สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

หุ่นยนต์เต่า

TURTLE ROBOT

โดย

นายเขมวิทย์ ปัญญากาบ รหัสประจำตัว 42015546

นายสถาพร ศรีมณี รหัสประจำตัว 42015576

นางสาวสิริพร สังข์เจริญ รหัสประจำตัว 42015579

อาจารย์ที่ปรึกษา

อาจารย์สุธีรา พัทธ์ธีรานุรักษ์

ผศ.ดร.ปีติเขต สู้รักษา

เลขหมู่.....
เลขทะเบียน 42255
วัน, เดือน, ปี 16 พ.ค. 2545

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรอุตสาหกรรมศาสตรบัณฑิต

สาขาวิชาเทคโนโลยีอิเล็กทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อโครงงานภาษาไทย	หุ่นยนต์เต่า
ชื่อโครงงานภาษาอังกฤษ	TURTLE ROBOT
ผู้จัดทำ	นายเขมวิทย์ ปัญญากาบ
	นายสถาพร ศรีมณี
	นางสาวสิริพร สังข์เจริญ
อาจารย์ที่ปรึกษา	อาจารย์สุธีรา พันธุ์ธีรานุรักษ์
	ผศ.ดร.ปิติเขต สู้รักษา
ภาควิชา	เทคนิคอุตสาหกรรม
ปีการศึกษา	2544

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้รับปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรอุตสาหกรรมศาสตรบัณฑิต

ลงชื่อ.....อาจารย์ที่ปรึกษา
(อาจารย์สุธีรา พันธุ์ธีรานุรักษ์)

ลงชื่อ.....อาจารย์ที่ปรึกษา
(ผศ.ดร.ปิติเขต สู้รักษา)

คณะกรรมการสอบปริญญานิพนธ์

.....กรรมการ
( )

.....กรรมการ
( )

.....กรรมการ
( )

.....กรรมการ
( )

.....กรรมการ
( )

ลิขสิทธิ์ของคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หุ่นยนต์เต่า

จัดทำโดย	นายเขมวิทย์ ปัญญากาบ	42015546
	นายสถาพร ศรีมณี	42015576
	นางสาวสิริพร สังข์เจริญ	42015579
อาจารย์ที่ปรึกษา	อาจารย์สุธีรา พันธุ์ธีรานุรักษ์	
	ผศ.ดร.ปิติเขต สู้รักษา	
ปีการศึกษา	2544	

บทคัดย่อ

ปริญญานิพนธ์นี้เสนอการสร้างและควบคุมหุ่นยนต์เต่า โดยออกแบบให้ง่ายต่อการสร้างและมีราคาถูก ใช้เซอร์โวมอเตอร์เป็นตัวขับเคลื่อน และไมโครคอนโทรลเลอร์มาควบคุมการทำงานของเซอร์โวมอเตอร์ ซึ่งจะทำหน้าที่ควบคุมการก้าวเดินของตัวหุ่นยนต์ให้สามารถเคลื่อนที่สามารถทำงานในโหมดควบคุมโดยมนุษย์ โดยใช้รีโมทคอนโทรลในการควบคุมหุ่นยนต์สามารถที่จะเคลื่อนที่บนบกหรือเคลื่อนที่ในน้ำได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TURTLE ROBOT

By	Mr. KHEMWIT PUNYAKAB	42015546
	Mr. SATHAPORN SRIMANEE	42015576
	Miss. SIRIPORN SANGCHAREON	42015579
Adviser	Miss. SUTEERA PANTEERANURUK	
	Asst. Prof. Dr. PITIKHATE SOORAKSA	
Academic year	2001	

Abstrasct

A turtle robot is designed and built in this study with low-cost and easy-to-build criteria. Servomotors are employed as the driver; whereas, microcontrollers are used for controlling the movement remotely and manually. With a good seal, the robot can navigate on the ground and also in the water.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

จากความสำเร็จหลายๆด้านในการสร้างหุ่นยนต์เต่าคณะผู้จัดทำขอขอบคุณ ท่านอาจารย์ ผศ .ดร.ปิติเขต สู้รักษา อาจารย์สุธีรา พันธุ์ธีรานุรักษ์ เพื่อนๆห้อง 3L และพี่ร้านลาดกระบังเรซิ่น ที่ให้คำปรึกษา ช่วยเหลือ แนะนำในทุกๆด้าน และที่สำคัญขอขอบพระคุณพ่อแม่ที่เป็นกำลังใจที่ยิ่งใหญ่ให้อยู่เสมอ สุดท้ายขอขอบพระคุณ ภาคเทคนิคอุตสาหกรรม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้ให้การสนับสนุนจนกระทั่งโครงการนี้ได้สำเร็จลุล่วงไปด้วยดี

ลงชื่อ.....

(นายเขมวิทย์ ปัญญากาบ)

ลงชื่อ.....

(นายสถาพร ศรีมณี)

ลงชื่อ.....

(นางสาวสิริพร สังข์เจริญ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	VI
บทที่ 1 บทนำ	
1.1 ความเป็นมา	1
1.2 ชื่อโครงงาน	1
1.3 จุดประสงค์	1
1.4 ขอบเขตและผลที่คาดว่าจะได้รับ	2
1.5 เนื้อหาโดยสังเขป	2
บทที่ 2 ทฤษฎีและหลักการ	
2.1 ไมโครคอนโทรลเลอร์ตระกูล MCS-51	3
2.2 เซอร์โวมอเตอร์	4
บทที่ 3 โครงสร้างและการออกแบบของหุ่นยนต์เต่า	
3.1 การออกแบบขา	6
3.2 การออกแบบโครงสร้างของตัวหุ่นยนต์	16
3.3 ซิลิโคนหุ้มขาหุ่นยนต์เต่า	19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

เรื่อง	หน้า
บทที่ 4 ระบบการควบคุม	
4.1 หลักการควบคุม	21
4.2 แนวทางในการควบคุม	22
4.3 การทดลองการเคลื่อนที่ของหุ่นยนต์เต่า	25
4.4 รูปสัญญาณพัลส์ควบคุมเซอร์โวมอเตอร์	30
บทที่ 5 สรุปโครงงานและข้อเสนอแนะ	
5.1 สรุปโครงงาน	34
5.2 ปัญหาในการทำงาน	34
5.3 ข้อเสนอแนะ	35
ภาคผนวก ก รหัสโปรแกรมการเคลื่อนที่ของหุ่นยนต์เต่า	
ภาคผนวก ข วงจรควบคุมไมโครคอนโทรลเลอร์	
ภาคผนวก ค รูปภาพตัวหุ่นยนต์	
ภาคผนวก ง รายละเอียดของ AT89C52	
บรรณานุกรม	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

เรื่อง	หน้า
รูปที่ 2.1 แสดงคาบเวลาของพัลส์ที่เป็นตัวกำหนดองศาเซอร์โวมอเตอร์	5
รูปที่ 3.1 แสดงภาพแท่นที่ใช้ยึดเซอร์โวมอเตอร์สองตัวแรกขาด้านขวา	7
รูปที่ 3.2 แสดงภาพแท่นที่ใช้ยึดเซอร์โวมอเตอร์สองตัวแรกขาด้านซ้าย	8
รูปที่ 3.3 แสดงขนาดชิ้นที่ได้ออกแบบ	9
รูปที่ 3.4 แสดงชิ้นส่วนแต่ละส่วนที่จะเป็นตัวเชื่อมระหว่างแท่นใหญ่กับเซอร์โวมอเตอร์ตัวที่สาม	10
รูปที่ 3.5 แสดงการประกอบเสร็จแล้วจากชิ้นส่วนในรูปที่ 3.4	10
รูปที่ 3.6 แสดงขนาดชิ้นส่วนในรูปที่ 3.4	11
รูปที่ 3.7 แสดงแผ่นยึดเซอร์โวมอเตอร์ตัวที่สาม	12
รูปที่ 3.8 แสดงภาพที่แผ่นยึดยึดกับมอเตอร์ตัวที่สามแล้วที่	12
รูปที่ 3.9 แสดงขนาดของแต่ละชิ้นส่วนปลายขาเต่าและลักษณะที่ประกอบแล้ว	13
รูปที่ 3.10 ภาพการประกอบชิ้นส่วนของขาทั้งหมดกับเซอร์โวมอเตอร์	14
รูปที่ 3.11 แสดงภาพลักษณะและขนาดของแท่นยึดมอเตอร์สองตัว	14
รูปที่ 3.12 แสดงขนาดของแผ่น PVC ส่วนบนและส่วนล่างของลำตัว	16
รูปที่ 3.13 ส่วนประกอบด้านข้างของลำตัวหุ่นยนต์เต่าที่เจาะไว้พร้อมยึดกับขาของหุ่นยนต์	17
รูปที่ 3.14 แสดงภาพด้านบนของตัวหุ่นยนต์เต่า	17
รูปที่ 3.15 แสดงภาพด้านข้างของตัวหุ่นยนต์เต่า	18
รูปที่ 3.16 แสดงภาพด้านหน้าของหุ่นยนต์เต่า	18
รูปที่ 3.17 ลักษณะของขาที่หล่อจากซิลิโคนสำเร็จแล้ว	19
รูปที่ 3.18 แสดงภาพขนาดของซิลิโคนที่หล่อ	20
รูปที่ 4.1 แสดงไดอะแกรมการควบคุมเซอร์โวมอเตอร์	22
รูปที่ 4.2 ลักษณะของสัญญาณที่ควบคุมเซอร์โวมอเตอร์	22
รูปที่ 4.3 Flowchart ของการควบการทำงานของหุ่นยนต์	24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

เรื่อง หน้า

รูปที่ 4.4 การเคลื่อนที่ของหุ่นยนต์เต่า ในการเคลื่อนที่ไปด้านหน้า 26
รูปที่ 4.5 การเคลื่อนที่ของหุ่นยนต์เต่า ในการเคลื่อนที่ไปด้านหน้าในน้ำ 27
รูปที่ 4.6 การลอยตัวของหุ่นยนต์เต่าในน้ำ 27
รูปที่ 4.7 การว่ายเลี้ยวซ้าย 28
รูปที่ 4.8 การว่ายเลี้ยวขวา 29
รูปที่ 4.9 แสดงรูปขาที่ประกอบเสร็จแล้ว 30
รูปที่ 4.10 รูปสัญญาณเมื่อหุ่นยนต์ยืนอยู่กับที่ 30
รูปที่ 4.11 รูปสัญญาณเมื่อหุ่นยนต์เดินหน้า 32
รูปที่ 4.12 รูปสัญญาณเมื่อหุ่นยนต์ว่ายน้ำ 33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมา

เนื่องจากในปัจจุบันมีบุคคลให้ความสำคัญกับหุ่นยนต์ขนาดเล็กกันมากยิ่งขึ้น โดยเฉพาะหุ่นที่เรียนแบบท่าทางของสัตว์ต่างๆ ดังนั้นเต่าก็เป็นสัตว์ที่น่าสนใจที่น่านำมาเรียนแบบให้เป็นหุ่นยนต์เต่า นอกจากนี้จะเห็นได้ว่าในการพัฒนาหุ่นยนต์นี้เป็นพื้นฐานอย่างดีในการประยุกต์ใช้ไมโครคอนโทรลเลอร์และจะรู้จักหลักการทำงานของไมโครคอนโทรลเลอร์ไปควบคุมระบบการขับเคลื่อนหรือควบคุมมอเตอร์นั่นเอง

ในโครงงานนี้เองได้เลือกใช้ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ในการออกแบบในการควบคุมส่วนในระบบการขับเคลื่อนเลือกใช้เซอร์โวมอเตอร์มาเป็นหัวใจของโครงงานนี้ โดยมีจุดประสงค์และขอบเขตของโครงงานนี้ดังนี้

1.2 ชื่อโครงงาน

หุ่นยนต์เต่า(TURTLE ROBOT)

1.3 จุดประสงค์

การสร้างและควบคุมหุ่นยนต์เต่าและศึกษาการทำงานของไมโครคอนโทรลเลอร์ ซึ่งในส่วนของฮาร์ดแวร์ของหุ่นยนต์เต่าจะเน้นการออกแบบที่ง่ายต่อการสร้าง ราคาถูกและสามารถเคลื่อนที่ได้ในระดับที่ต้องการ โดยจะแบ่งออกเป็น 2 ส่วนคือ

1) ส่วนประกอบทางแมคคานิกส์ ซึ่งจะกล่าวถึงโครงสร้างและส่วนประกอบของแมคคานิกส์ทั้งหมดของหุ่นยนต์เต่า เช่น ส่วนขา ตัวหุ่นยนต์ จะมีภาพประกอบอธิบาย
2) ส่วนประกอบของระบบควบคุมและการทำงานของหุ่นยนต์เต่า ซึ่งจะกล่าวถึงวงจรควบคุมและวงจรอิเล็กทรอนิกส์ต่างๆ เช่น ทรานซิสเตอร์ที่ใช้ขับเซอร์โวมอเตอร์ ลักษณะการต่อของไมโครคอนโทรลเลอร์กับทรานซิสเตอร์ที่ใช้ขับเซอร์โวมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ขอบเขตและผลที่คาดว่าได้รับ

โครงงานหุ่นยนต์เต่าที่คณะผู้จัดทำได้สร้างขึ้นนี้ ในความต้องการเบื้องต้นคือการสร้างหุ่นยนต์ที่มีลักษณะในการเคลื่อนที่ ท่าทาง เหมือนเต่า เราจึงได้ศึกษาท่าทางเพื่อที่จะเลียนแบบเต่า ลักษณะนิสัยของเต่านี้เป็นสัตว์ครึ่งบก ครึ่งน้ำ จากลักษณะนี้เองโครงงานหุ่นยนต์เต่านี้ ก็สามารถเคลื่อนที่ได้เหมือนเต่า สามารถลอยตัวในน้ำได้

1.5 เนื้อหาโดยสังเขป

บทที่ 2 เป็นระบบการควบคุมซึ่งจะกล่าวถึงทฤษฎีและหลักการของไมโครคอนโทรลเลอร์เซอร์โวมอเตอร์

บทที่ 3 เป็นการออกแบบและโครงสร้างของหุ่นยนต์เต่า ซึ่งจะกล่าวถึงการออกแบบส่วนขา ส่วนลำตัวและลักษณะโครงสร้างของหุ่นยนต์เต่า

บทที่ 4 เป็นการควบคุมและการเคลื่อนที่ของหุ่นยนต์เต่า

บทที่ 5 กล่าวถึงข้อเสนอแนะและปัญหาของโครงงาน

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการ

ทฤษฎีและหลักการที่นำมาศึกษาในโครงงานนี้มีทฤษฎีที่สำคัญคือหลักการการทำงานและโครงสร้างของไมโครคอนโทรลเลอร์ตระกูลMCS-51 และเซอร์โวมอเตอร์มีดังนี้

2.1 ไมโครคอนโทรลเลอร์ตระกูลMCS-51

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ได้ถูกออกแบบมาโดยรวมอุปกรณ์ที่สำคัญในการใช้งานของระบบไมโครคอลโทรลเลอร์ไว้ในภายในตัวมันเองไม่ว่าจะเป็น วงจรออสซิลเลเตอร์ วงจรผลิตสัญญาณนาฬิกา ตัวนับ พอร์ตอินพุต เอาต์พุต หน่วยความจำ วงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอล ระบบการอินเตอร์รัปต์ ระบบป้องกันความผิดพลาดที่จะเกิดขึ้นจากโปรแกรม และสามารถสรุปคุณสมบัติของไมโครคอลโทรเลอร์ได้ดังนี้

คุณสมบัติทั่วไปของไมโครคอนโทรลเลอร์ MCS-51

- เป็นไมโครคอนโทรลเลอร์ขนาด 8 บิต
- มีวงจรออสซิลเลเตอร์และวงจรผลิตสัญญาณนาฬิกาภายในไอซี
- มีขาสัญญาณอินพุตเอาต์พุตจำนวน 32 บิต
- สามารถเชื่อมต่อหน่วยความจำข้อมูลภายนอก(external data memory) โดยอ้างตำแหน่งแอดเดรสได้ถึง 64 K
- มีหน่วยความจำโปรแกรมภายในตัว (on-chip program memory) ขนาด 4 K โดยเฉพาะเบอร์ 8052 จะมีหน่วยความจำในส่วนนี้ถึง 8 K สำหรับเบอร์ 8031 และM8032 จะไม่มีหน่วยความจำในส่วนนี้
- มีหน่วยความจำข้อมูลภายในตัว (on-chip data memory) ขนาด 128 ไบต์ โดยเฉพาะเบอร์ 8052 และ 8032 จะมีหน่วยความจำในส่วนนี้ถึง 256 ไบต์
- หน่วยความจำข้อมูลภายในบางส่วนสามารถเข้าถึงข้อมูลระดับบิตได้ด้วย ทำให้การควบคุมหรือการตรวจสอบสถานะบิตทำได้ง่ายส่งผลให้การเขียนโปรแกรมทำได้ง่ายขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มีไทม์เมอร์/ เคาน์เตอร์(timer/counters) ขนาด 16 บิต จำนวน 2 ตัว โดยเฉพาะเบอร์ 8052 และ 8032 จะมีไทม์เมอร์/ เคาน์เตอร์จำนวน 3 ตัว
- การอินเตอร์รัปต์สามารถทำได้จาก 5 แหล่งกำเนิด โดยเฉพาะเบอร์ 8052 และ 8032 จะทำการอินเตอร์รัปต์ได้จาก 6 แหล่งกำเนิด
- มีพอร์ตสื่อสารอนุกรมภายในตัวเอง ซึ่งทำงานเป็นแบบฟูลดูเพล็กซ์(full duplex)
- มีพอร์ตแบบขนาน 4 พอร์ต
- มีคำสั่งในการคำนวณคณิตศาสตร์และทางตรรกศาสตร์
- คำสั่งส่วนใหญ่ใช้เวลาเพียง 1 ไมโครวินาที เมื่อใช้คริสตอลความถี่ 12 เมกะเฮิรตซ์
- ต้องการแหล่งจ่ายไฟ 5 โวลต์ เพียงชุดเดียว

2.2 เซอร์โวมอเตอร์

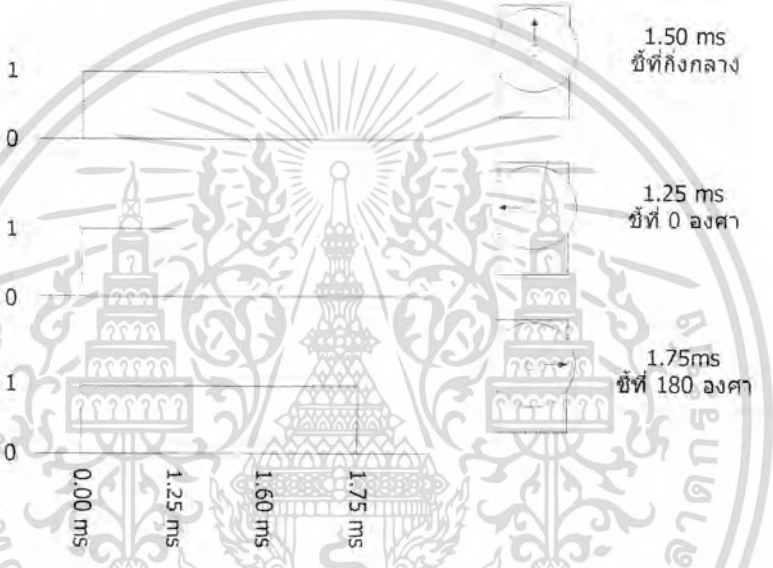
เซอร์โวมอเตอร์ประกอบด้วยมอเตอร์ที่มีความเร็วสูงภายในมีเฟืองทดรอบให้หมุนช้าลง เพื่อจะได้กำลังแรงบิดที่สูงขึ้น เซอร์โวมอเตอร์เป็นมอเตอร์ทดเฟืองขนาดเล็กโดยมีแกนส่งกำลัง 1 อัน โดยปกติเซอร์โวจะหมุนได้เพียง 180 องศา เท่านั้น หรือบางตัวอาจจะถึง 210 องศาขึ้นอยู่กับบริษัทผู้ผลิตและแกนส่งกำลังนี้สามารถที่จะควบคุมทิศทางที่หันไปหรือตำแหน่งที่ต้องการได้ โดยการส่งรหัสควบคุมให้กับตัวเซอร์โวมอเตอร์ ตำแหน่งของแกนกำลังจะอยู่มุมเดิมได้นั้นจะต้องมีการส่งสัญญาณดังกล่าวเข้าที่ขาอินพุตตลอดเวลา เซอร์โวมอเตอร์สามารถนำไปประยุกต์ใช้งานกับอุปกรณ์ได้หลากหลาย ส่วนใหญ่เราจะเห็นอยู่กับเครื่องเล่นที่บังคับด้วยวิทยุทั้งหลาย เช่น นำมาบังคับเลี้ยวและควบคุมอัตราการวิ่งของรถบังคับวิทยุ ,ควบคุมปีกของเครื่องบิน ,หรือหางเสือเรือและการเดินของหุ่นยนต์ เป็นต้น

หลักการทำงานของเซอร์โวมอเตอร์

มอเตอร์เซอร์โวมีวงจรควบคุมการหมุนโดยใช้ตัวต้านทานปรับค่าได้หรือ VR ค่า 5 กิโลโอร์ม ตัวต้านทานนี้จะติดอยู่กับแกนส่งกำลัง เพื่อใช้วัดระยะองศาของแกนหมุนโดยใช้ร่วมกับวงจรควบคุมการกำหนดมุมของเซอร์โว ถ้าแกนอยู่ในมุมที่ถูกต้องมอเตอร์ก็จะปิดเองอัตโนมัติแต่ถ้าวงจรตรวจสอบพบว่ามุมยังไม่ถูกต้องมอเตอร์ก็จะหมุนไปในทางที่ถูกจนกระทั่งได้มุมที่ถูกต้อง และจะหยุดอยู่ตรงนั้นตลอดไปจนกว่าจะมีการป้อนข้อมูลเข้ามาใหม่ ตัวแกนของเซอร์โวนี้สามารถหมุนได้เพียง 180-210 องศา (เนื่องจาก VR หมุนได้เท่านี้) และสามารถสั่งให้เซอร์โวทำการกำหนดองศาของการหมุนได้โดยการส่งสัญญาณพัลส์ค่าระหว่าง 1.5 มิลลิวินาที ถึง 2 มิลลิวินาที เซอร์โวมอเตอร์มอเตอร์จะมีสายไฟสามเส้น คือ สายไฟเลี้ยง สายไฟกราวด์ และสายสัญญาณพัลซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ควบคุม ซึ่งลักษณะของสัญญาณพัลซ์ที่ใช้ควบคุมตำแหน่งเซอร์โวมอเตอร์จะเป็นการส่งสัญญาณพัลซ์ที่มีความกว้างต่างกัน เพื่อให้เซอร์โวมอเตอร์หมุนไปในตำแหน่งที่ต้องการ โดยที่ความกว้างของพัลซ์จะเป็นตัวกำหนดขนาดและทิศทางของการหมุนเซอร์โวมอเตอร์ สำหรับคาบเวลาหรือระยะห่างระหว่างพัลซ์แต่ละลูกจะเป็นตัวกำหนดแรงบิดของเซอร์โวมอเตอร์ ส่วนในรูปที่ 3 นี้เป็นการแสดงลักษณะความกว้างของพัลส์ กับทิศทางที่เซอร์โวหมุนไป



รูปที่ 2.1 แสดงคาบเวลาของพัลส์ที่เป็นตัวกำหนดองศาเซอร์โว

จากรูปที่ 3 สรุปได้ว่าถ้าเราส่งพัลส์ที่มีขนาด 1.5 มิลลิวินาที จะทำให้เซอร์ปรับมุมไปที่ตำแหน่งกึ่งกลางของแกนหรือที่ 90 องศา ส่วนถ้าป้อนสัญญาณพัลส์ที่มีค่าน้อยกว่า 1.5 มิลลิวินาที เซอร์โวจะหมุนไปทางซ้ายหรือเข้าสู่ 0 องศา และถ้าป้อนสัญญาณพัลส์มากกว่า 1.5 มิลลิวินาที ก็จะทำให้เซอร์โวหมุนไปทางขวาหรือเข้าสู่องศาที่ 180 นั้นเอง สัญญาณพัลส์ก็คือสัญญาณ pulse width modulation (PWM) ซึ่งสามารถใช้เบสิกแสตมป์เป็นตัวส่งได้โดยใช้คำสั่ง puslout เพียงคำสั่งเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
โครงสร้างและการออกแบบของหุ่นยนต์เต่า

3.1 การออกแบบขา

ในการออกแบบลักษณะขาของหุ่นยนต์สิ่งที่ต้องคำนึงถึงเป็นสำคัญคือ ขาของตัวหุ่นยนต์ต้องสามารถเคลื่อนไหวได้สะดวกมากที่สุด โดยเฉพาะกับการออกแบบกับหุ่นยนต์เต่าต้องเคลื่อนไหวในน้ำและบนบกได้ วัสดุอุปกรณ์ในการประกอบที่ใช้ก็ต้องเหมาะสมกับการใช้งานนี้และที่สำคัญอีกอย่างขนาดของขาต้องสมดุลกับลำตัวหุ่นยนต์เต่าด้วยจึงทำการออกแบบดังนี้ โดยที่แต่ละขาจะติดตั้งมอเตอร์สามตัว

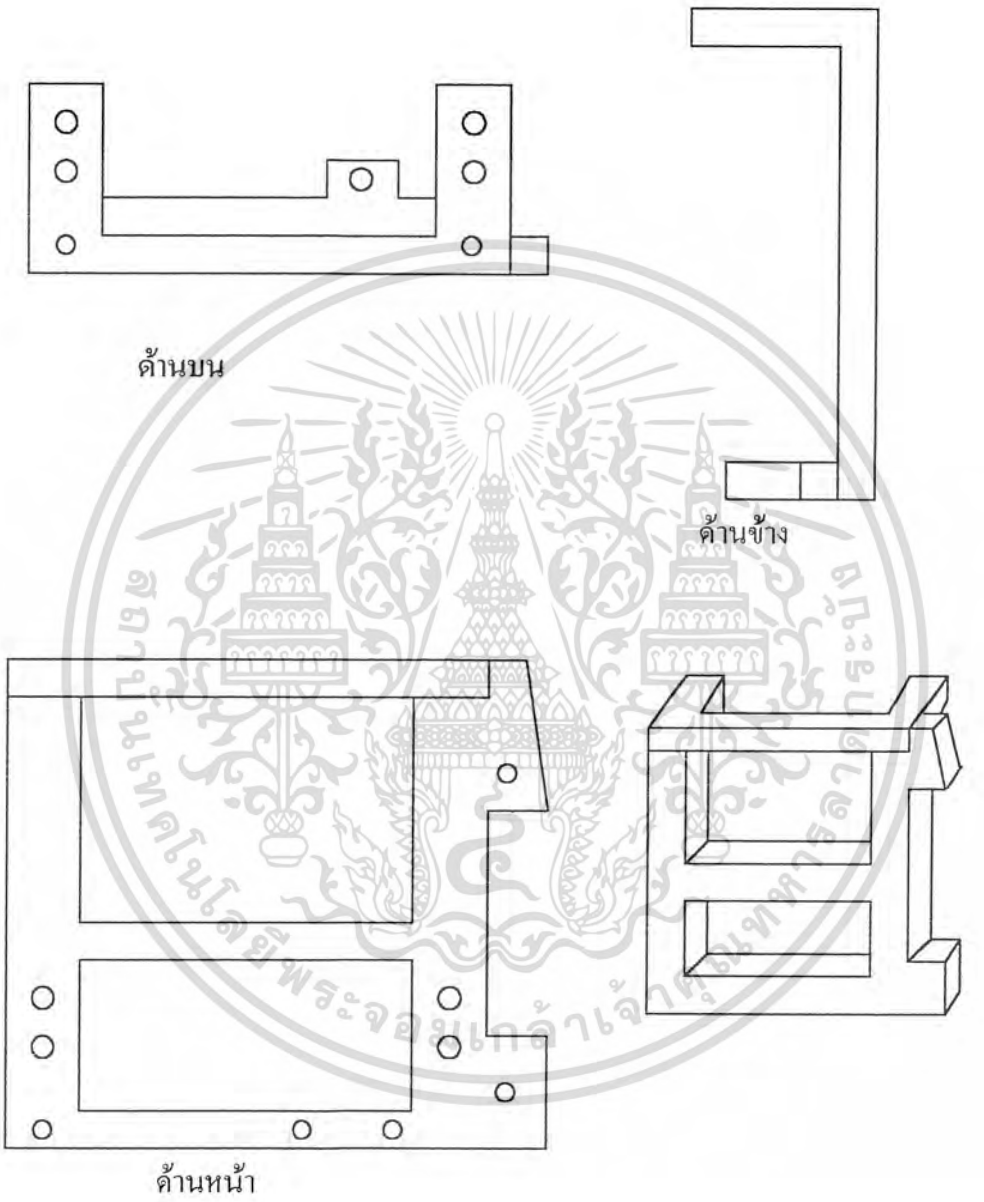
ขาที่ทำการออกแบบ

การออกแบบส่วนนี้ค่อนข้างยุ่งยากซับซ้อนกว่าส่วนอื่นดังนั้นวัสดุที่เลือกใช้ก็คือแผ่น PVC เพราะว่ามีคุณสมบัติที่ค่อนข้างเหนียว ดงทน และที่สำคัญง่ายต่อการตกแต่ง และการออกแบบขานี้เราต้องออกแบบชิ้นส่วนสำหรับยึดเซอร์โวมอเตอร์ทั้งสามตัว จากนี้เราสามารถแยกการออกแบบเป็นสองส่วนคือ ส่วนแรกชิ้นส่วนยึดเซอร์โวมอเตอร์สองตัว ส่วนที่สองยึดติดกับเซอร์โวมอเตอร์อีกหนึ่งตัว เป็นดังนี้

ส่วนแแรกจะประกอบไปด้วยชิ้นส่วนสามส่วนประกอบติดกันชิ้นส่วนส่วนบนจะทำหน้าที่ยึดเซอร์โวมอเตอร์ตัวที่ทำหน้าที่เป็นตัวหมุนลักษณะก้าวไปข้างหน้า ชิ้นส่วนที่สองมีลักษณะเป็นแท่นใหญ่หน่อยจะทำหน้าที่หลักเป็นตัวยึดเซอร์โวมอเตอร์ตัวที่ทำหน้าที่เป็นตัวยกขาและเป็นแกนหลักในการยึดชิ้นส่วนของขาแต่ละส่วนให้เข้ากัน ในการออกแบบส่วนนี้การออกแแบบขาด้านซ้ายและด้ารขวาแตกต่างกันเพียงเล็กน้อยตรงส่วนแท่นใหญ่ที่กลับกันคนละข้างเท่านั้นเองดังแสดงในรูปต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

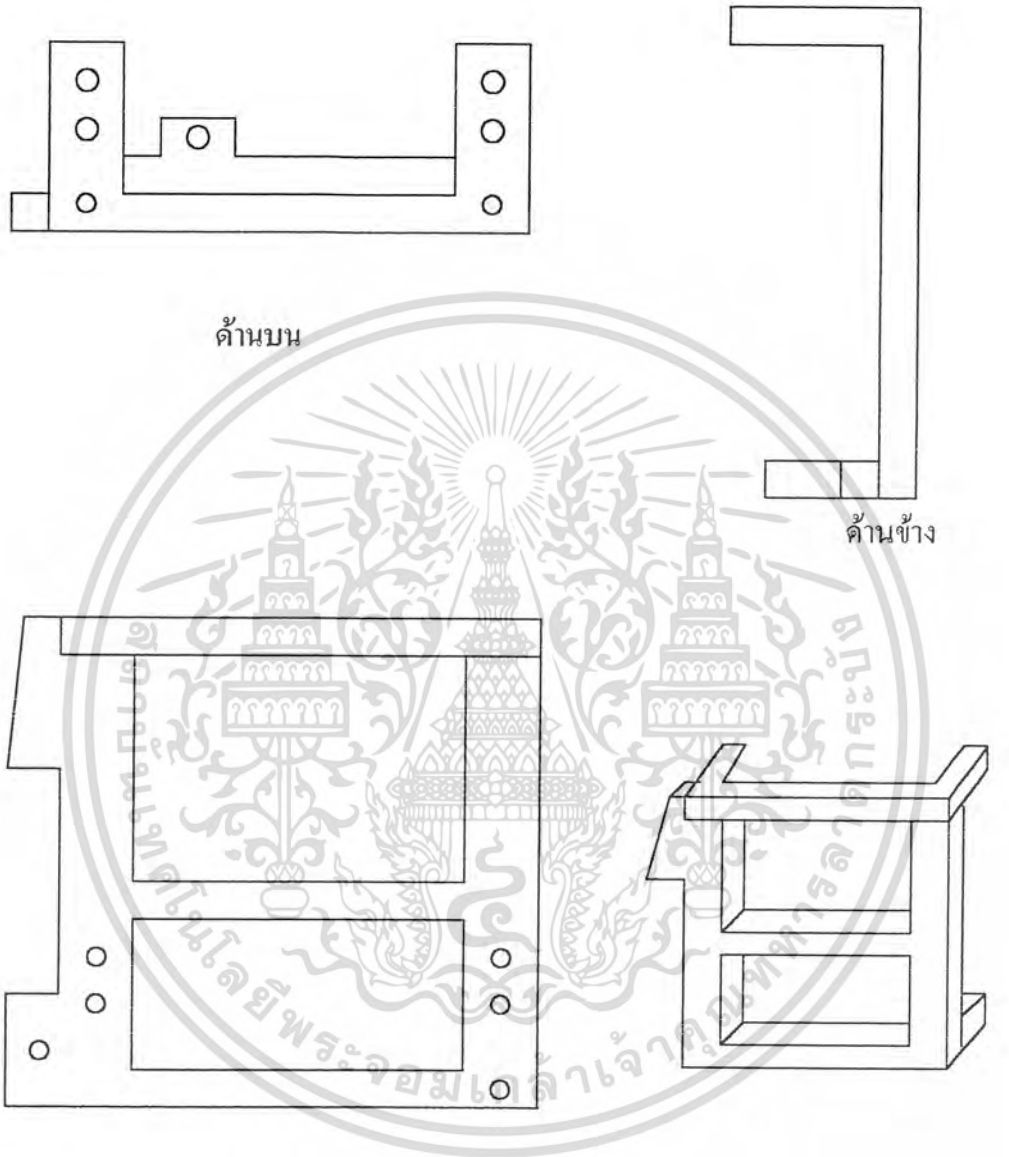
ขาด้านขวา



รูปที่ 3.1 แสดงภาพแท่นที่ใช้ยึดเซอร์โวมอเตอร์สองตัวแรกขาด้านขวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

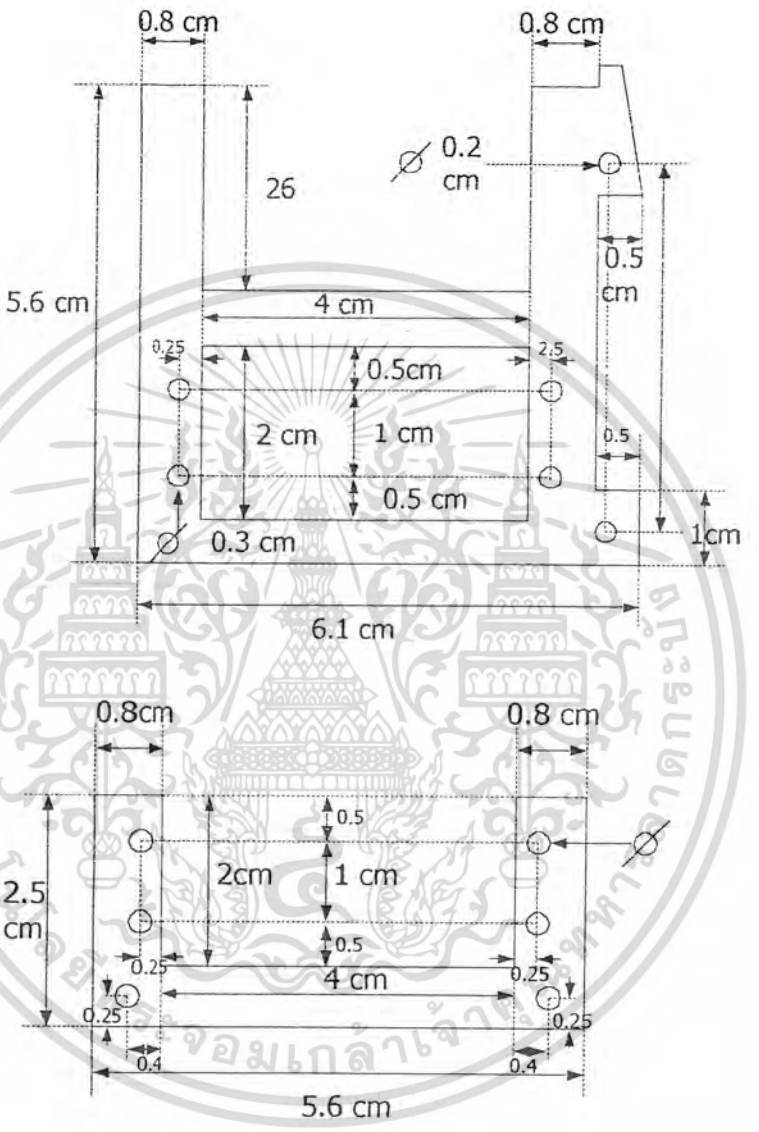
ขาด้านซ้าย

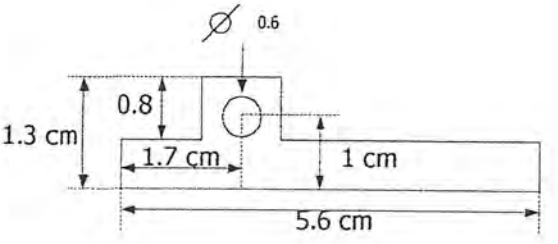


รูปที่ 3.2 แสดงภาพแท่นที่ใช้ยึดเซอร์โวมอเตอร์สองตัวแรกขาด้านซ้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.1 และรูปที่ 3.2 มีขนาดที่ได้ออกแบบแต่ละชิ้นส่วนดังนี้

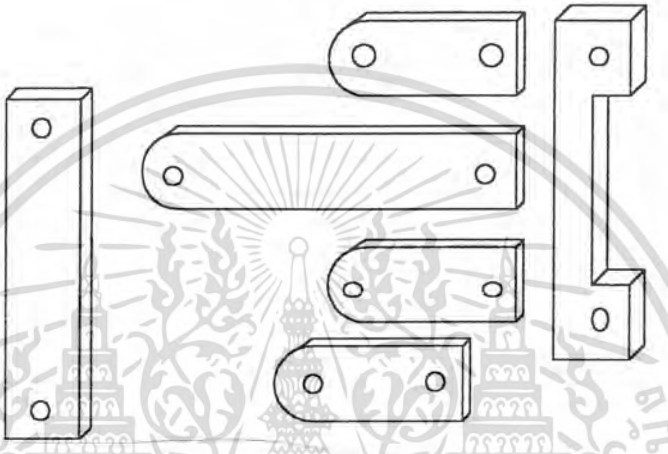




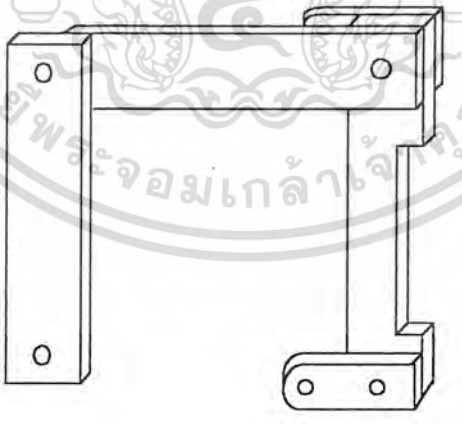
รูปที่ 3.3 แสดงขนาดชิ้นที่ได้ออกแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนที่สองนี้มีชิ้นส่วนที่ค่อนข้างละเอียด ส่วนประกอบส่วนนี้จะเชื่อมระหว่างเซอร์โวมอเตอร์สองตัวที่กล่าวมากับเซอร์โวมอเตอร์ที่ทำหน้าที่หมุนปลายขาในขณะที่อยู่ในน้ำโดยจะมีแผ่นยึดกับตัวเซอร์โวมอเตอร์ก่อน ส่วนประกอบส่วนที่สองนี้จะมีชิ้นส่วนขาซ้ายกับขาขวาเหมือนกันดังแสดงในรูปต่อไปนี้

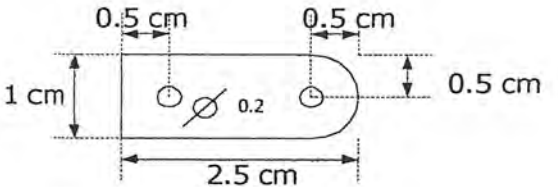


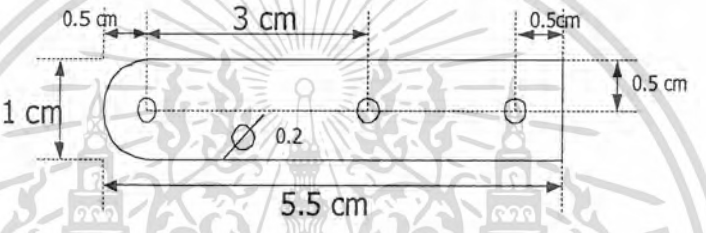
รูปที่ 3.4 แสดงชิ้นส่วนแต่ละชิ้นที่จะเป็นตัวเชื่อมระหว่างแท่นใหญ่กับเซอร์โวมอเตอร์ตัวที่สาม

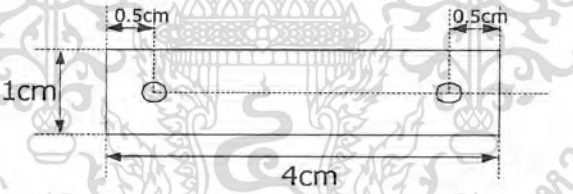


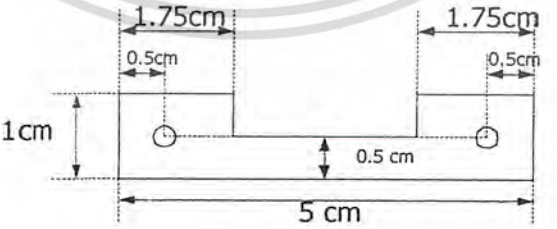
รูปที่ 3.5 แสดงการประกอบเสร็จแล้วจากชิ้นส่วนในรูปที่ 3.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





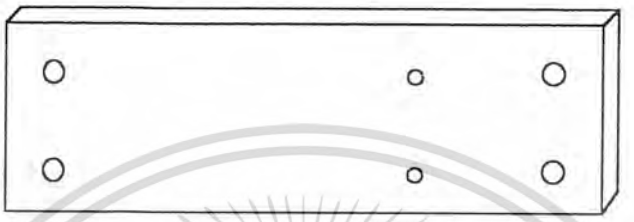




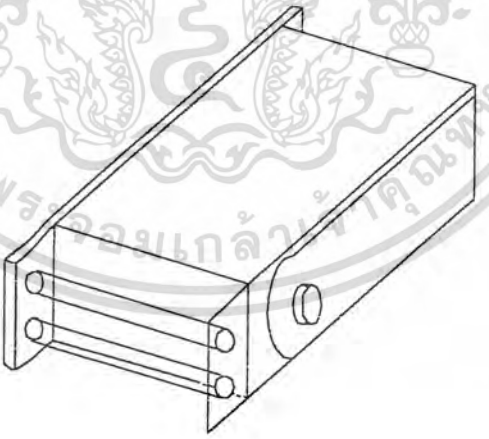
รูปที่ 3.6 แสดงขนาดของชิ้นส่วนจากรูปที่ 3.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.5 จะสามารถยึดกับเซอร์โวมอเตอร์อีกตัวได้จะต้องมีแผ่นยึดก่อน



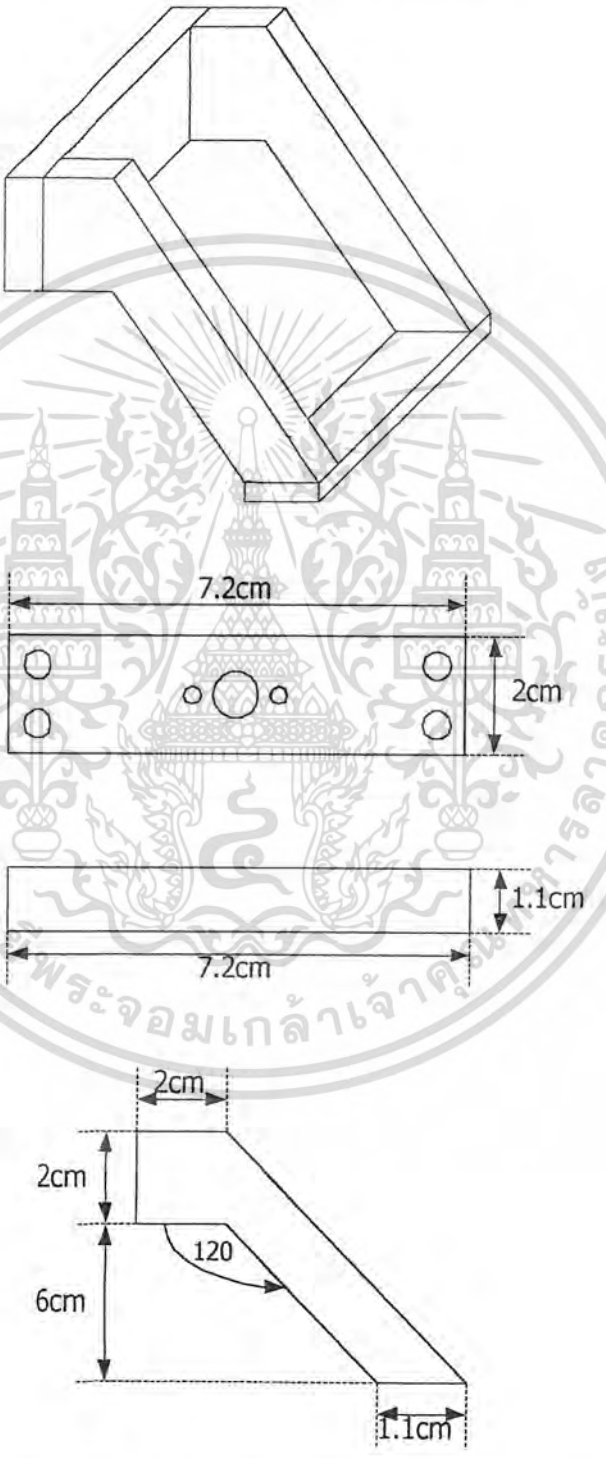
รูปที่ 3.7 แสดงแผ่นยึดเซอร์โวมอเตอร์ตัวที่สาม



รูปที่ 3.8 แสดงภาพที่แผ่นยึดยึดกับมอเตอร์ตัวที่สามแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

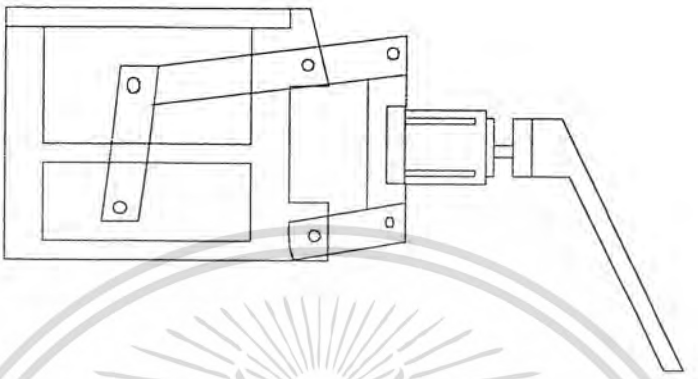
จากรูปเซอร์โวมอเตอร์ที่ผ่านมานี้เซอร์โวมอเตอร์ตัวนี้ก็จะยึดอยู่กับปลายขาของหุ่นยนต์เต่าโดยจะมีชิ้นส่วนที่จะประกอบกันเป็นปลายขามีขนาดและลักษณะดังนี้



รูปที่ 3.9 แสดงขนาดของแต่ละชิ้นส่วนปลายขาเต่าและลักษณะที่ประกอบแล้ว

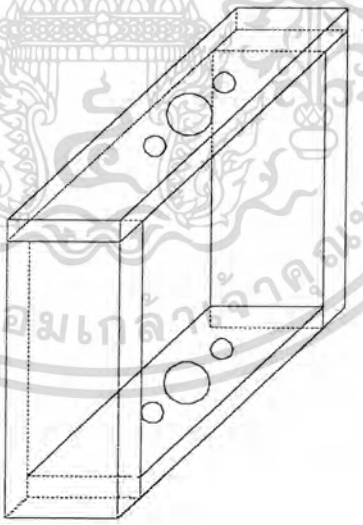
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

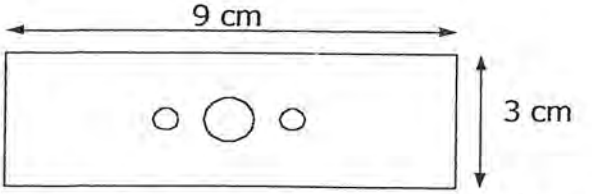
จากการออกแบบขาทั้งหมดที่กล่าวมานี้เมื่อประกอบชิ้นส่วนทั้งหมดกับเซอร์โวมอเตอร์ทั้งสามตัวจะมีลักษณะดังนี้



รูปที่ 3.10 ภาพการประกอบชิ้นส่วนของขาทั้งหมดกับเซอร์โวมอเตอร์

จากรูปที่ 3.10 นี้จำเป็นต้องมีแกนยึดกับเซอร์โวมอเตอร์ทั้งสองแรกเพื่อเป็นจุดศูนย์กลางในการหมุนและการยึดติดกับลำตัว โดยที่แกนนั้นจะมีลักษณะและขนาดดังนี้





รูปที่ 3.11 แสดงภาพลักษณะและขนาดของแท่นยึดมอเตอร์สองตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการออกแบบขาใช่ว่าเราจะต้องให้ความสนใจแต่เพียงส่วนประกอบของขาเราจะต้องคำนึงถึงข้อต่อด้วย จึงจะทำให้การออกแบบง่ายขึ้นและมีปัญหาน้อยลง

การทำงานของข้อต่อ

มีหลายวิธีที่จะทำให้ข้อต่อทำงานโดยใช้ลักษณะการเคลื่อนของข้อต่อมาเลือกใช้วิธีที่จะทำให้ข้อต่อทำงานแต่ถึงอย่างการติดตั้งมอเตอร์นั้นสามารถทำการติดตั้งโดยตรงกับข้อต่อได้เลย จากจุดนี้เราจำเป็นต้องติดตั้งตัวขับเคลื่อนอยู่ใกล้กับข้อต่อมากที่สุด การติดตั้งเช่นนี้ทำให้มีผลต่อทางไดนามิคต่อขาหุ่นยนต์แต่ก็อาจจะชดเชยโดยการใช้ตัวควบคุม หรืออาจต้องเลือกใช้มอเตอร์ที่มีกำลังสูงๆ เพื่อใช้ในการขับเคลื่อนที่มีน้ำหนักมาก ในระบบไฮโดรริกส์ นิวเมติกส์ หรือมอเตอร์ ซึ่งก็ต้องขึ้นอยู่กับความเหมาะสมและความต้องการที่จะนำไปใช้ไม่ว่าจะเป็นแบบใดการพิจารณาในการออกแบบคือขนาดของการเคลื่อนไหวของขา เช่นการยกขา การหมุนเป็นต้น

ที่กล่าวมาทั้งหมดนี้จะมีผลโดยตรงกับกับขนาดของขานั่นก็คือ ความยาวของขนาดของขา ความสามารถที่หุ่นยนต์สามารถยกขึ้น การหมุน ผลที่ได้ก้ต้องหมายถึงการเคลื่อนไหวแต่ละครั้งก็ต้องมีการเปลี่ยนแปลงไปด้วย

ดังนั้นในการออกแบบสิ่งที่สำคัญอีกอย่างคือ ต้องพยายามให้ตัวขับเคลื่อนอยู่ติดกับฐานของขา การติดตั้งชุดการขับเคลื่อนต้องอยู่ศูนย์กลางจะให้ผลกระทบทางไดนามิคลดลง แล้วยังลดภาระทางไดนามิคแต่ละขาไปอีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การออกแบบโครงสร้างของตัวหุ่นยนต์

วัสดุที่เลือกใช้ต้องมีความทนทาน เหนียว ตัดประกอบและตกแต่งง่าย มีน้ำหนักเบา จากคุณนี้ก็เช่นเดียวกับวัสดุส่วนขา เราจึงเลือกใช้แผ่น PVC ในการสร้างประกอบนี้ ส่วนจุดหมุนและข้อต่อทุกจุดเราใช้ลูกปืนแบริ่งขนาด 0.25 นิ้วตามขนาดและสัดส่วนของตัวหุ่นยนต์เพื่อลดแรงเสียดทานของจุดหมุนทุกจุด และเราก็ได้ออกแบบส่วนลำตัวตามลักษณะและขนาดดังนี้

หุ่นยนต์เต่าที่สร้างมีขนาด 20x36x12.5 เซนติเมตร

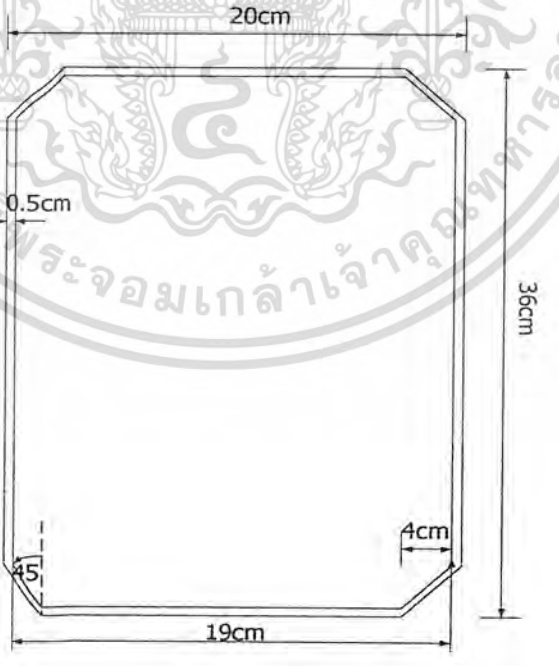
หุ่นยนต์เต่ามีน้ำหนัก 3 กิโลกรัม

ตัวหุ่นยนต์สูงจากพื้น 4 เซนติเมตร

หุ่นยนต์ยกขาสูงจากพื้น 2 เซนติเมตร

แต่ละขาเคลื่อนไหวอิสระต่อกัน

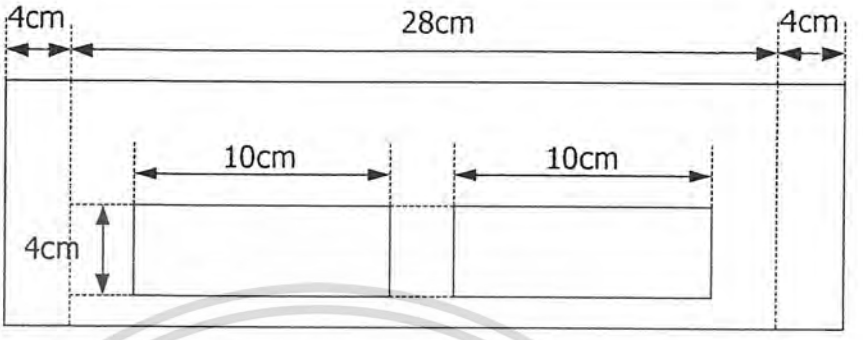
ลำตัวหุ่นยนต์เราได้ทำการออกแบบเป็นแปดเหลี่ยม(ออกแบบใกล้เคียงวงรีเพื่อให้มีลักษณะลำตัวเต่าที่สุดและเหมาะสม) ประกอบไปด้วยแผ่น PVC ทั้งหมด 10 แผ่น ด้านข้างจะเจาะสี่เหลี่ยมไว้ประกอบกับขาและเมื่อประกอบกันทั้งหมดแล้วก็จะมีพื้นไว้จัดวางอุปกรณ์ควบคุม โดยมีขนาดของส่วนประกอบแต่ละส่วนดังนี้



รูปที่ 3.12 แสดงขนาดของแผ่น PVC ส่วนบนและส่วนล่างของลำตัว

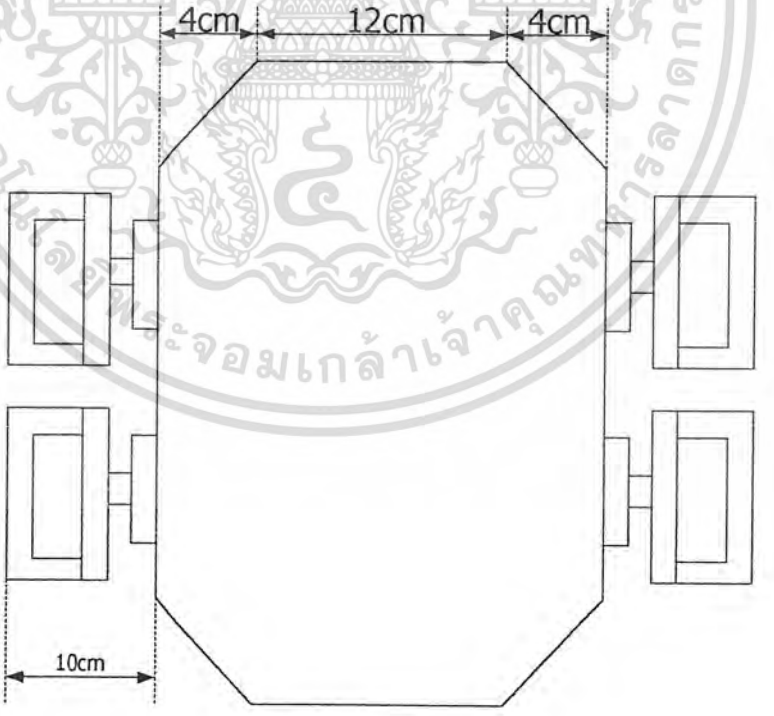
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



รูปที่ 3.13 ส่วนประกอบด้านข้างของลำตัวหุ่นยนต์เต่าที่เจาะไว้พร้อมยึดกับขาของหุ่นยนต์

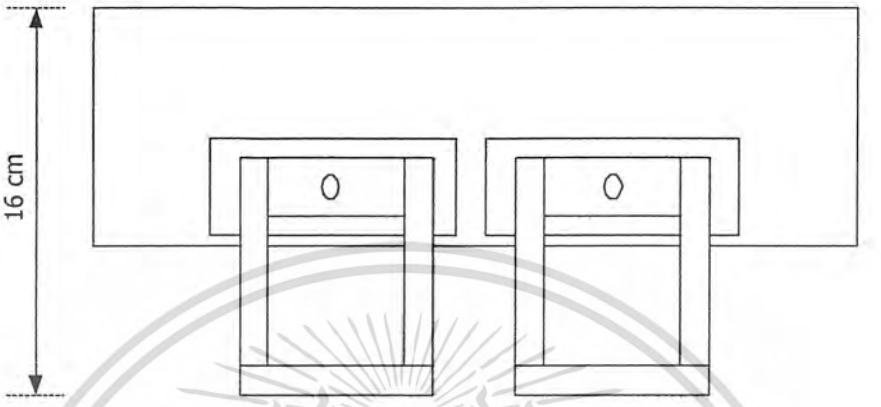
จากนี้เมื่อเราได้ทำการประกอบเซอร์โวมอเตอร์กับส่วนประกอบของขาหุ่นยนต์แล้วมาประกอบกับลำตัวก็จะได้ดังภาพต่อไปนี้



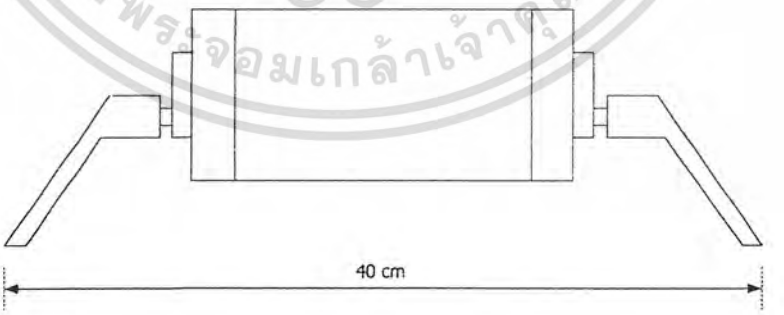
รูปที่ 3.14 แสดงภาพด้านบนของตัวหุ่นยนต์เต่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

42255



รูปที่ 3.15 แสดงภาพด้านข้างของตัวหุ่นยนต์เต่า



รูปที่ 3.16 แสดงภาพด้านหน้าของหุ่นยนต์เต่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 ซิลิโคนหุ้มขาหุ่นยนต์

เนื่องจากหุ่นยนต์เต่าต้องสามารถเคลื่อนไหวในน้ำและบนบกได้จากการออกแบบที่ผ่านมาเป็นเพียงการเคลื่อนบนบกเราต้องการให้ตัวหุ่นยนต์เคลื่อนไหวในน้ำด้วยจึงต้องทำการออกแบบเพื่อป้องกันปัญหาของขาตอนเคลื่อนไหวในน้ำไม่มีน้ำเข้าไปทำความเสียหายกับระบบควบคุมหุ่นยนต์เต่า ป้องกันโดยการซิลล์(seal) จะใช้ซิลิโคน(silicon) เนื่องจากมีคุณสมบัติที่ยืดหยุ่นและสามารถหล่อได้เป็นรูปตามความต้องการ การหล่อซิลิโคน(silicon) มีส่วนผสมคือ

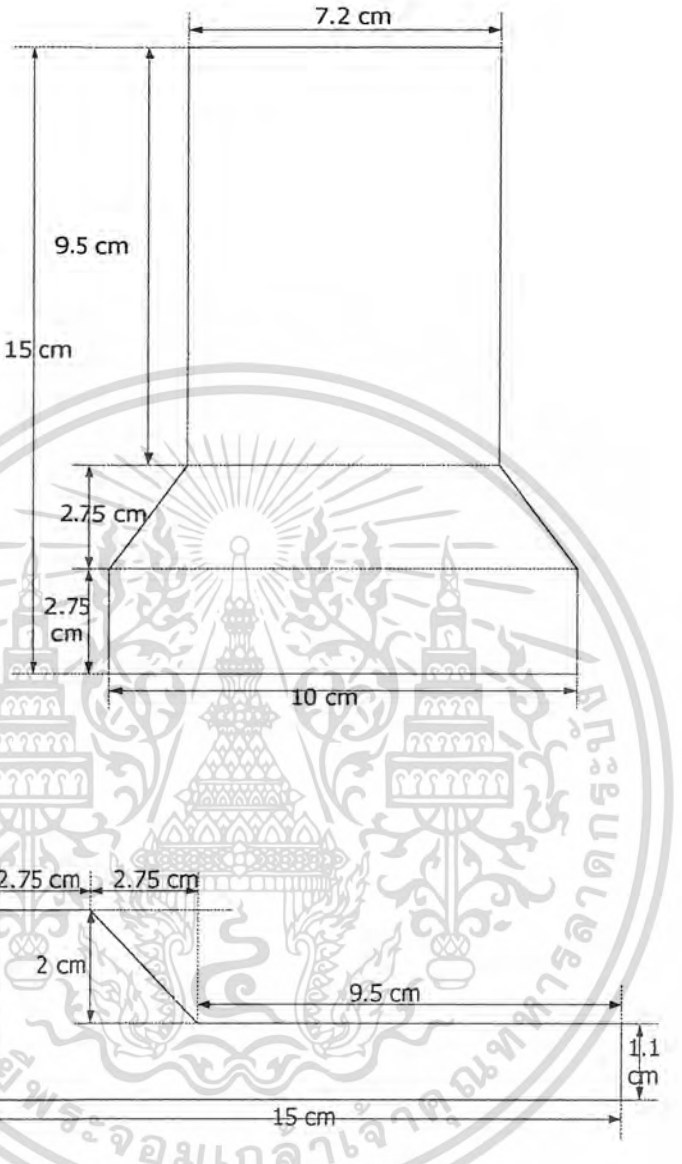
1.ยางซิลิโคน(silicon)

2.น้ำยาเร่งการแข็งตัวยางซิลิโคน

การผสมซิลิโคน(silicon) ผสมด้วยอัตราส่วนยางซิลิโคน(silicol)ต่อน้ำยาเร่งการแข็งตัวยางซิลิโคนเป็น 100:2 การแข็งตัวของยางซิลิโคน(silicon)ขึ้นอยู่กับน้ำยาเร่งการแข็งตัวของยางเพราะฉะนั้นต้องระวังในการผสมด้วย การหล่อซิลิโคนคุมขาหุ่นยนต์เต่าไว้มีลักษณะดังนี้

รูปที่ 3.17 ลักษณะของขาที่หล่อจากซิลิโคนสำเร็จแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.18 แสดงภาพขนาดของซิลิโคนที่หล่อ

จากการออกแบบนี้เราสามารถสรุปการทำงานของมอเตอร์แบ่งออกเป็นสามส่วนแต่ละส่วนก็ทำหน้าที่ต่างกันไป

ส่วนที่ 1 เป็นส่วนที่ใช้ในการให้หุ่นเคลื่อนที่ไปด้านหน้าหมุนไปประมาณ 60องศา

ส่วนที่ 2 เป็นส่วนที่ใช้ยกส่วนที่สามให้สูงจากพื้น โดยยกสูงจากพื้นได้ประมาณ 2 เซนติเมตร

ส่วนที่ 3 เป็นส่วนที่ใช้ในตอรเคลื่อนไหวในน้ำในลักษณะหมุนเพื่อให้หุ่นเคลื่อนที่ โดยหมุนประมาณ 90 องศา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ระบบการควบคุม

4.1 หลักการควบคุม

การควบคุมเซอร์โวมอเตอร์ทำได้โดยป้อนความถี่พัลซ์บวก ดังนั้นเราจึงสามารถใช้ไมโครคอนโทรลเลอร์สร้างพัลซ์ขึ้นมาควบคุมเซอร์โวมอเตอร์ทั้ง 12 ตัว เพื่อให้หมุนไปตามตำแหน่งที่เราต้องการได้ และพัลซ์บวกที่สร้างขึ้นมาจะมีความถี่ไม่เท่ากัน ถ้าตำแหน่งการหมุนของเซอร์โวมอเตอร์ต่างกันตลอดเวลาของการทำงาน จำเป็นที่ต้องสร้างพัลซ์ให้กับเซอร์โวมอเตอร์ตลอดเวลาเพื่อให้เซอร์โวมอเตอร์รักษาตำแหน่งของตัวเองไว้

จากโครงสร้างของตัวหุ่นยนต์ที่สร้างขึ้นเซอร์โวมอเตอร์ทางด้านซ้ายของตัวหุ่นยนต์และเซอร์โวมอเตอร์ทางด้านขวาของตัวหุ่นยนต์จะวางอยู่ในตำแหน่งตรงข้ามกันทั้งสองด้าน นั่นหมายถึงความถี่พัลซ์ที่ป้อนให้กับเซอร์โวทั้งด้านซ้ายและขวาจะต้องต่างกัน

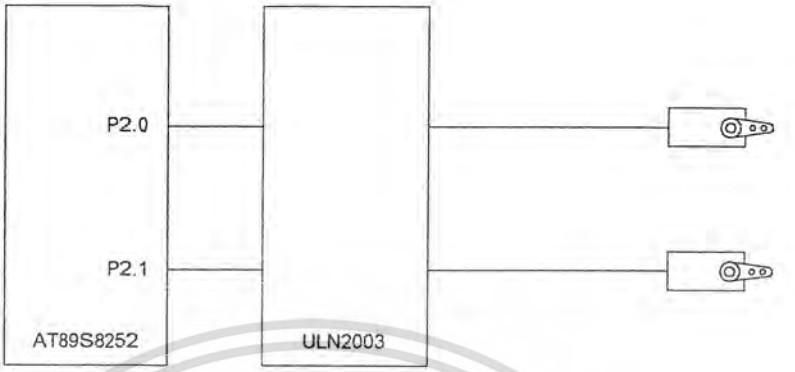
4.1.Iการเชื่อมต่อไมโครคอนโทรลเลอร์เข้ากับเซอร์โวมอเตอร์

ในการควบคุมเราเลือกใช้ ไมโครคอนโทรลเลอร์ AT89S8252 ผ่านทางไอซีบัฟเฟอร์ ULN2003 โดยใช้สัญญาณควบคุม 1 บิตต่อมอเตอร์ 1 ตัว ดังแสดงในรูปที่ 4.1 ซึ่งเป็นไดอะแกรมการควบคุมเซอร์โวมอเตอร์ และมีการเชื่อมต่อพอร์ตเข้ากับเซอร์โวมอเตอร์คือพอร์ต 2 บิตที่ 0 ถึงบิตที่ 1 เชื่อมต่อกับมอเตอร์ 2 ตัว

เพราะในการควบคุมเราต้องการควบคุมให้มอเตอร์ที่อยู่มุมทะแยงกันทำงานพร้อมกัน ฉะนั้นการทำงานของมอเตอร์จะทำงานพร้อมกันเป็นคู่ๆ

ในการควบคุมเซอร์โวมอเตอร์จะใช้ ไมโครคอนโทรลเลอร์ 1 ตัวต่อมอเตอร์ 2 ตัว ซึ่งไมโครคอนโทรลเลอร์ชุดนี้จะมีทั้งหมด 2 ชุดต่อขาหุ่น 1 ขา เพื่อให้ง่ายต่อการเขียนโปรแกรม และแต่ละชุดจะถูกควบคุมโดยไมโครคอนโทรลเลอร์อีกตัวหนึ่ง ซึ่งจะเป็นตัวควบคุมและเรียกใช้งานไมโครคอนโทรลเลอร์แต่ละชุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

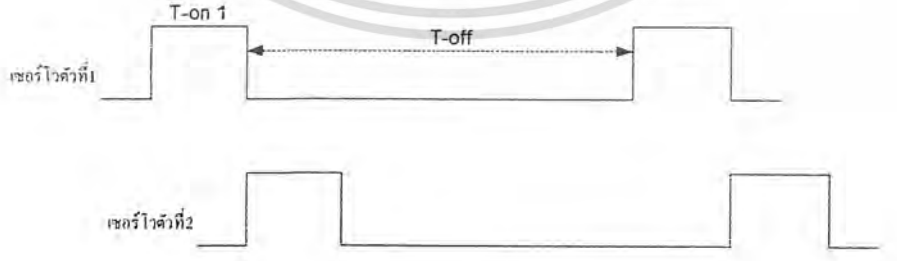


รูปที่ 4.1 แสดงไดอะแกรมการควบคุมเซอร์โวมอเตอร์

4.2 แนวทางในการควบคุม

4.2.1 การจัดสัญญาณพัลซ์ให้กับเซอร์โวมอเตอร์

การที่จะให้ไมโครคอนโทรลเลอร์สร้างความถี่ต่างกันในเวลาที่พร้อมกันนั้นทำได้ยาก เราหลีกเลี่ยงโดยการสร้างสัญญาณพัลซ์บวกตามเวลา T-on ให้กับเซอร์โวมอเตอร์แต่ละตัว โดยอาศัยช่วงจังหวะ T-off สร้างสัญญาณ T-on ให้กับเซอร์โวมอเตอร์ตัวต่อไปจนครับทั้ง 2 ตัว ดังรูปที่ 4.2 แล้วจะใช้ไมโครคอนโทรลเลอร์อีกตัวหนึ่งเพื่อใช้ควบคุมและเรียกใช้งานไมโครคอนโทรลเลอร์แต่ชุด



รูปที่ 4.2 ลักษณะของสัญญาณที่ควบคุมเซอร์โวมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อครบรอบการทำงานจะมีการวนลูป ประมวลผลเพื่อควบคุมขาหุ่นยนต์ในแต่ละตำแหน่งซึ่งตำแน่งของขาในแต่ละลำดับ จะวนมาสร้างสัญญารพัลซ์ให้กับเซอร์โวมอเตอร์แต่ละคู่อีก เพื่อย้ายตำแหน่งของขาหุ่นให้เคลื่อนที่ในตำแหน่งที่ต้องการ ซึ่งการวนลูปอย่างนี้เรื่อยๆ ไปจะทำให้ดูเหมือนว่าเซอร์โวมอเตอร์ได้รับพัลซ์พร้อมกัน

4.2.2 การสร้างสัญญาณพัลซ์โดยไมโครคอนโทรเลอร์

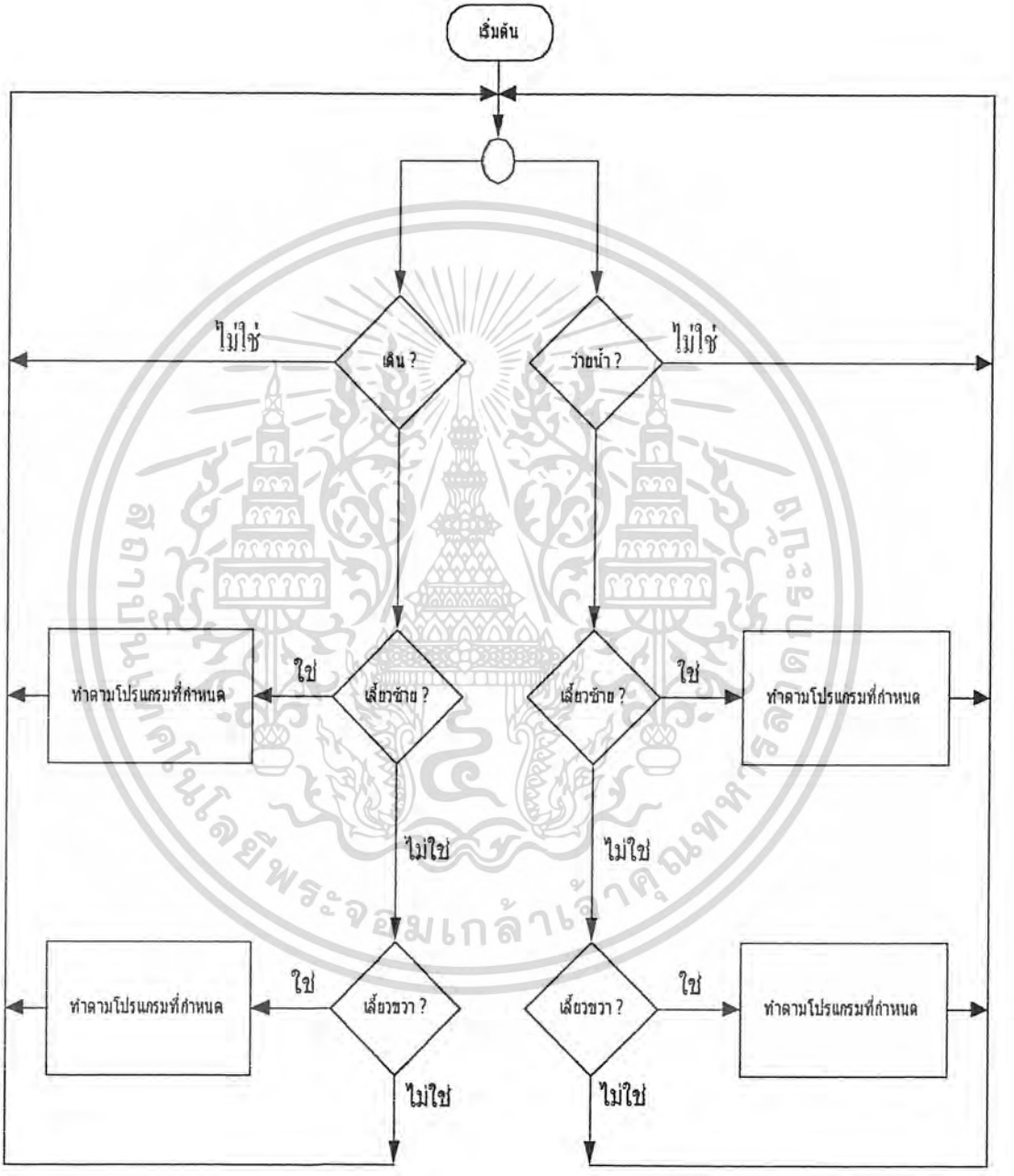
หลักการสร้างสัญญาณพัลซ์โดยไมโครคอนโทรเลอร์ คือสัญญาณข้อมูล 1 บิต สำหรับ 1 ช่องสัญญาณพัลซ์โดยมีขั้นตอนดังนี้

1. ทำการส่งข้อมูล '1' ไปยังบิตที่ต้องการให้เกิดพัลซ์บวก หรือ ช่วง T-on
2. ทำการหน่วงเวลาตามระยะเวลาของ T-on
3. ส่งข้อมูล '0' เมื่อต้องการให้อยู่ในช่วง T-off
4. ทำการหน่วงเวลาตามระยะเวลาของ T-off
5. ทำการวนลูปกลับไปขั้นตอนที่ 1 อีกครั้ง

หลักการดังกล่าว เป็นหลักการที่ง่ายสำหรับการสร้างพัลซ์ ช่องสัญญาณเดียว แต่ถ้าหากต้องการสร้างสัญญาณพัลซ์หลายๆช่องสัญญาณที่ความถี่ต่างกันตลอดเวลา ซึ่งหลักการดังที่กล่าวข้างต้นจะเป็นหลักการที่นำไปสู่การเขียนโปรแกรมที่ซับซ้อนมากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flowchart



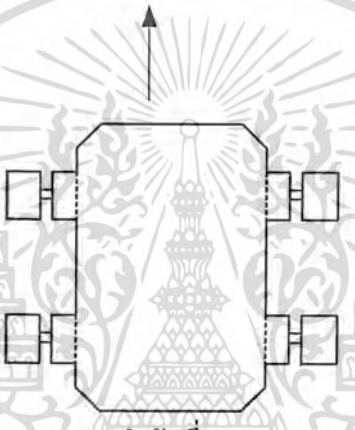
รูปที่ 4.3 Flowchart ของการควบการทำงานของหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

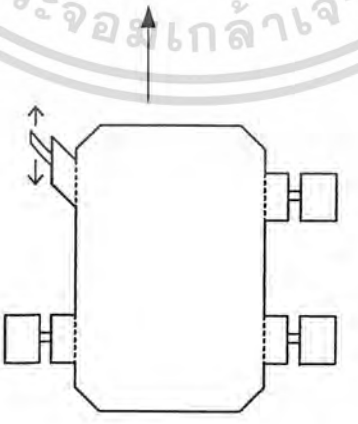
4.3 การทดลองการเคลื่อนที่ของหุ่นยนต์เต่า

4.3.1 การทดลองการเคลื่อนที่ของหุ่นยนต์เต่า ในการเคลื่อนที่ไปด้านหน้า

ในหัวข้อนี้ได้นำเสนอหลักการทดสอบแนวคิดและหลักการเคลื่อนที่ของหุ่นยนต์ในทิศทางต่างๆ โดยหลักการเคลื่อนที่จะเคลื่อนที่ทีละหนึ่งขา แสดงดังรูปที่ 4.4 แต่จะเกิดข้อเสียเมื่อขาของหุ่นยนต์เต่าเคลื่อนที่ในพื้นที่ ที่ไม่ใช่พื้นราบ จะไม่สามารถกำหนดทิศทางและเคลื่อนที่ไปผิดจากทิศทางที่กำหนด

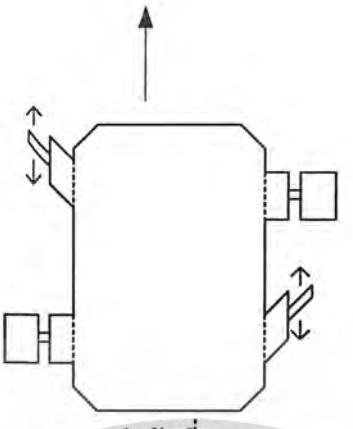


ลำดับที่ 1

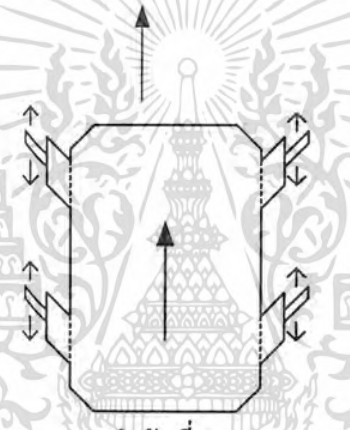


ลำดับที่ 2

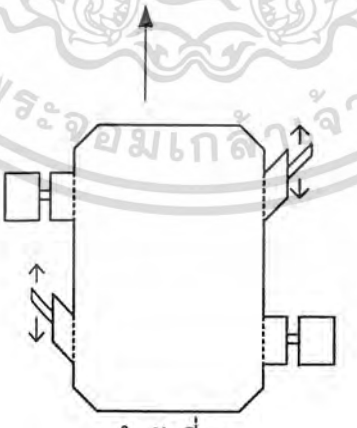
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ลำดับที่ 3



ลำดับที่ 4



ลำดับที่ 5

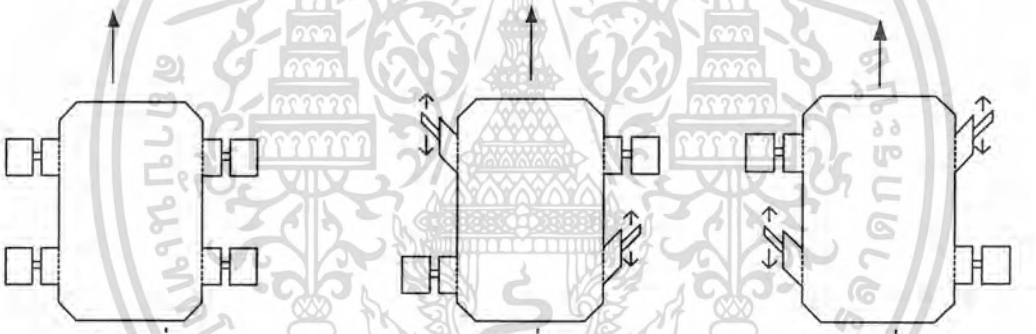
รูปที่ 4.4 การเคลื่อนที่ของหุ่นยนต์เต่า ในการเคลื่อนที่ไปด้านหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นจากผลการทดลองในการเคลื่อนที่ไปด้านหน้าของหุ่นยนต์เต่า โดยแต่ละขาประกอบด้วยมอเตอร์ สามตัวในการเดินนั้นกำหนดให้โปรแกรมเลือกใช้มอเตอร์แค่สองตัว และเลือกใช้ในการเคลื่อนที่บนผิวน้ำสองตัวสลับกัน คือ ตัวที่ทำหน้าที่ยกขาให้ลอยสูงจากพื้นดิน และตัวที่ทำหน้าที่ผลักดันตัวหุ่นยนต์ให้เคลื่อนที่ไปด้านหน้าโดยใช้ค่าทดสอบมุมยก 10 องศา และมุมผลักดัน 120 องศา

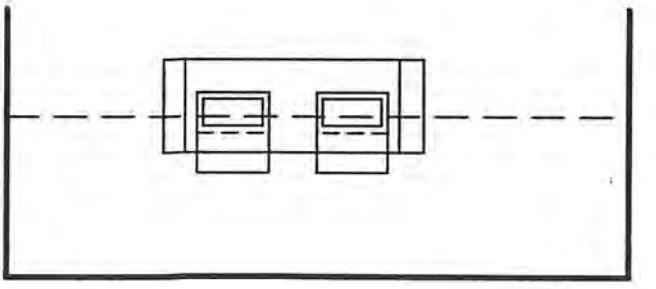
4.3.2 การทดลองการเคลื่อนที่ของหุ่นยนต์เต่า ในการเคลื่อนที่ไปด้านหน้าในน้ำ

จากแนวคิดในการเดินการเดินไปด้านหน้ากำหนดให้โปรแกรมเลือกใช้มอเตอร์สองตัวดังที่ได้กล่าวข้างต้น ตัวแรกทำหน้าที่เคลื่อนที่ไปด้านหน้า-หลังตามรูปที่ 4.4 ตัวที่สองจะทำหน้าที่หมุนต้านน้ำในตอนที่มอเตอร์ตัวแรกเคลื่อนที่มาด้านหลังเพื่อให้หุ่นมาพื้นที่ต้านน้ำมากและสามารถเลื่อนที่ได้ ในการทดสอบการเคลื่อนที่ในน้ำใช้มุมก้าว 30 องศา และ90 องศา ตัวแรกและตัวที่สองตามลำดับ ความเร็วในการเคลื่อนที่ในน้ำสามารถเคลื่อนที่ได้ 30วินาทีต่อฟุต



รูปที่ 4.5 การเคลื่อนที่ของหุ่นยนต์เต่า ในการเคลื่อนที่ไปด้านหน้าในน้ำ

ในการเคลื่อนที่ในน้ำหุ่นสามารถลอยตัวในน้ำได้ดังรูปที่ 4.6

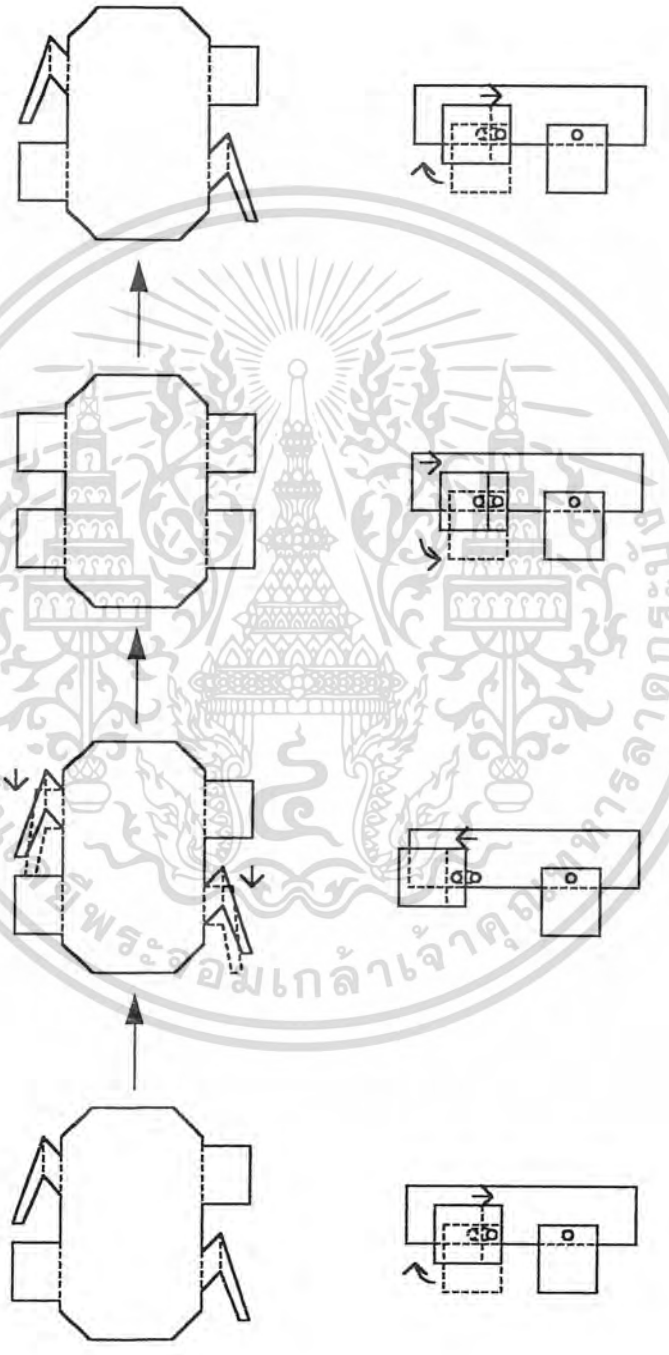


รูปที่ 4.6 การลอยตัวของหุ่นยนต์เต่าในน้ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเคลื่อนที่แบบหมุนตัวเมื่อเคลื่อนที่ในน้ำ

-การว่ายเลี้ยวซ้าย

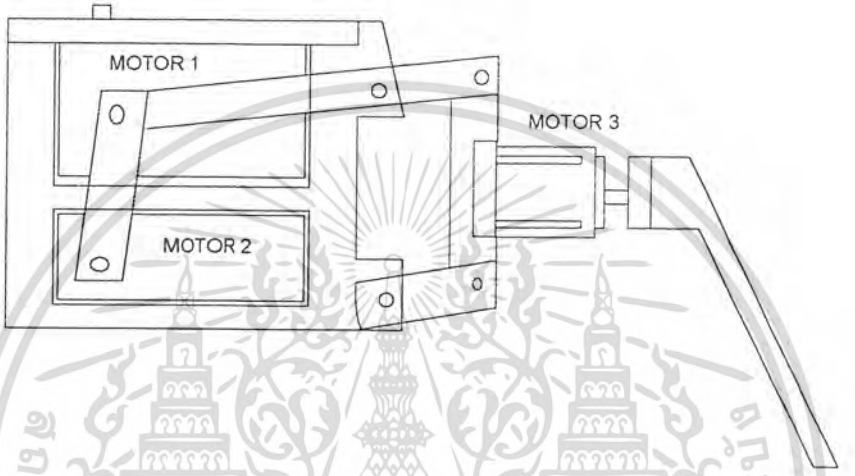


รูปที่ 4.8 การว่ายเลี้ยวซ้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

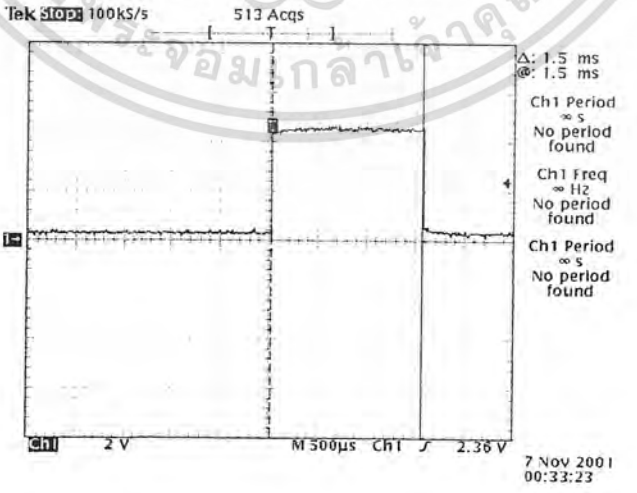
4.4 รูปสัญญาณพัลส์ควบคุมเซอร์โวมอเตอร์

เพื่อให้เข้าใจการจ่ายพัลส์เพื่อควบคุมให้หุ่นสามารถเคลื่อนไหวได้ โดยในการเดินนั้นจะใช้เซอร์โวมอเตอร์ตัวที่ 1 และตัวที่ 2 และในการว่ายน้ำนั้นจะใช้เซอร์โวมอเตอร์ตัวที่ 1 และตัวที่ 3



รูปที่ 4.9 แสดงรูปขาที่ประกอบเสร็จแล้ว

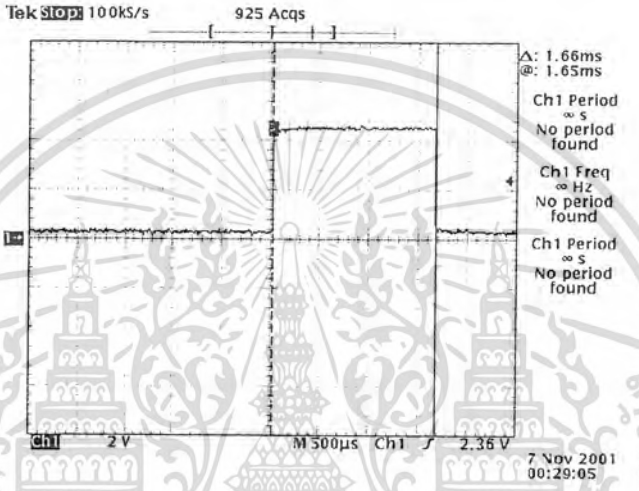
เมื่อเริ่มต้นการทำงานของหุ่นจะจ่ายพัลส์ขนาด 1.5 mS เพื่อให้แกนของเซอร์โวมอเตอร์ทุกตัว อยู่ที่ตำแหน่งจุดศูนย์กลางของมอเตอร์คือ 90 องศา



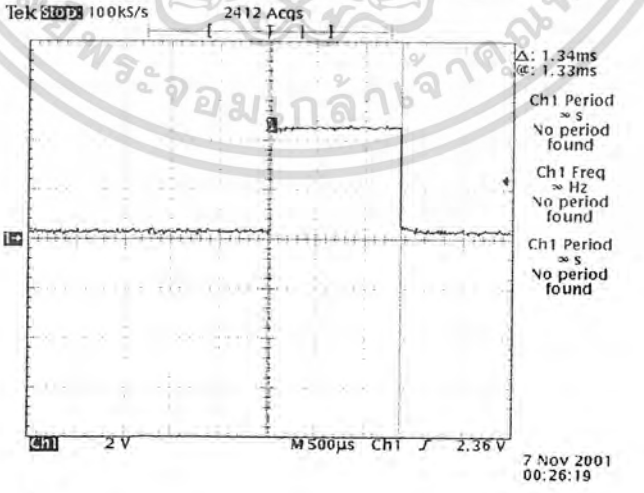
รูปที่ 4.10 รูปสัญญาณเมื่อหุ่นยนต์ยืนอยู่กับที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อต้องการให้หุ่นเดินไปข้างหน้า จะต้องจ่ายพัลส์ที่มีความกว้าง 1.66mS และ 1.34 mS ให้กับเซอร์โวมอเตอร์ตัวที่ 1 เพื่อให้เซอร์โวมอเตอร์หมุนไปที่มุม 120 และ 60 องศาได้ตามลำดับ และจะต้องจ่ายพัลส์ที่มีความกว้าง 1.55 mS และ 1.45 mS ให้กับเซอร์โวมอเตอร์ตัวที่ 2 เพื่อให้เซอร์โวมอเตอร์หมุนไปที่มุม 100 และ 80 องศาได้ตามลำดับ

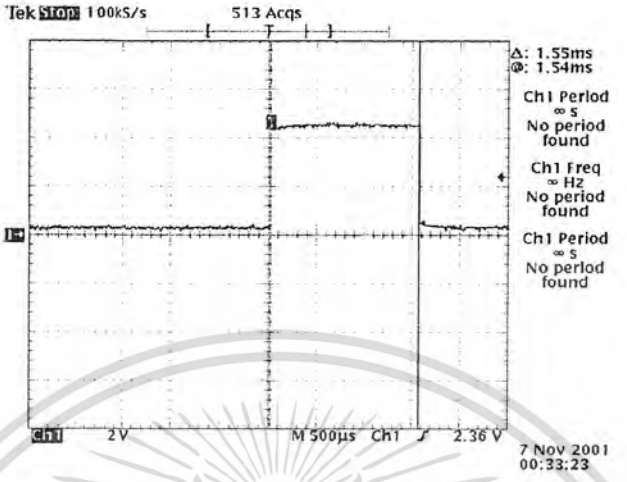


ลำดับที่ 1 จ่ายพัลส์เพื่อให้เซอร์โวมอเตอร์หมุน 120 องศา

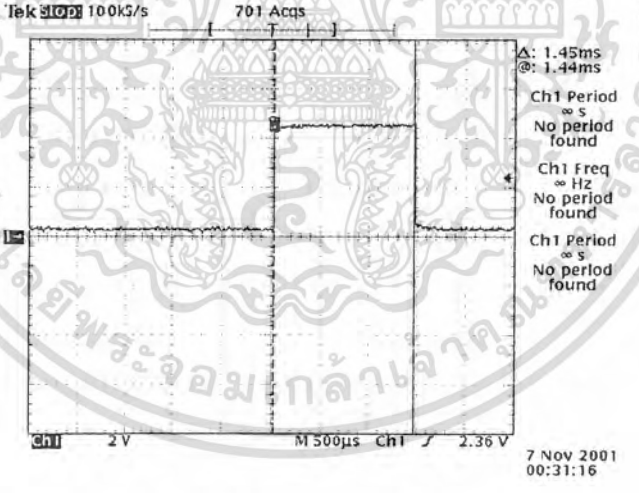


ลำดับที่ 2 จ่ายพัลส์เพื่อให้เซอร์โมอเตอร์หมุน 60 องศา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ลำดับที่ 3 จ่ายพัลส์เพื่อให้เซอร์โมอเตอร์หมุน 100 องศา

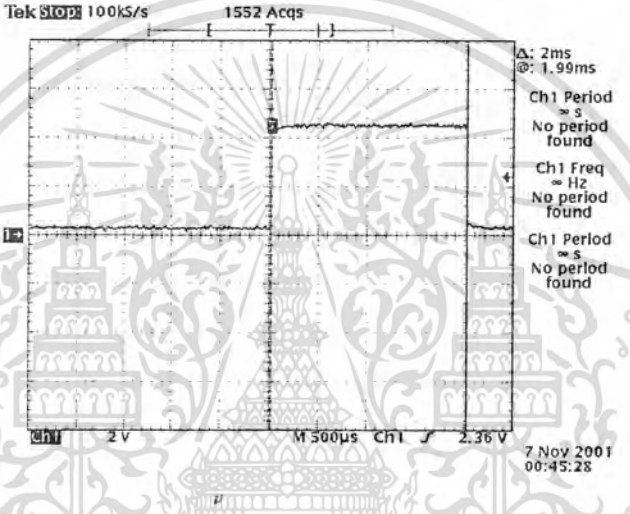


ลำดับที่ 4 จ่ายพัลส์เพื่อให้เซอร์โมอเตอร์หมุน 80 องศา

รูปที่ 4.11 รูปสัญญาณเมื่อหุ่นยนต์เดินหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อต้องการให้หุ่นเคลื่อนไหวในน้ำได้ ซึ่งการทำงานในตอนว่ายน้ำนี้จะใช้เซอร์โวมอเตอร์ตัวที่ 1 และ 3 ในการเคลื่อนไหว ซึ่งจะจ่ายพัลส์ให้กับเซอร์โวมอเตอร์ตัวที่ 1 เหมือนกับการเดินและจะจ่ายพัลส์ที่มีขนาดความกว้าง 2 mS ให้กับเซอร์โวมอเตอร์ตัวที่ 3 คือจะหมุนไปที่ตำแหน่ง 180 องศา



รูปที่ 4.12 รูปสัญญาณเมื่อหุ่นยนต์ว่ายน้ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปโครงงาน ปัญหาและข้อเสนอแนะ

5.1 สรุปโครงงาน

หุ่นยนต์เต่าที่สร้างจะเน้นในส่วนของขาหุ่นยนต์มากเนื่องจากเป็นส่วนสำคัญในการที่จะให้หุ่นยนต์เคลื่อนที่และต้องทำการซิลล์(seal) เพื่อกันน้ำในการลงน้ำ ในแต่ละขาของหุ่นยนต์ประกอบด้วย เซอร์โวมอเตอร์สามตัว ตัวที่หนึ่งทำหน้าที่หมุนไปในแนวราบเพื่อให้หุ่นเคลื่อนที่ไปด้านหน้า หมุนไปประมาณ 60 องศา ตัวที่สองทำหน้าที่ยกขาของหุ่นยนต์ขึ้นในแนวตั้งประมาณ 2 เซนติเมตร เพื่อให้หุ่นยนต์สามารถก้าวเดินไปได้ ตัวที่สามทำหน้าที่ หมุนในตอนเคลื่อนที่ในน้ำ หมุนไปประมาณ 90 องศา โดยที่เซอร์โวมอเตอร์ทั้งหมดถูกควบคุมโดยไมโครคอลโทรลเลอร์ ตัวหุ่นยนต์ที่สร้างมีขนาด 20x36x12.5 เซนติเมตร น้ำหนัก 3 กิโลกรัม แต่ละขาเคลื่อนไหวอิสระต่อกัน หุ่นยนต์ขับเคลื่อนโดยแบตเตอรี่ 12 โวลท์ 3 แอมแปร์ ชุดบังคับใช้สัญญาณวิทยุขนาดแรงดันไฟฟ้า 3 โวลท์ 1.5 แอมแปร์

จากการทดสอบหุ่นยนต์เต่าสามารถเดินไปด้านหน้าและสามารถเคลื่อนที่ไปด้านหน้าในน้ำ ทดสอบโดยควบคุมแบบควบคุมโดยมนุษย์

จุดที่ต้องปรับปรุงและแก้ไขคือ ระบบแมคคานิกส์ในส่วนของโครงสร้างของหุ่นยนต์ที่มีน้ำหนักมาก ขาหุ่นยนต์และการควบคุมทิศทางและความเร็วของหุ่นยนต์

5.2 ปัญหาในการทำโครงงาน

5.2.1 ปัญหาในการสร้างตัวหุ่น

- ผู้จัดทำขาดความรู้ด้านแมคคานิกส์ ไม่สามารถแก้ไขทางแมคคานิกส์ได้
- ปัญหาในการซิลล์ ด้วยซิลลิโคน เนื่องจากหุ่นต้องเคลื่อนที่ในน้ำ
- ปัญหาในการในการยืดหยุ่นของ ของซิลิโคล เนื่องจากในการเคลื่อนที่ของหุ่นยนต์เต่าต้องการพื้นที่ในการเคลื่อนที่ของขา
- อุปกรณ์ในการสร้างหุ่นยนต์เต่ามีราคาแพง ต้องใช้เงินทุนที่มากพอสมควร
- การจัดหาและเลือกวัสดุในการทำหุ่นยนต์ใช้เวลานาน

5.2.2 ปัญหาในการควบคุม

- ขณะหุ่นยนต์ทำการเคลื่อนที่หุ่นยนต์ใช้พลังงานจากแบตเตอรี่ค่อนข้างสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก
รหัสโปรแกรมการเคลื่อนที่ของหุ่นยนต์เต่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;       ***************************************************
;       TURTLE      BACK  LEFT1
;       ***************************************************
        ORG         0000H

P_0     EQU         05      ;       0.5 ms
P_45    EQU         12
P_70    EQU         14
P_90    EQU         15
P_110   EQU         16
P_135   EQU         17
P_180   EQU         25      ;       2.5 ms
;       ***************************************************
NP_0    EQU         200-05
NP_45   EQU         200-12
NP_70   EQU         200-14
NP_90   EQU         200-15
NP_110  EQU         200-16
NP_135  EQU         200-17
NP_180  EQU         200-25
;       ***************************************************
T0_B    EQU         030H
T1_B    EQU         031H
T0_L    EQU         033H
T0_H    EQU         034H
T1_L    EQU         035H
T1_H    EQU         036H
;       ***************************************************
        AJMP        MAIN
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                ORG         000BH
                AJMP        T0_INT
                ORG         001BH
                AJMP        T1_INT
                ORG         002BH
FLAG:           DS          1
T1_F            EQU         FLAG.1
T0_F            EQU         FLAG.2
;               ************************************************
;               INTERRUPT   TIMER 0      FOR   - TURN -  SERVO
;               ************************************************
T0_INT:         CLR         TF0
                DJNZ        R0,E_T0
                JNB         T0_F,T0_LOW
                SETB        P2.0
                MOV         R0,T0_H
                CLR         T0_F
                RETI
T0_LOW:         MOV         R0,T0_L
                CPL         P2.0
                SETB        T0_F
E_T0:           RETI
;               ************************************************
;               INTERRUPT   TIMER 1      FOR  - SHIFT -  SERVO
;               ************************************************
T1_INT:         CLR         TF1
                DJNZ        R1,E_T1
                JNB         T1_F,T1_LOW
                SETB        P2.1
                MOV         R1,T1_H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                CLR     T1_F
                RETI

T1_LOW:         MOV     R1,T1_L
                CPL     P2.1
                SETB    T1_F
E_T1:           RETI

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

MAIN:           MOV     DPTR,#0F000H
POR1:           INC     DPTR
                MOV     A,DPH
                CJNE    A,#0,POR1
                MOV     SP,#100
;               ************************************************
                MOV     TMOD,#022H
                MOV     TH1,#-100
                MOV     TH0,#-100
                MOV     TL1,#-100
                MOV     TL0,#-100
                SETB    TR1                     ;
                MOV     FLAG,#1
                MOV     R0,#P_90
                MOV     R1,#P_90
                MOV     T0_H,#P_90      ;       START OF  90 DEGREE
                SETB    TR0
                MOV     T0_L,#NP_90             ;
                MOV     T1_H,#P_90              ;
                MOV     T1_L,#NP_90             ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;       ***************************************************
        SETB    EA
        SETB    ET0
        SETB    ET1
        ACALL   DEL
        CLR     TR0
        CLR     TR1
;       ***************************************************
;   *               INPUT                       *
;       ***************************************************

        MOV     P3,#0FFH
JLL:    MOV     A,P3
        CPL     A
        CJNE    A,#00000010B,CK1
        ;WALK RIGHT  LEFT   GO  BACK
        CLR     P1.1
        SETB    TR1
        SETB    TR0
        SJMP    WALK
CK1:    CJNE    A,#00000001B,CENT
        ;SWIMMING   RIGHT  LEFT   GO  BACK
        CLR     P1.0
        CLR     TR1
        SETB    TR0
        LJMP    SWIMMING
        SJMP    JLL
CENT:   LCALL   CENTER
        CLR     TR0
        CLR     TR1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                SJMP        JLL
WALK:
                MOV         T1_H,#P_45      ;SHIFT
                MOV         T1_L,#NP_45
                ACALL       DEL
                MOV         T0_H,#P_70      ;TURN
                MOV         T0_L,#NP_70     ;       VALUE TO OFF
                ACALL       DEL
                ACALL       DEL
                MOV         T1_H,#P_90
                MOV         T1_L,#NP_90
                ACALL       DEL
                ACALL       DEL
                MOV         T0_H,#P_110
                MOV         T0_L,#NP_110
                ACALL       DEL
                ACALL       DEL
                AJMP        JLL

SWIMMING:       MOV         T0_H,#P_70          ;TURN
                MOV         T0_L,#NP_70     ;       VALUE TO OFF
                ACALL       DEL
                ACALL       DEL
                MOV         T0_H,#P_110
                MOV         T0_L,#NP_110
                ACALL       DEL
                ACALL       DEL
                AJMP        JLL
CENTER:         MOV         TH1,#-100
                MOV         TL1,#-100
```

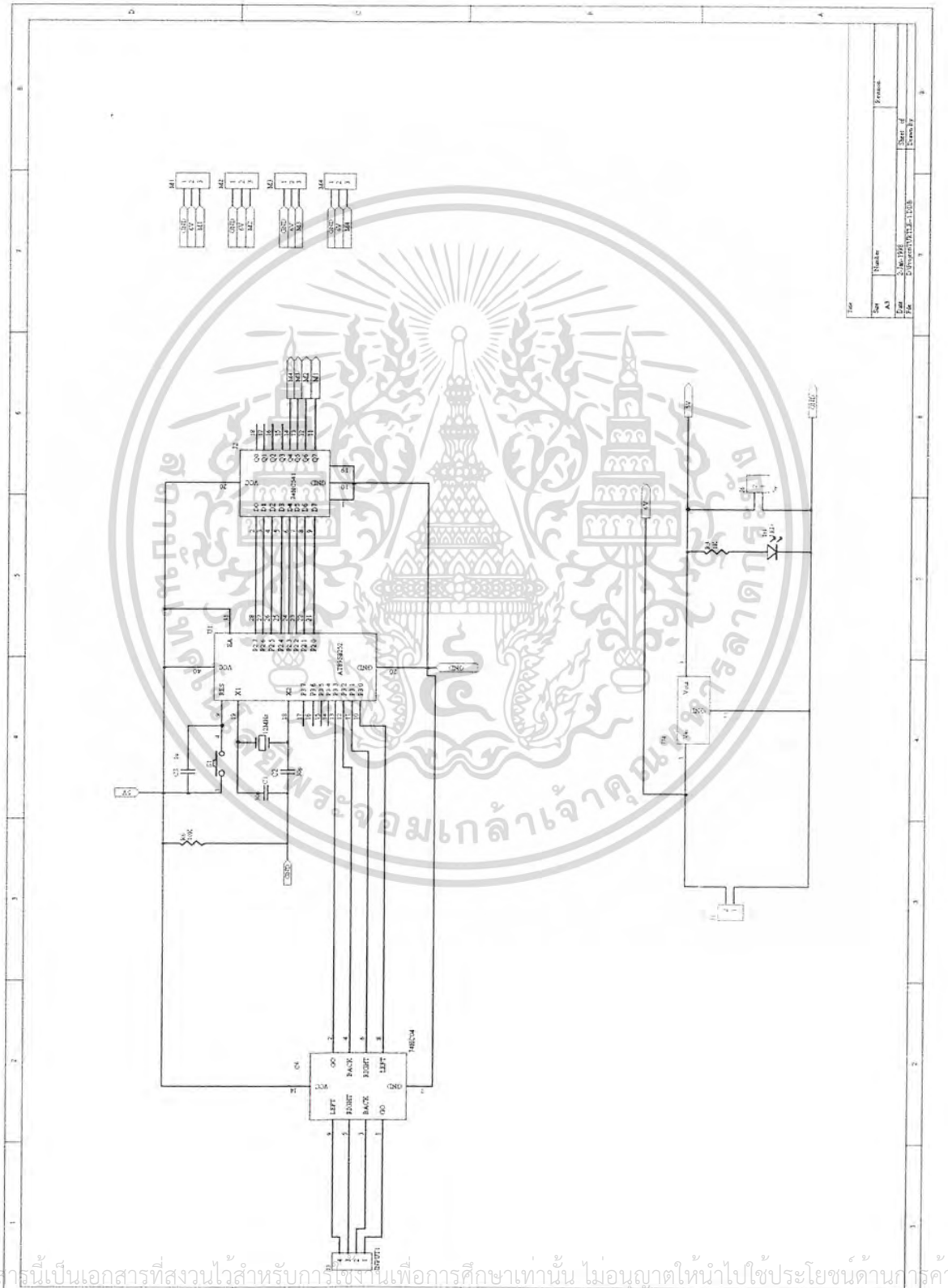
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV     TH0,#-100
                MOV     TL0,#-100
                SETB    TR1
;               INITIAL VALUE
                MOV     R0,#P_90
                MOV     R1,#P_90
                MOV     T0_H,#P_90      ;       VALUE TO ON
                SETB    TR0
                MOV     T0_L,#NP_90     ;       VAULE TO OFF
                MOV     T1_H,#P_90      ;       VAULE TO ON
                MOV     T1_L,#NP_90     ;       VALUE TO OFF
                ACALL   DEL
                RET
DEL:            MOV     R7,#2       ;   TO CYCLE TO1 TURN
DEL1:           MOV     R6,#010
DEL2:           MOV     R5,#020
DEL3:           MOV     R4,#050
                DJNZ    R4,$
                DJNZ    R5,DEL3
                DJNZ    R6,DEL3
                DJNZ    R7,DEL1
                RET
END
```

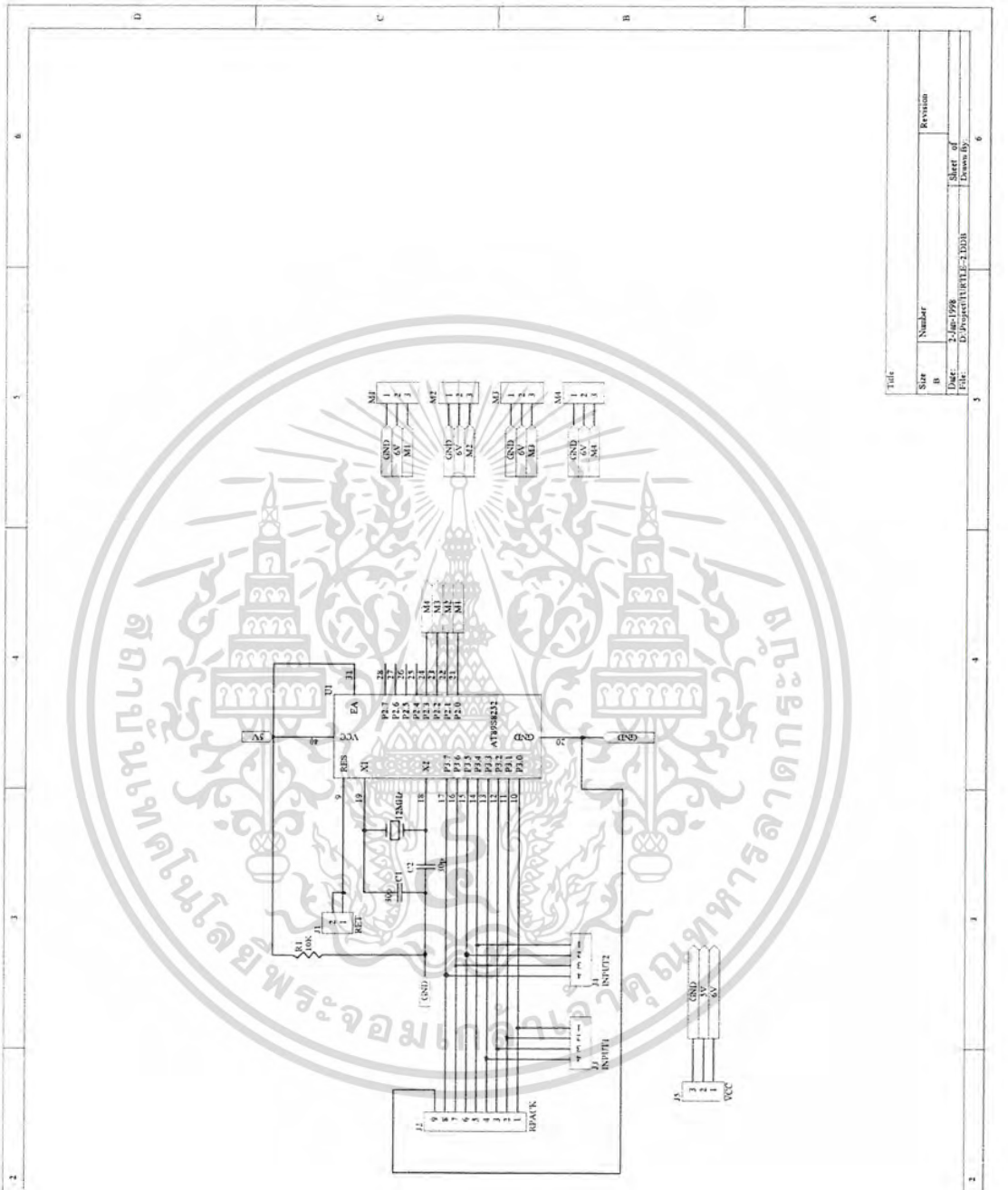
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข
วงจรควบคุมไมโครคอนโทรลเลอร์

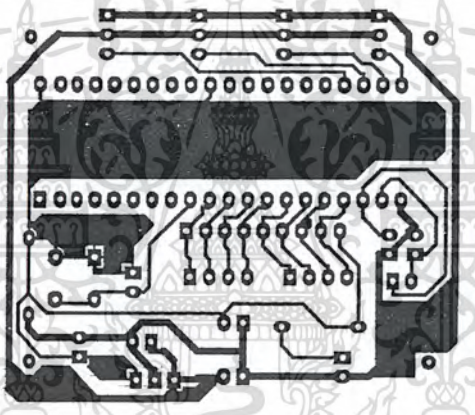
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

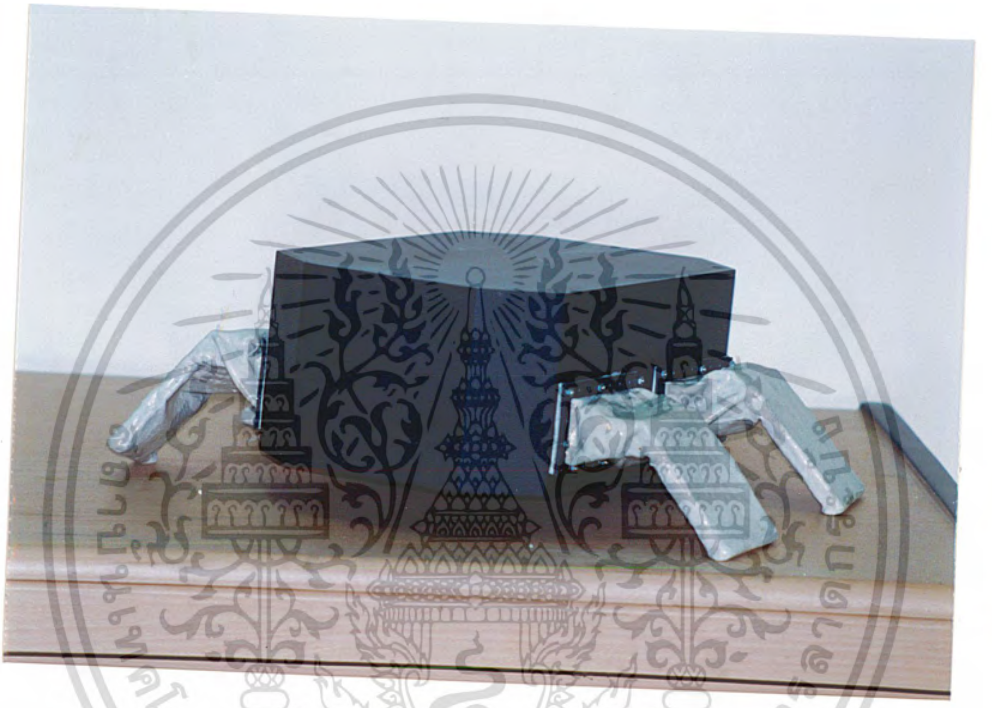


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

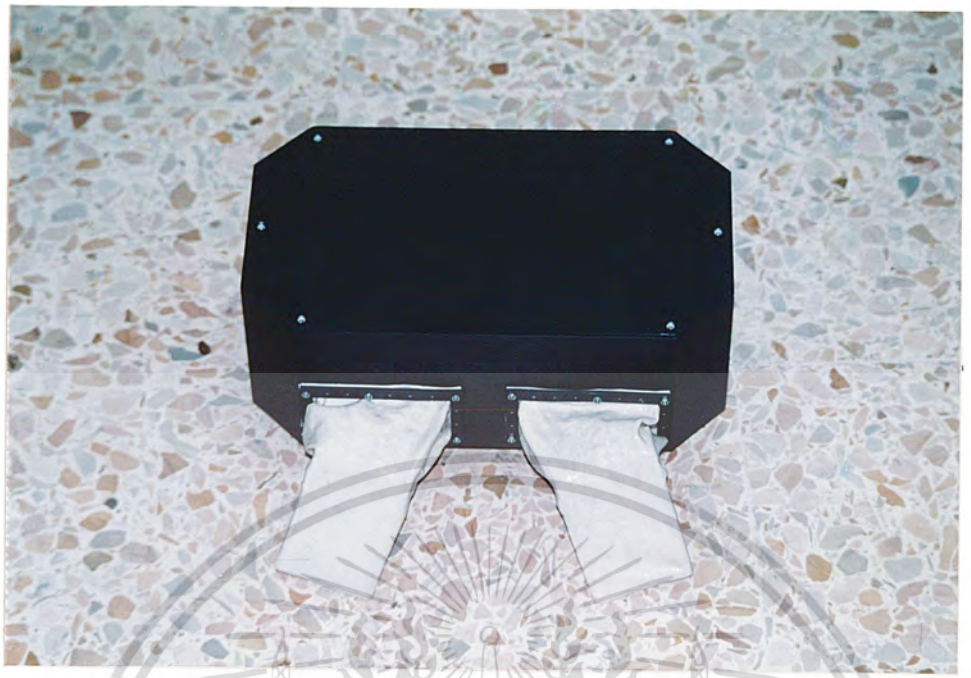
ภาคผนวก ค
รูปภาพตัวหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ (ก) แสดงภาพเต็มตัวของหุ่นยนต์เต่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ (ข) แสดงขณะหุ่นยนต์เต่าเคลื่อนไหวบนบก

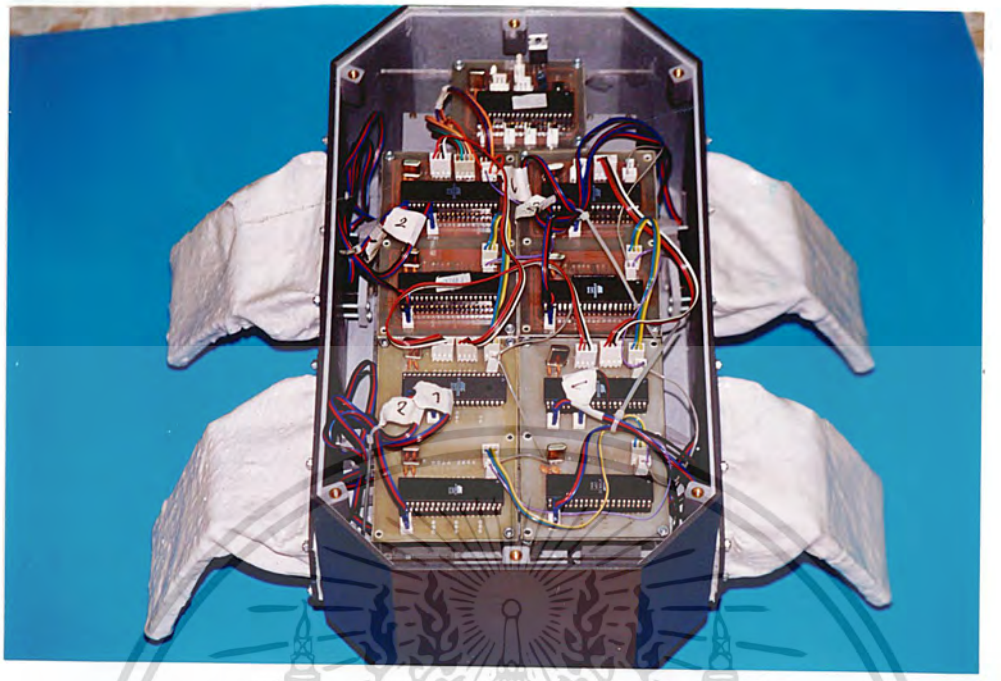
รูปที่ (ค) แสดงขณะหุ่นยนต์เต่าเลี้ยวบนบก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

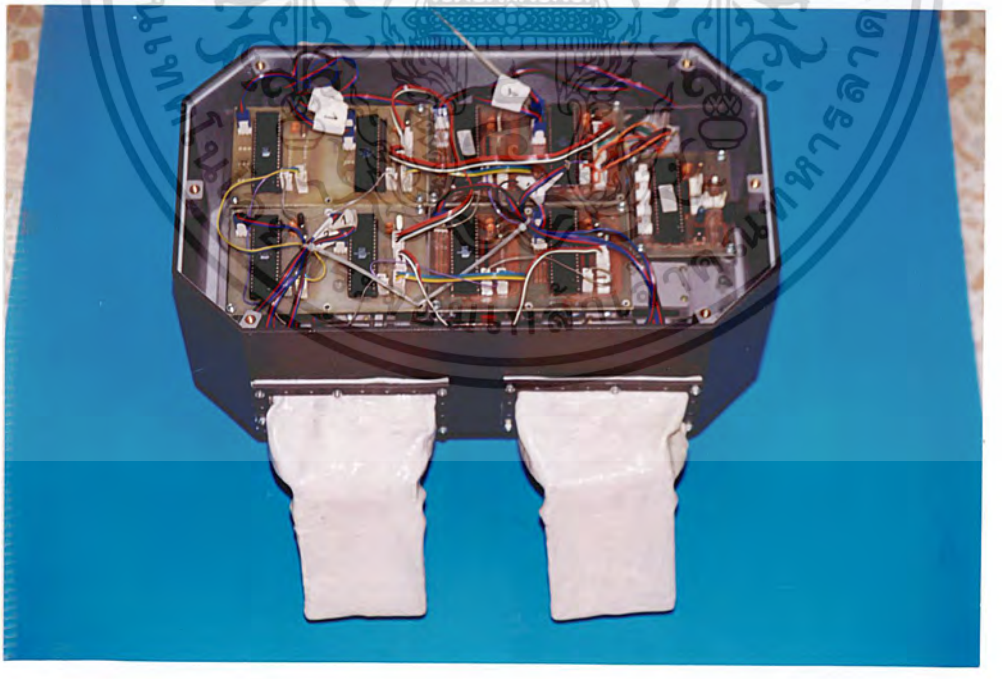
รูปที่ (ง) แสดงขณะหุ่นยนต์เต่าเคลื่อนไหวในน้ำ

รูปที่ (จ) แสดงลักษณะขาของหุ่นยนต์เต่าเมื่ออยู่ในน้ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ (ฉ) การติดตั้งบอร์ดในตัวหุ่นยนต์เต่าเมื่อมองจากด้านหน้า



รูปที่ (ช) การติดตั้งบอร์ดในหุ่นยนต์เต่าเมื่อมองจากด้านข้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง

รายละเอียดของ AT89C52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Features

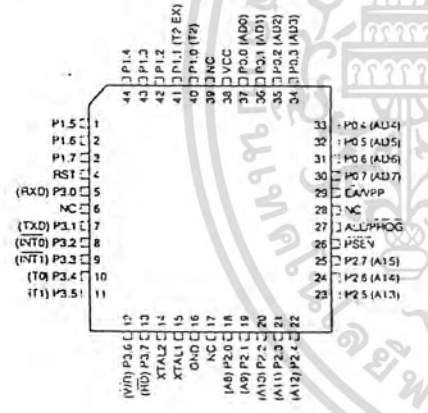
- Compatible with MCS-51™ Products
- 8K Bytes of In-System Reprogrammable Flash Memory
- Endurance: 1,000 Write/Erase Cycles
- Fully Static Operation: 0 Hz to 24 MHz
- Three-level Program Memory Lock
- 256 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Three 16-bit Timer/Counters
- Eight Interrupt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

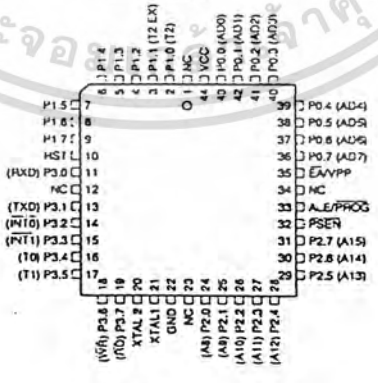
Pin Configurations

PQFP/TQFP



PDIP

PLCC



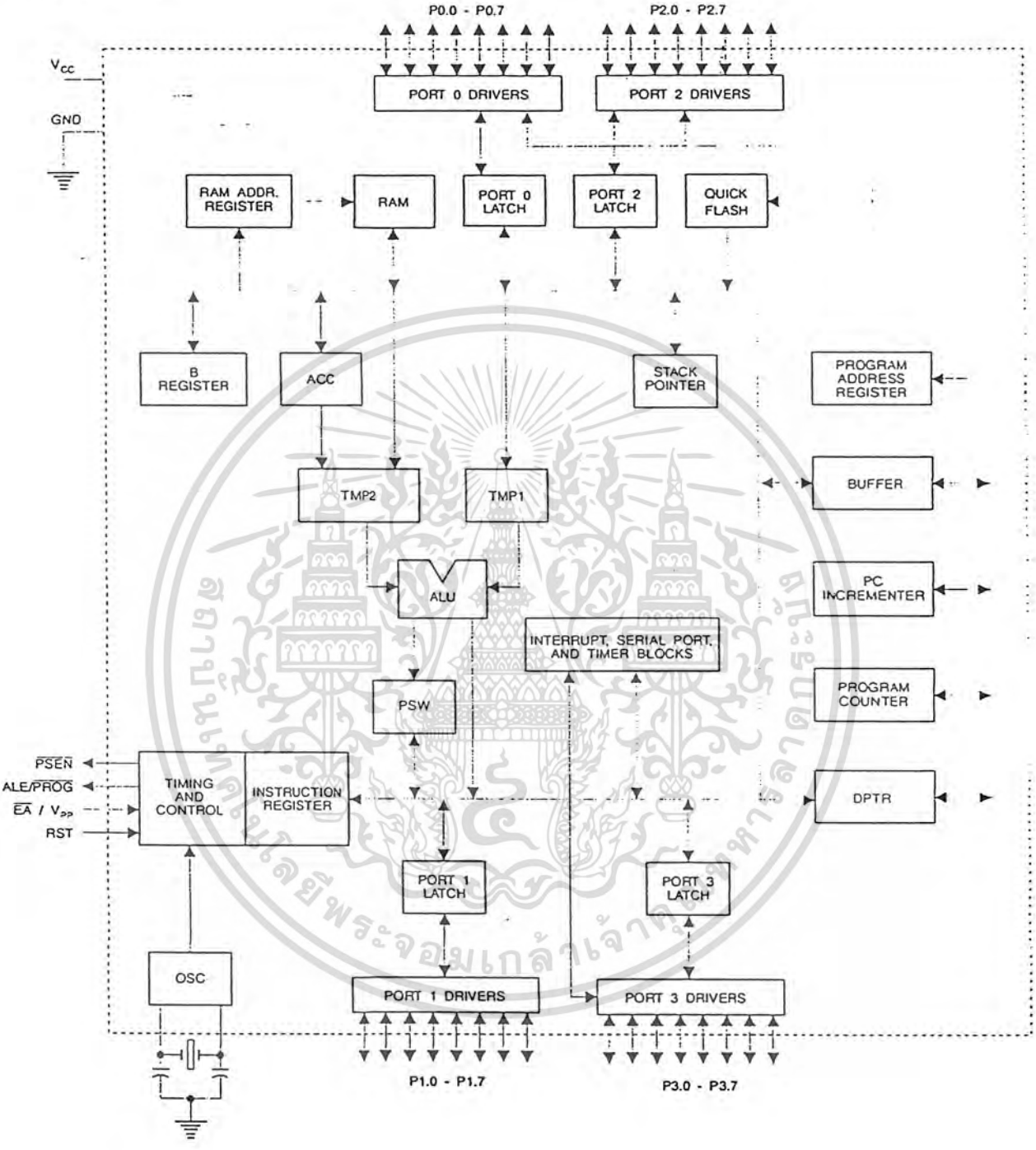
8-bit Microcontroller with 8K Bytes Flash

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.

PSEN

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

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

EA/VPP

External Access Enable. 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, EA will be internally latched on reset.

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

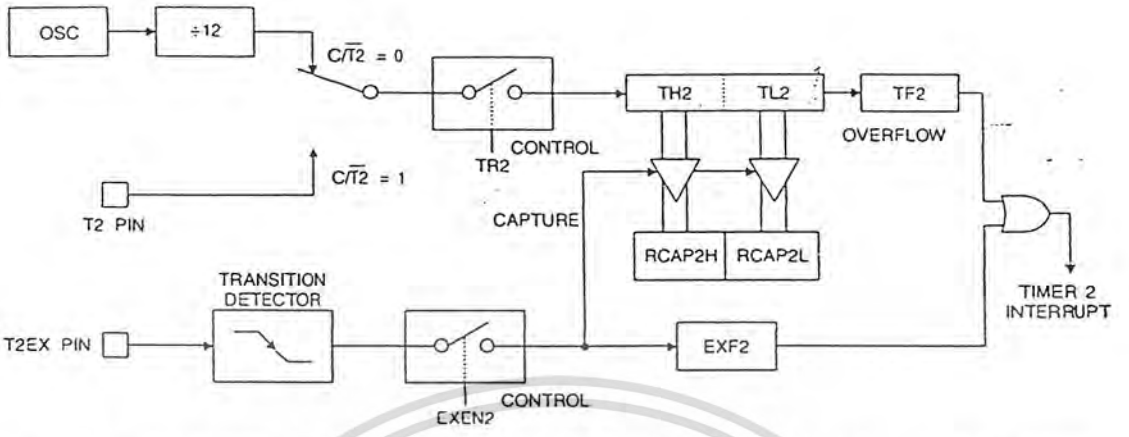


Figure 2 shows Timer 2 automatically counting up when DCEN = 0. In this mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 counts up to 0FFFFH and then sets the TF2 bit upon overflow. The overflow also causes the timer registers to be reloaded with the 16-bit value in RCAP2H 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)

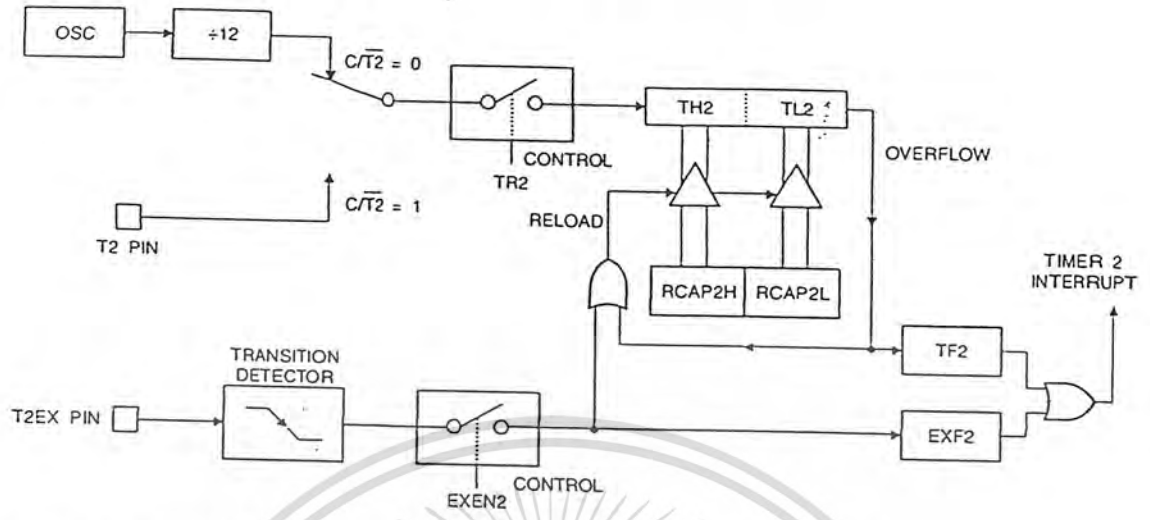


Table 4. T2MOD – Timer 2 Mode Control Register

T2MOD Address = 0C9H

Reset Value = XXXX XX00B

Not Bit Addressable

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

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)

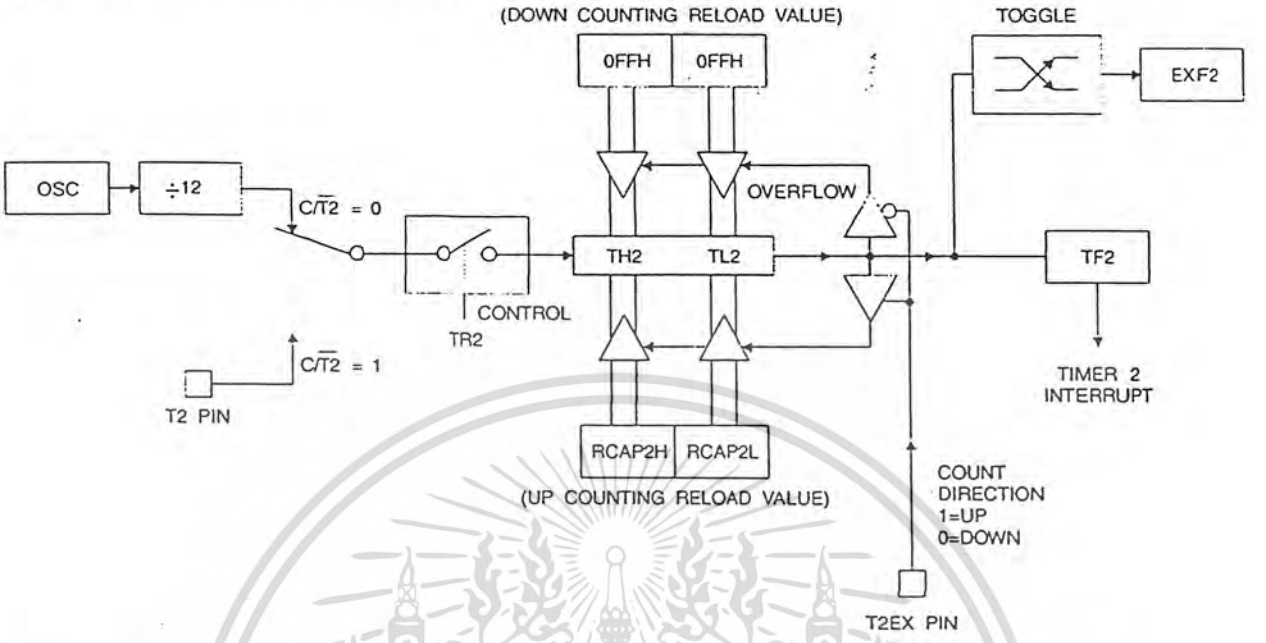
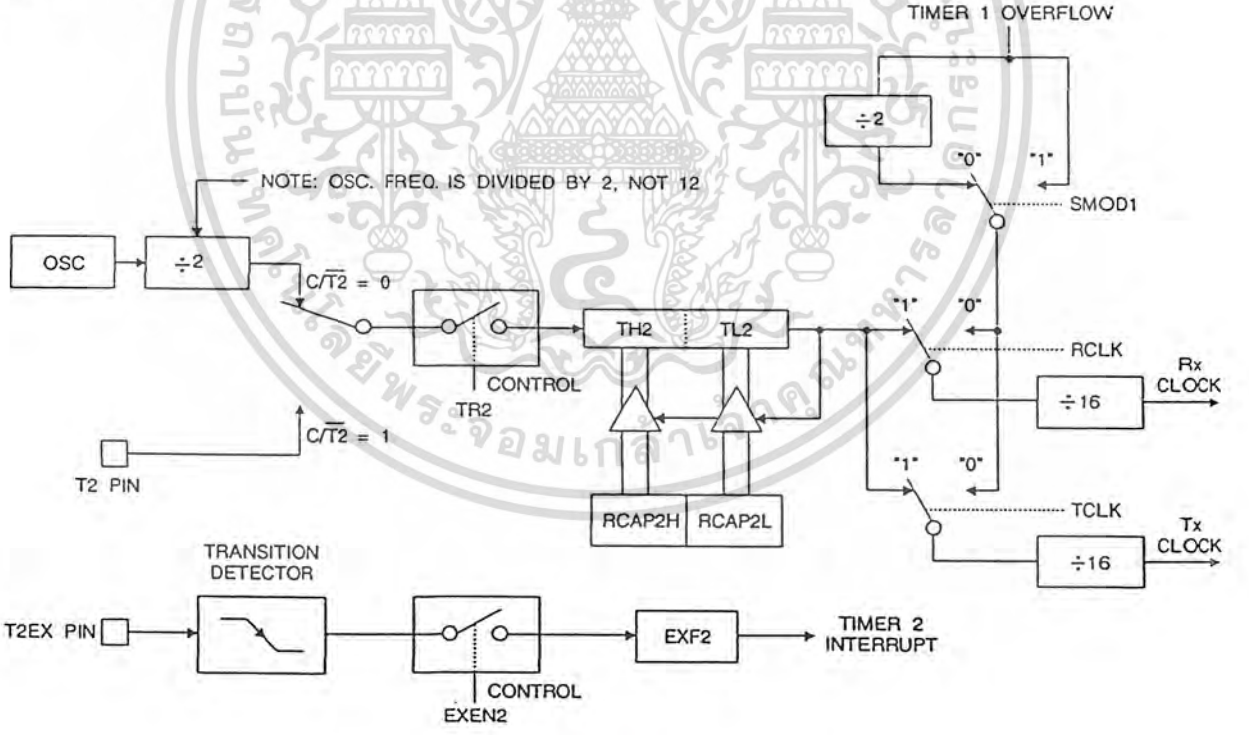


Figure 4. Timer 2 in Baud Rate Generator Mode



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.

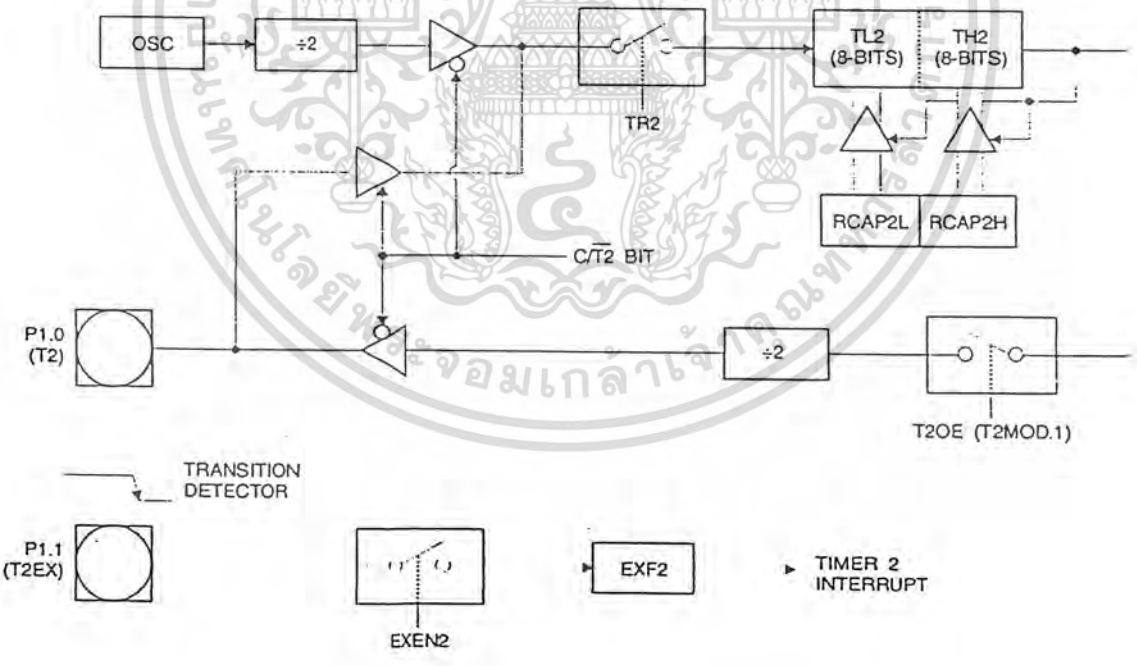
$$\frac{\text{Modes 1 and 3}}{\text{Baud Rate}} = \frac{\text{Oscillator Frequency}}{32 \times [65536 - (\text{RCAP2H,RCAP2L})]}$$

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

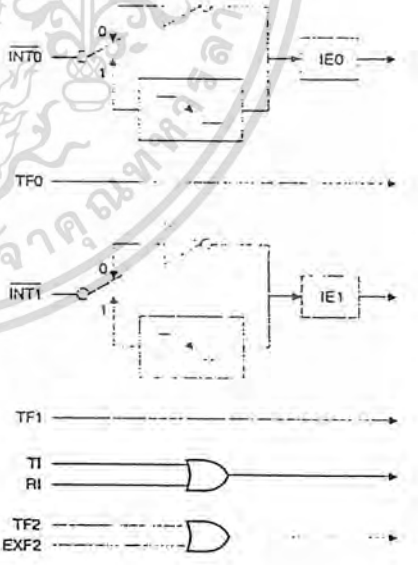
(MSB)							(LSB)
EA	–	ET2	ES	ET1	EX1	ET0	EX0

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

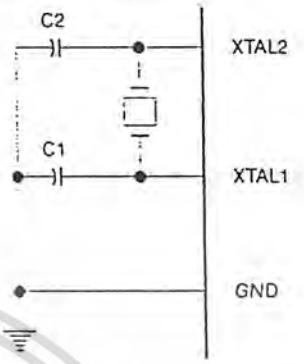
In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware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

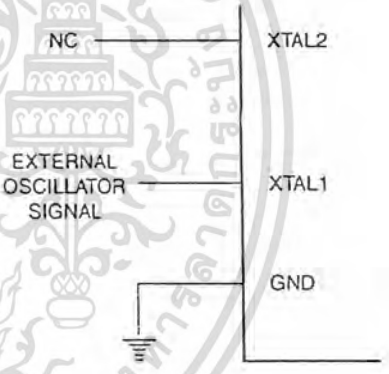
In the power-down mode, the oscillator is stopped, and the instruction that invokes power-down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power-down mode is terminated. The only exit from power-down is a hardware reset. Reset redefines 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/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

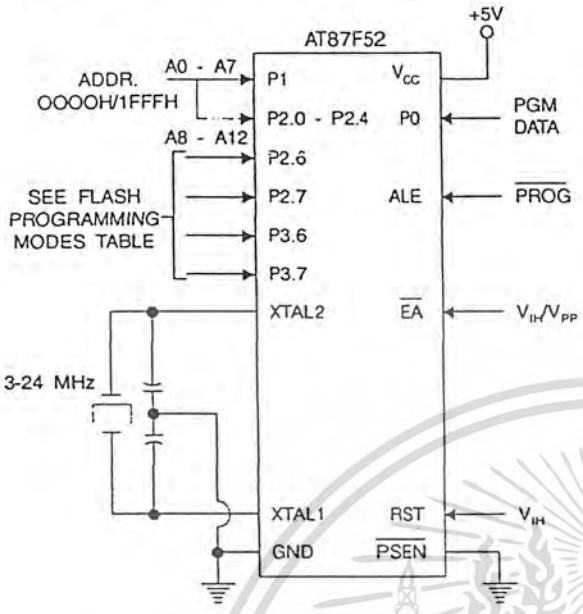
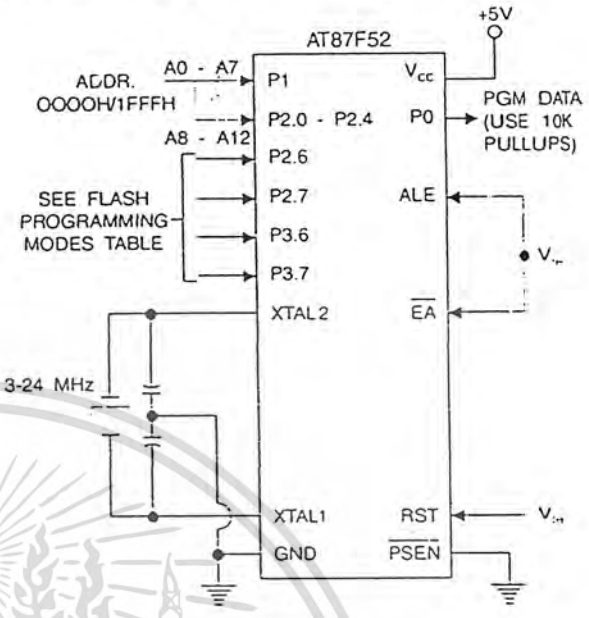


Figure 10. Verifying the Flash Memory



Flash Programming and Verification Characteristics

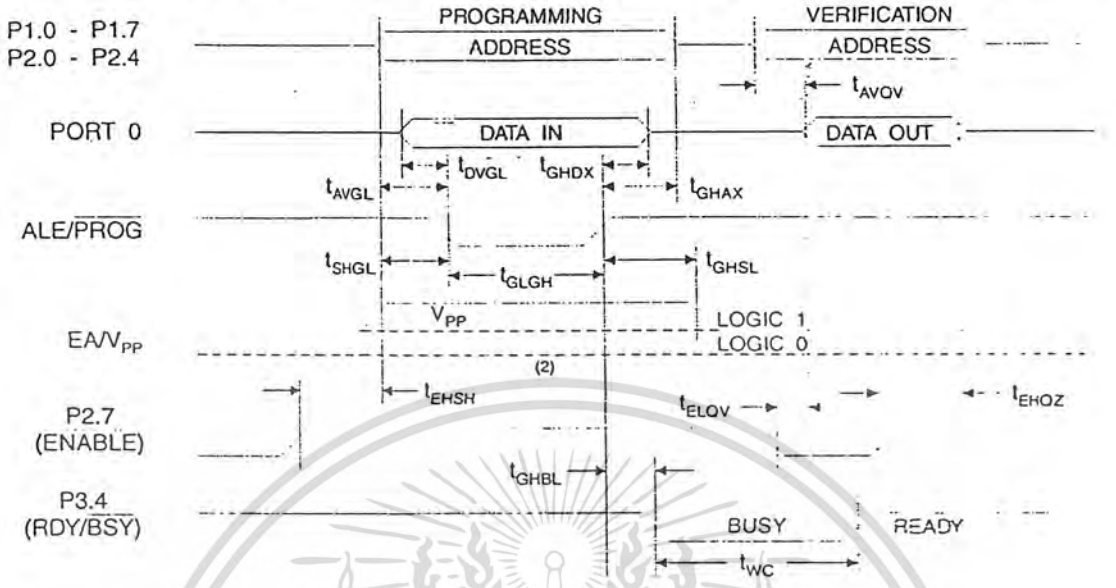
T_A = 0°C to 70°C, V_{CC} = 5.0 ± 10%

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 $\overline{PROG}$ Low	$48t_{CLCL}$		
t_{GHAX}	Address Hold after $\overline{PROG}$	$48t_{CLCL}$		
t_{DVGL}	Data Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
t_{GHDX}	Data Hold After $\overline{PROG}$	$48t_{CLCL}$		
t_{EHSH}	P2.7 (ENABLE) High to V_{PP}	$48t_{CLCL}$		
t_{SHGL}	V_{PP} Setup to $\overline{PROG}$ Low	10		µs
t_{GHSL}[1]	V_{PP} Hold after $\overline{PROG}$	10		µs
t_{GLGH}	$\overline{PROG}$ Width	1	110	µs
t_{AVQV}	Address to Data Valid		$48t_{CLCL}$	
t_{ELQV}	$\overline{ENABLE}$ Low to Data Valid		$48t_{CLCL}$	
t_{EHQZ}	Data Float after $\overline{ENABLE}$	0	$48t_{CLCL}$	
t_{GHBL}	$\overline{PROG}$ High to $\overline{BUSY}$ Low		1.0	µs
t_{WC}	Byte Write Cycle Time		2.0	ms

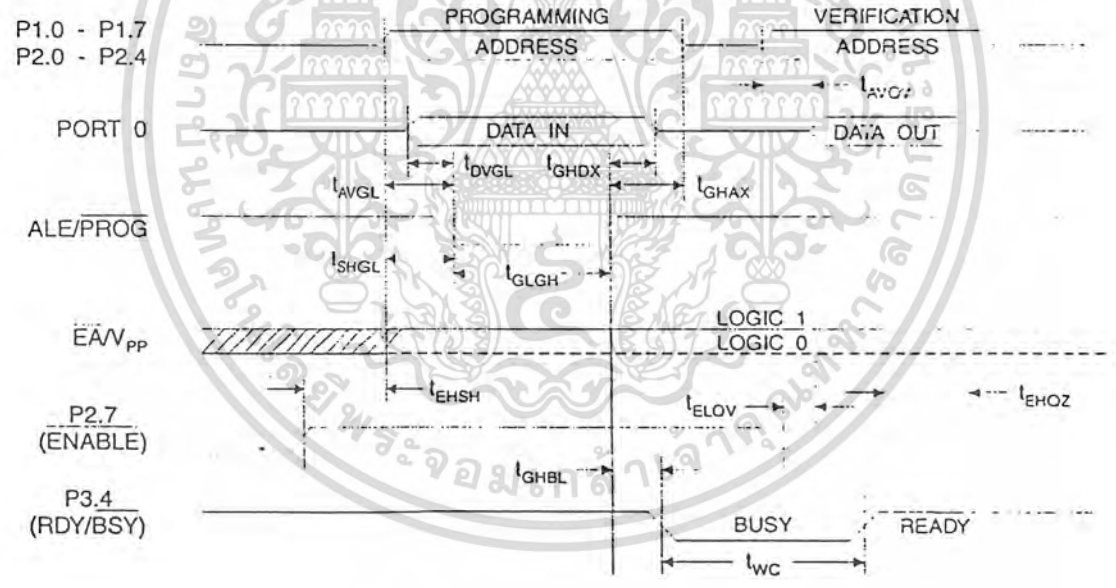
Note: 1. Only used in 12-volt programming mode.

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 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

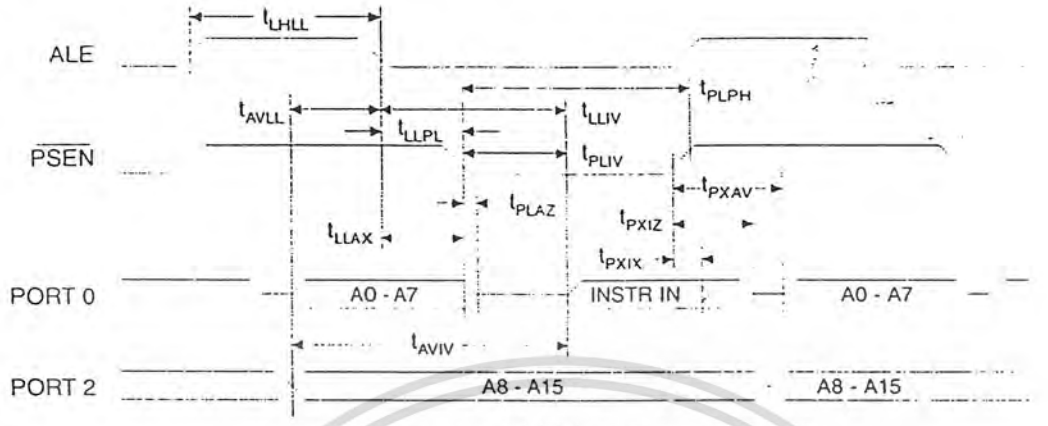
Under operating conditions, load capacitance for Port 0, ALE/$\overline{PROG}$, and $\overline{PSEN}$ = 100 pF; load capacitance for all other outputs = 80 pF.

External Program and Data Memory Characteristics

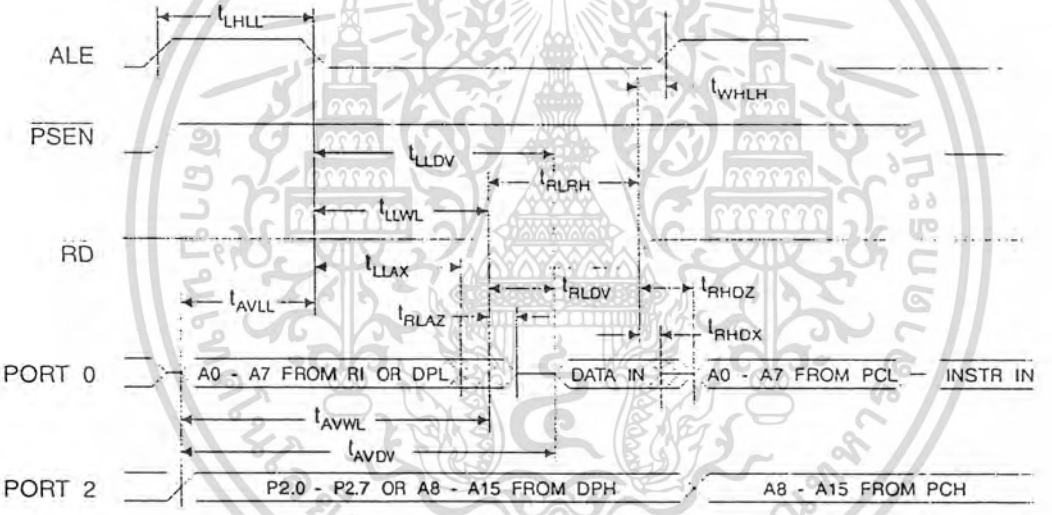
Symbol	Parameter	12 MHz Oscillator		Variable Oscillator		Units
		Min	Max	Min	Max	
$1/t_{CLCL}$	Oscillator Frequency			0	24	MHz
t_{LHLL}	ALE Pulse Width	127		$2t_{CLCL}$-40		ns
t_{AVLL}	Address Valid to ALE Low	43		t_{CLCL}-13		ns
t_{LLAX}	Address Hold After ALE Low	48		t_{CLCL}-20		ns
t_{LLIV}	ALE Low 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

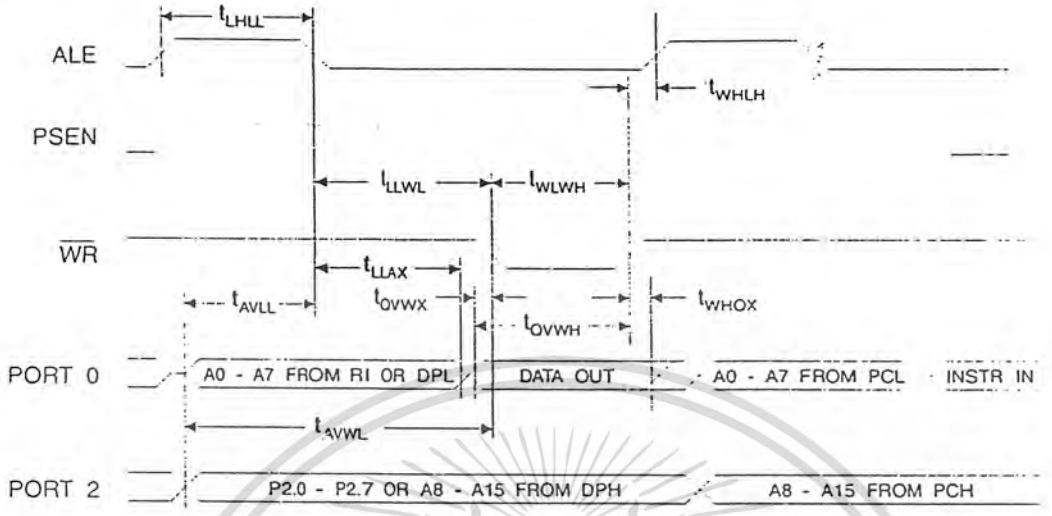


External Data Memory Read Cycle

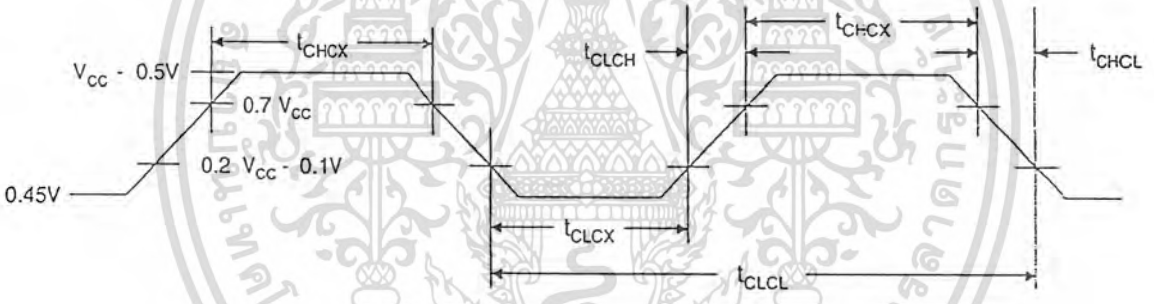


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

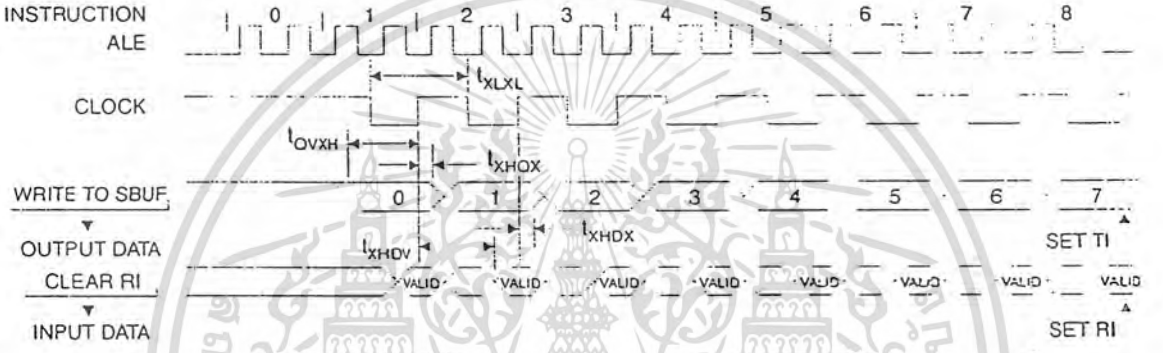
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Serial Port Timing: Shift Register Mode Test Conditions

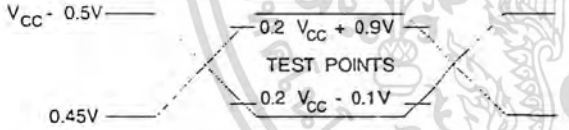
The values in this table are valid for V_{CC} = 5.0V ± 20% and Load Capacitance = 80 pF.

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

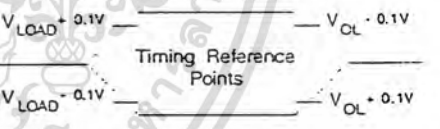


AC Testing Input/Output Waveforms[1]



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

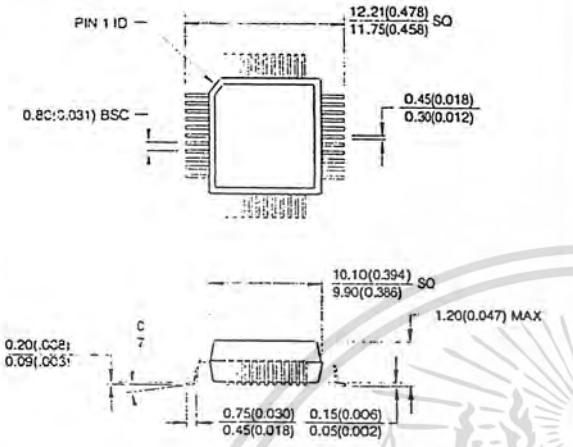
Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
12	5V ± 20%	AT89C52-12AC	44A	Commercial (0°C to 70°C)
		AT89C52-12JC	44J	
		AT89C52-12PC	40P6	
		AT89C52-12QC	44Q	
		AT89C52-12AI	44A	Industrial (-40°C to 85°C)
		AT89C52-12JI	44J	
		AT89C52-12PI	40P6	
		AT89C52-12QI	44Q	
16	5V ± 20%	AT89C52-16AC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

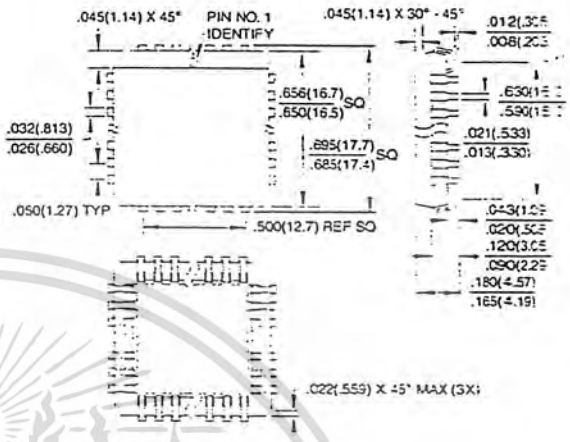
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

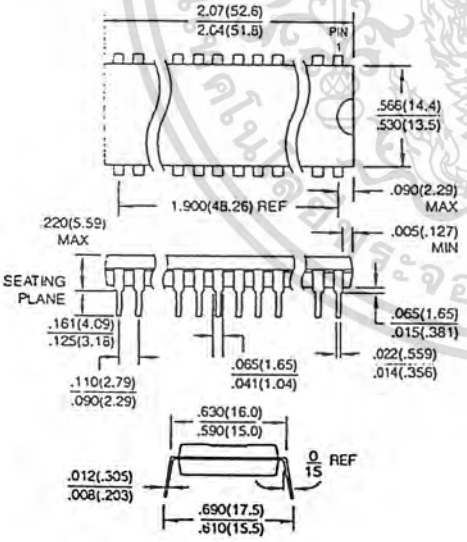


44J, 44-lead, Plastic J-leaded Chip Carrier (PLCC)
Dimensions in Inches and (Millimeters)
JEDEC STANDARD MS-018 AC

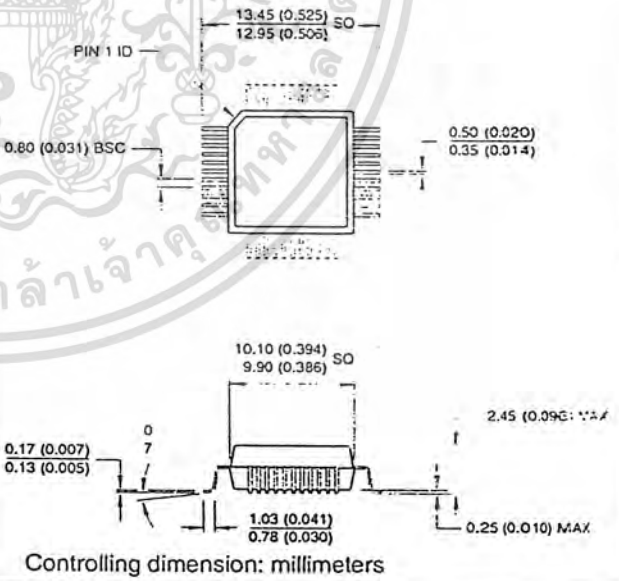


Controlling dimension: millimeters

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

บรรณานุกรม

J.billingsley, "Robot and automated manufacture" , Peter Peregrinus Ltd,1990

L.Jones , "Mobile Robots Inspiration to Implenentation" , A K Peter Wellesley Massachusetts, 1993

ไกรวุฒิ โรจน์ประเสริฐสุด , "ไมโครโปรเซสเซอร์ 2" , บริษัทซีเอ็ด ยูเคชั่น 2539

"ไมโครคอนโทรลเลอร์ AT89C52" , www.pantip.com

"เทคนิคการควบคุมมอเตอร์",1998,www.panmanee.com

"Robot Kits" , 1998 ,www.lynxmotion.com

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้