics

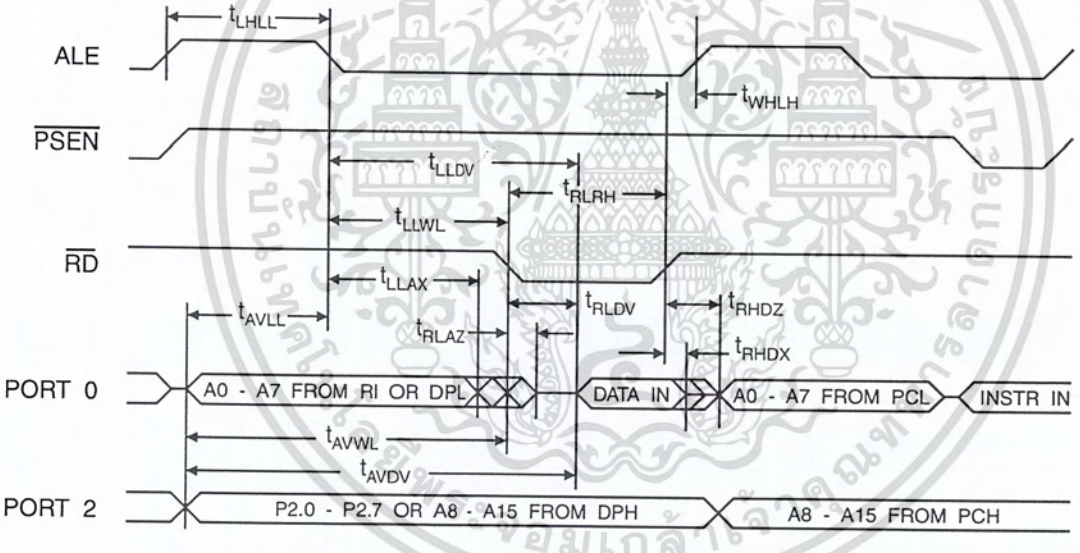
Symbol	Parameter	12 MHz Oscillator		Variable Oscillator		Units
		Min	Max	Min	Max	
$1/t_{CLCL}$	Oscillator Frequency			0	24	MHz
t_{LHLL}	ALE Pulse Width	127		$2t_{CLCL}$-40		ns
t_{AVLL}	Address Valid to ALE Low	43		t_{CLCL}-13		ns
t_{LLAX}	Address Hold After ALE Low	48		t_{CLCL}-20		ns
t_{LLIV}	ALE Low to Valid Instruction In		233		$4t_{CLCL}$-65	ns
t_{LLPL}	ALE Low to $\overline{PSEN}$ Low	43		t_{CLCL}-13		ns
t_{PLPH}	$\overline{PSEN}$ Pulse Width	205		$3t_{CLCL}$-20		ns
t_{PLIV}	$\overline{PSEN}$ Low to Valid Instruction In		145		$3t_{CLCL}$-45	ns
t_{PXIX}	Input Instruction Hold after $\overline{PSEN}$	0		0		ns
t_{PXIZ}	Input Instruction Float after $\overline{PSEN}$		59		t_{CLCL}-10	ns
t_{PXAV}	$\overline{PSEN}$ to Address Valid	75		t_{CLCL}-8		ns
t_{AVIV}	Address to Valid Instruction In		312		$5t_{CLCL}$-55	ns
t_{PLAZ}	$\overline{PSEN}$ Low to Address Float		10		10	ns
t_{RLRH}	$\overline{RD}$ Pulse Width	400		$6t_{CLCL}$-100		ns
t_{WLWH}	$\overline{WR}$ Pulse Width	400		$6t_{CLCL}$-100		ns
t_{RLDV}	$\overline{RD}$ Low to Valid Data In		252		$5t_{CLCL}$-90	ns
t_{RHDX}	Data Hold After $\overline{RD}$	0		0		ns
t_{RHDZ}	Data Float After $\overline{RD}$		97		$2t_{CLCL}$-28	ns
t_{LLDV}	ALE Low to Valid Data In		517		$8t_{CLCL}$-150	ns
t_{AVDV}	Address to Valid Data In		585		$9t_{CLCL}$-165	ns
t_{LLWL}	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	200	300	$3t_{CLCL}$-50	$3t_{CLCL}$+50	ns
t_{AVWL}	Address to $\overline{RD}$ or $\overline{WR}$ Low	203		$4t_{CLCL}$-75		ns
t_{QVWX}	Data Valid to $\overline{WR}$ Transition	23		t_{CLCL}-20		ns
t_{QVWH}	Data Valid to $\overline{WR}$ High	433		$7t_{CLCL}$-120		ns
t_{WHQX}	Data Hold After $\overline{WR}$	33		t_{CLCL}-20		ns
t_{RLAZ}	$\overline{RD}$ Low to Address Float		0		0	ns
t_{WHLH}	$\overline{RD}$ or $\overline{WR}$ High to ALE High	43	123	t_{CLCL}-20	t_{CLCL}+25	ns

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

External Program Memory Read Cycle

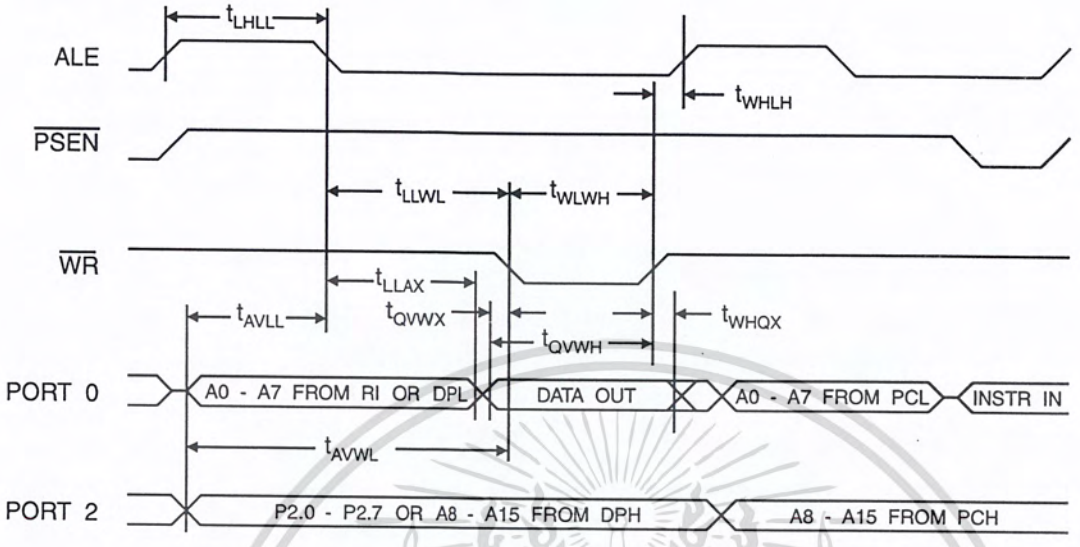


External Data Memory Read Cycle

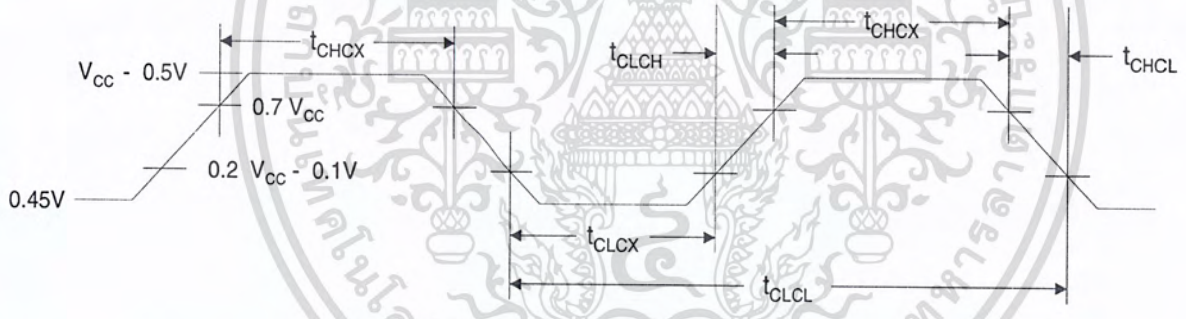


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

External Data Memory Write Cycle



External Clock Drive Waveforms



External Clock Drive

Symbol	Parameter	Min	Max	Units
$1/t_{CLCL}$	Oscillator Frequency	0	24	MHz
t_{CLCL}	Clock Period	41.6		ns
t_{CHCX}	High Time	15		ns
t_{CLCX}	Low Time	15		ns
t_{CLCH}	Rise Time		20	ns
t_{CHCL}	Fall Time		20	ns

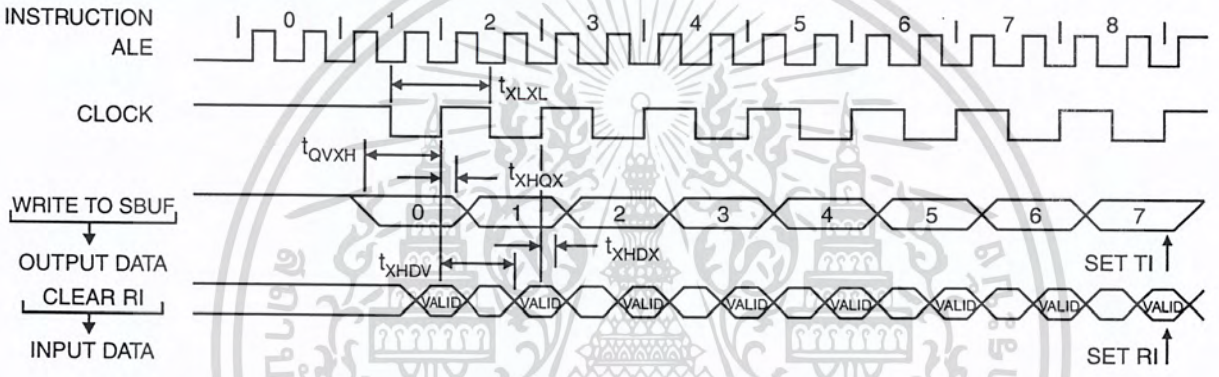
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Serial Port Timing: Shift Register Mode Test Conditions

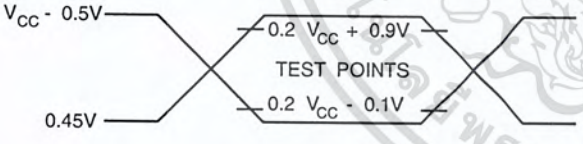
The values in this table are valid for V_{CC} = 5.0V ± 20% and Load Capacitance = 80 pF.

Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
t_{XLXL}	Serial Port Clock Cycle Time	1.0		$12t_{CLCL}$		µs
t_{QVXH}	Output Data Setup to Clock Rising Edge	700		$10t_{CLCL}$-133		ns
t_{XHQX}	Output Data Hold After Clock Rising Edge	50		$2t_{CLCL}$-117		ns
t_{XHDX}	Input Data Hold After Clock Rising Edge	0		0		ns
t_{XHDV}	Clock Rising Edge to Input Data Valid		700		$10t_{CLCL}$-133	ns

Shift Register Mode Timing Waveforms

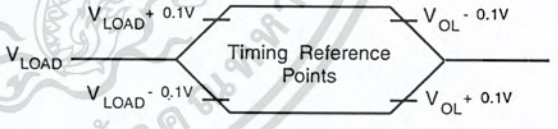


AC Testing Input/Output Waveforms(1)



Note: 1. AC Inputs during testing are driven at V_{CC} - 0.5V for a logic 1 and 0.45V for a logic 0. Timing measurements are made at V_{IH} min. for a logic 1 and V_{IL} max. for a logic 0.

Float Waveforms(1)



Note: 1. For timing purposes, a port pin is no longer floating when a 100 mV change from load voltage occurs. A port pin begins to float when a 100 mV change from the loaded V_{OH}/V_{OL} level occurs.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ordering Information

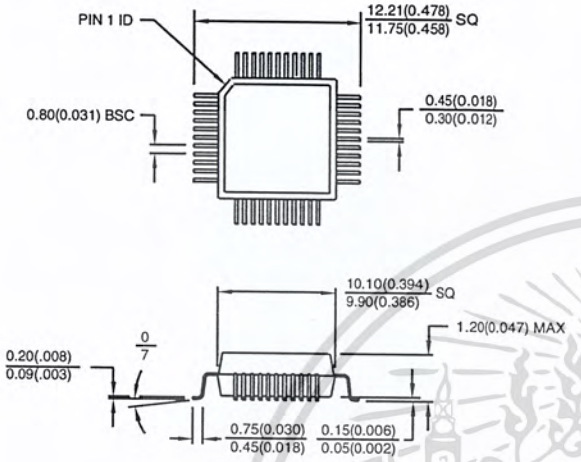
Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
12	5V ± 20%	AT89C52-12AC	44A	Commercial (0°C to 70°C)
		AT89C52-12JC	44J	
		AT89C52-12PC	40P6	
		AT89C52-12QC	44Q	
		AT89C52-12AI	44A	Industrial (-40°C to 85°C)
		AT89C52-12JI	44J	
		AT89C52-12PI	40P6	
		AT89C52-12QI	44Q	
16	5V ± 20%	AT89C52-16AC	44A	Commercial (0°C to 70°C)
		AT89C52-16JC	44J	
		AT89C52-16PC	40P6	
		AT89C52-16QC	44Q	
		AT89C52-16AI	44A	Industrial (-40°C to 85°C)
		AT89C52-16JI	44J	
		AT89C52-16PI	40P6	
		AT89C52-16QI	44Q	
20	5V ± 20%	AT89C52-20AC	44A	Commercial (0°C to 70°C)
		AT89C52-20JC	44J	
		AT89C52-20PC	40P6	
		AT89C52-20QC	44Q	
		AT89C52-20AI	44A	Industrial (-40°C to 85°C)
		AT89C52-20JI	44J	
		AT89C52-20PI	40P6	
		AT89C52-20QI	44Q	
24	5V ± 20%	AT89C52-24AC	44A	Commercial (0°C to 70°C)
		AT89C52-24JC	44J	
		AT89C52-24PC	40P6	
		AT89C52-24QC	44Q	
		AT89C52-24AI	44A	Industrial (-40°C to 85°C)
		AT89C52-24JI	44J	
		AT89C52-24PI	40P6	
		AT89C52-24QI	44Q	

Package Type	
44A	44-lead, Thin Plastic Gull Wing Quad Flatpack (TQFP)
44J	44-lead, Plastic J-leaded Chip Carrier (PLCC)
40P6	40-lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
44Q	44-lead, Plastic Gull Wing Quad Flatpack (PQFP)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

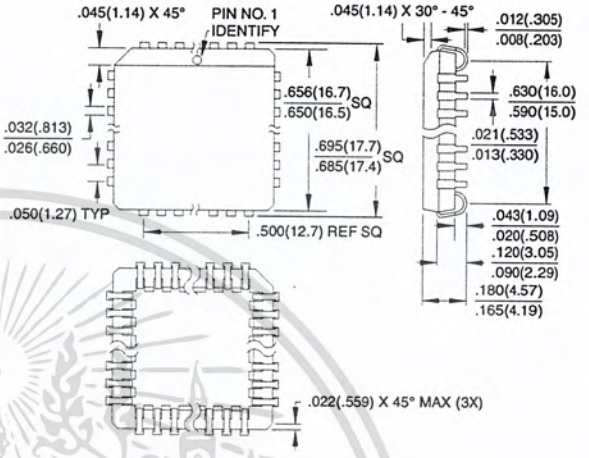
Packaging Information

44A, 44-lead, Thin (1.0 mm) Plastic Gull Wing Quad Flatpack (TQFP)
Dimensions in Millimeters and (Inches)*
JEDEC STANDARD MS-026 ACB

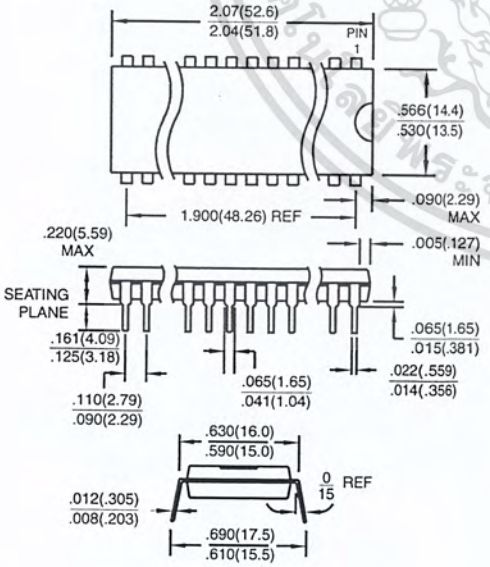


Controlling dimension: millimeters

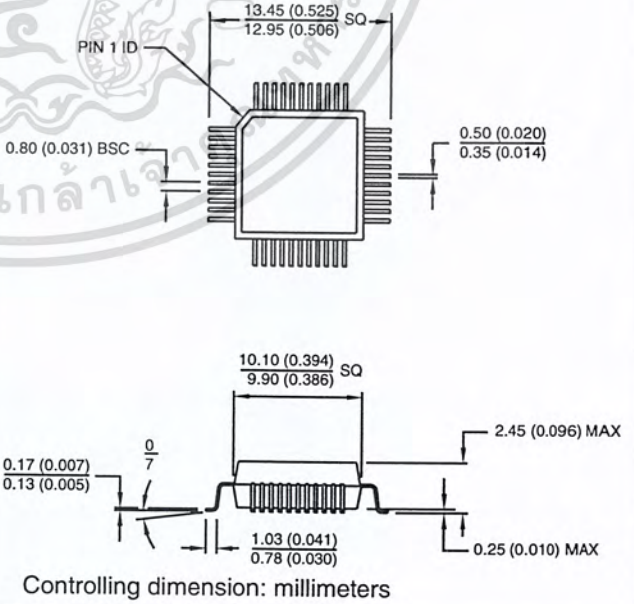
44J, 44-lead, Plastic J-leaded Chip Carrier (PLCC)
Dimensions in Inches and (Millimeters)
JEDEC STANDARD MS-018 AC



40P6, 40-lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
Dimensions in Inches and (Millimeters)



44Q, 44-lead, Plastic Quad Flat Package (PQFP)
Dimensions in Millimeters and (Inches)*
JEDEC STANDARD MS-022 AB



Controlling dimension: millimeters

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Atmel Headquarters

Corporate Headquarters
2325 Orchard Parkway
San Jose, CA 95131
TEL (408) 441-0311
FAX (408) 487-2600

Europe
Atmel U.K., Ltd.
Coliseum Business Centre
Riverside Way
Camberley, Surrey GU15 3YL
England
TEL (44) 1276-686-677
FAX (44) 1276-686-697

Asia
Atmel Asia, Ltd.
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimhatsui
East Kowloon
Hong Kong
TEL (852) 2721-9778
FAX (852) 2722-1369

Japan
Atmel Japan K.K.
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
TEL (81) 3-3523-3551
FAX (81) 3-3523-7581

Atmel Operations

Atmel Colorado Springs
1150 E. Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL (719) 576-3300
FAX (719) 540-1759

Atmel Rousset
Zone Industrielle
13106 Rousset Cedex
France
TEL (33) 4-4253-6000
FAX (33) 4-4253-6001

Fax-on-Demand
North America:
1-(800) 292-8635
International:
1-(408) 441-0732

e-mail
literature@atmel.com

Web Site
http://www.atmel.com

BBS
1-(408) 436-4309

© Atmel Corporation 1999.
Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical components in life support devices or systems.

Marks bearing ® and/or ™ are registered trademarks and trademarks of Atmel Corporation.

Terms and product names in this document may be trademarks of others.

Printed on recycled paper.

0313H–02/00/xM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

ไกรวุฒิ โรจน์ประเสริฐสุด “ไมโครโปรเซสเซอร์ 2” , บริษัทซีเอ็ดยูเคชั่นจำกัด , 2539

ดร.ณสรรค์ ผลโภค “ Concept in physics” , สำนักพิมพ์แม็คจำกัด , 2536

ชิตชัย สุทธาศวิน “อิเล็กทรอนิกส์ทั่วไป” , 23บุ๊คเซนเตอร์ , 2531

www.cim.mcgill.ca

www.panmanee.com

http://www.uwe.ac.uk/clawer/newsletters/issue2/Roma.html/

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้