สำนักวิทยบริการฯ พระจอมเกล้าลาดกระบัง

[หุ่นยนต์มด]

นำเสนอโดย

นายวิสันต์ แสงดารา

นายสัญญา อภิรติกร

เลขหมู่.....
เลขทะเบียน 42687
วัน, เดือน, ปี 6 มิ.ย. 2545

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตร์บัณฑิต
สาขาวิชาเทคโนโลยีอีเล็กทรอนิกส์ ภาควิชาเทคนิคอุตสาหกรรม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ANT ROBOT

BY

Mr.Wisan Sangdara

Mr.Sanya Apiratikorn

Project Report Submitted in Partial Fulfillment of The equipment

For The Bachelor 's Degree

Department Of Industrial Technology

Faculty of Engineering

Academic Year 2000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	หุ่นยนต์มด [ANT ROBOT]
นำเสนอโดย	นายวิสันต์ แสงดารา
	นายสัญญา อภิรติกร
อาจารย์ที่ปรึกษา	อ .บุณย์ชนะ ภู่ระหงษ์
	รศ.นิกร สุขุตตมตันติ
ภาควิชา	เทคนิคอุตสาหกรรม
ปีการศึกษา	2543

คณะวิศวกรรมศาสตร์สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังอนุมัติให้ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตร์บัณฑิต

คณะกรรมการการสอบปริญญานิพนธ์

---------- อาจารย์ที่ปรึกษา
( )

---------- อาจารย์ที่ปรึกษา
( )

---------- คณะกรรมการ
( )

---------- คณะกรรมการ
( )

---------- คณะกรรมการ
( )

---------- คณะกรรมการ
( )

---------- คณะกรรมการ
( )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	ANT ROBOT
By	Mr.Wisan Sangdara
	Mr.Sanya Apiratikorn
Advisor	Assoc.Prof.Nikorn Sukutomtanti
	Boonchana purahong
Department of	Industrial Technology
Academic year	2000

Accept By The Faculty of Engineering, King Mongkut 's Institute of technology ladkrabang in Partial Fulfillment for The bachilor'sDegree

Project report committee

----------Advisor

( )

----------Advisor

( )

----------Member

( )

----------Member

( )

----------Member

( )

----------Member

( )

----------Member

( )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ANT ROBOTS (หุ่นยนต์มด)

นำเสนอโดย

นายวิสันต์ แสงดารา

นายสัญญา อภิรติกร

อาจารย์ที่ปรึกษา

อ.บุญย์ชนะ ภู่ระหงษ์

รศ.นิกร สุขุตมตันติ

ปีการศึกษา 2543

บทคัดย่อ

หุ่นยนต์มดเป็นการสร้างหุ่นยนต์ที่มีขนาดเล็กกว่าหุ่นยนต์โดยทั่วไปเราใช้ MICROCONTROLLER ในการควบคุมการทำงานทั้งหมด จะมีส่วนประกอบต่าง ๆ อาทิเช่น ส่วนที่ใช้ในการควบคุมการเดิน ควบคุมการเคลื่อนที่ และควบคุมการหนีบ อุปกรณ์ SENSOR และอุปกรณ์อื่น ๆ มีการทำงานร่วมกันเป็นกลุ่ม ตัว SENSOR จะทำหน้าที่ในการตรวจจับสิ่งต่าง ๆ และสั่งไปประมวลผลที่ MICROCONTROLLER เพื่อสั่งงานให้ทำตาม PROGARM ที่เรากำหนดไว้ การทำงานของหุ่นยนต์มดเป็นแบบอัตโนมัติทั้งหมด จึงค่อนข้างจะซับซ้อนในการเขียน PROGARE ที่จะทำให้ HARD WARE และ SOLF WARE ทำงานอย่างสัมพันธ์กันเพราะการเขียน PROGARM เราจำเป็นจะต้องอ้างถึง REGISTOR ภายในแต่อย่างไรก็ตามเรายังสามารถควบคุมได้โดยอาศัยความรู้ทาง HARD WARE และ SOLF WARE ที่ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ANT ROBOT

BY	Mr. Wisan	Sangdara
	Mr. Sanya	Apiratikorn
Advisor	Assoc.Prof.Nikorn	Sukutomtanti
	Boonchana	purahong
Academic year	2000	

Abstract

Making the ant robot is make it small when compear other we use the Microcontroller is control the robot which has got component are control driver, control grip and sensor are component main.

The ant robot can team make from program given and make of automatic control from program but write program busy because we tell to internal register however we should be know right hard ware and solf ware for make robot

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบพระคุณอาจารย์บุญย์ชนะ ภู่ระหงษ์ ที่ให้ความกรุณาเกี่ยวกับคำแนะนำต่าง ๆ ที่เกี่ยวกับผลงานในชิ้นนี้ ตลอดจนจัดหาอุปกรณ์เพื่ออำนวยความสะดวกในการทำงานและขอขอบคุณรศ.นิกร สุขุตมตันติ ที่ให้ความกระจ่าง เกี่ยวกับส่วนของ Hard ware และนาย สุรชาติ ใยถาวรพงษ์ ที่ให้คำแนะนำในเรื่องของการจัดซื้ออุปกรณ์ต่าง ๆ และการใช้งาน Solf ware

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบาญ

	หน้า
บทที่ 1 (introduction)	1
ลักษณะการทำงาน	2
ความหมายของ HARD WARE และ SOLF WARE	4
บทที่ 2 (ส่วนประกอบทาง HARD WARE)	6
บทที่ 3 (รายละเอียดเกี่ยวกับ HARD WARE)	10
รายละเอียดการติดตั้งการใช้งาน SENSOR	12
คุณสมบัติของ LED INFRARED	15
Volt Regulator	17
การควบคุม MOTOR (L293D)	19
ตารางการควบคุม	21
การใช้งานร่วมกับ MOTOR	22
กลไกการหนีบ	23
ลักษณะการทำงาน	24
สมการแสดงความสัมพันธ์	25
การใช้ Limit switch ควบคุม	27
Microcontroller และคุณสมบัติ	28
รายละเอียดการใช้งาน Controller และ XTAL	29
บทที่ 4 รายละเอียดทาง SOLF WARE	30
Flow Chart ลักษณะที่ 1	30
Flow Chart ลักษณะที่ 2	31
Flow Chart ลักษณะที่ 3	32
Flow Chart ลักษณะที่ 4	33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบาญรูป

	หน้า
(1-1) ลักษณะการทำงานของ Ant Robot	2
(1-2) การทำงานแบบช่วยกันทำงาน	3
(1-3) แสดงรูป Ant Robot	5
(2-1) Block Diagram แสดงการทำงานพื้นฐาน	6
(2-2) Block Diagram แสดงการควบคุม	8
(3-1) Block Diagram แสดงการทำงานของ Infrared	11
(3-2) รายละเอียดการติดตั้ง Sensor	12
(3-3) ทิศทางในการส่งลำแสง Infrared	13
(3-4) แสดงการทำงานของชุด Sensor อาหาร	14
(3-5) Sensor ที่ใช้ในการตรวจจับสิ่งขีดขวาง	16
(3-6) การต่อใช้งาน Sensor	17
(3-7) Regulator	18
(3-8) Block Diagram การต่อใช้งาน L293D	19
(3-9) โครงสร้างภายในของ L293D	20
(3-10) ตารางแสดงการควบคุม L293D	21
(3-11) ขาต่าง ๆ ในการใช้งานกับ MOTOR	22
(3-12) การต่อใช้งาน L293D	22
(3-13) แสดงการติดตั้งชุดกลไกการหนีบ	23
(3-14) แสดงการพิจารณามุมและระยะทาง	24
(3-15) การใช้ Limit Switch ควบคุม	27
(3-16) การต่อใช้งาน XTAL	29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ (Introduction)

ในบทแรกนี้จะขอกล่าวถึงคุณประโยชน์ต่าง ๆ รูปแบบในการทำงาน การประยุกต์ใช้งานในงานอุตสาหกรรมที่ต่อไปในอนาคต Project ชิ้นนี้อาจจะมีบทบาทในงานอุตสาหกรรมต่อไป

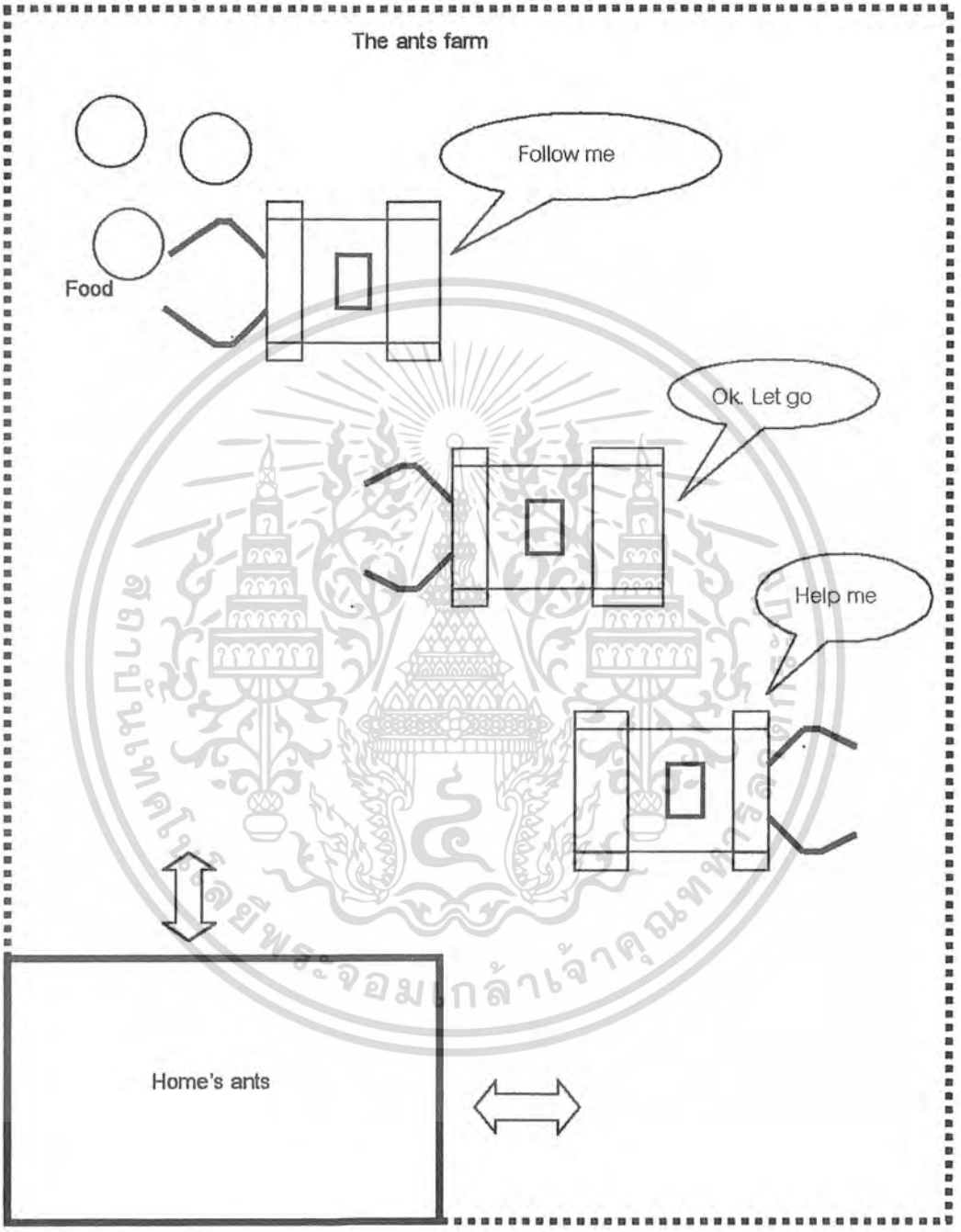
หุ่นยนต์มด เป็นการออกแบบและสร้างหุ่นยนต์ที่มีขนาดเล็ก โดยทำการศึกษาและสังเกตพฤติกรรมของมดมาทำการสร้างเป็นหุ่นยนต์ โดยเล็งเห็นว่าอุตสาหกรรมในอนาคตอาจจะมีขนาดเล็กลง เราสามารถนำหุ่นยนต์มดไปทำการขนถ่ายวัสดุที่มีขนาดเล็ก ซึ่งจะช่วยให้ประหยัดแรงงานมนุษย์ และช่วยประหยัดเวลาในการทำงาน ซึ่งในปัจจุบันสังเกตว่าในอุตสาหกรรมการผลิตต่าง ๆ จะมีหุ่นยนต์เข้าไปมีบทบาทในสายงานการผลิต ทำให้ลดข้อผิดพลาดต่าง ๆ แต่ในการขนถ่ายวัตถุดิบยังคงใช้แรงงานคน ในการขนถ่าย ดังนั้น ทางผู้จัดทำจึงได้ทำการสร้างหุ่นยนต์มด เพื่อที่ในอนาคตอาจจะสามารถนำไปประยุกต์ใช้ในการขนถ่ายวัตถุดิบ เช่น ในอุตสาหกรรม Semiconductor ที่มีอุปกรณ์ตัวเล็ก ๆ เราก็สามารถใช้หุ่นยนต์มดทำการขนถ่ายวัสดุชิ้นเล็กๆแทนแรงงานคนได้

ลักษณะการทำงาน

ลักษณะการทำงานของหุ่นยนต์มด จะมีการทำงานรวมกันเป็นกลุ่มโดยหุ่นยนต์แต่ละตัว จะสามารถที่จะติดต่อถึงกันได้โดยจะมีอุปกรณ์ Sensor ทำหน้าที่ในการส่งสัญญาณถึงตัวอื่น ๆ ให้ทราบว่า ขณะนี้ตนอยู่ที่ใดและให้มาเก็บอาหารได้ที่นี่ ในกรณีที่ตัวอื่นยังไม่พบอาหาร โดยจะมีอุปกรณ์ Sensor ในส่วนหน้าทำหน้าที่ในการตรวจสอบอาหารที่จะเก็บ จากนั้นจะมีส่วนของกลไกที่ใช้ในการหนีบอาหาร ไปเก็บไว้ที่รังซึ่งที่รังจะมีตัว Sensor ที่ทำหน้าที่บอกให้ทราบว่าถึงบริเวณที่ต้องปล่อยอาหารแล้วและส่งสัญญาณไปบอกให้ทราบว่ารังอยู่ที่ใด และเมื่อปล่อยอาหารเสร็จแล้วจากนั้นมันจะกลับไปเก็บอาหารมาใหม่ จะเห็นได้ว่าจากการทำงานเหล่านี้ เราสามารถที่จะนำ Project ชิ้นนี้ไปทำการประยุกต์ใช้ในงานอุตสาหกรรมใหญ่ๆ โดยใช้ขนถ่ายวัตถุดิบการทำงานของมันก็ไม่ยุ่งยากเพราะมันมีการทำงานแบบอัตโนมัติไม่ต้องคอยควบคุมดูแล ในส่วนของการสร้างและการออกแบบแบ่งได้เป็น 2 ส่วนใหญ่ ๆ นั่นก็คือในส่วนของ ฮาร์ดแวร์ และซอฟต์แวร์ ซึ่งจัดได้ว่าเป็นส่วนที่สำคัญจะขาดเสียไม่ได้ในการสร้างหุ่นยนต์ หรืออาจจะกล่าวได้ว่าหุ่นยนต์ทุก ๆ ประเภท ต้องมี 2 ส่วนนี้เป็นส่วนสำคัญซึ่งจะขอกล่าวถึงความหมายดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

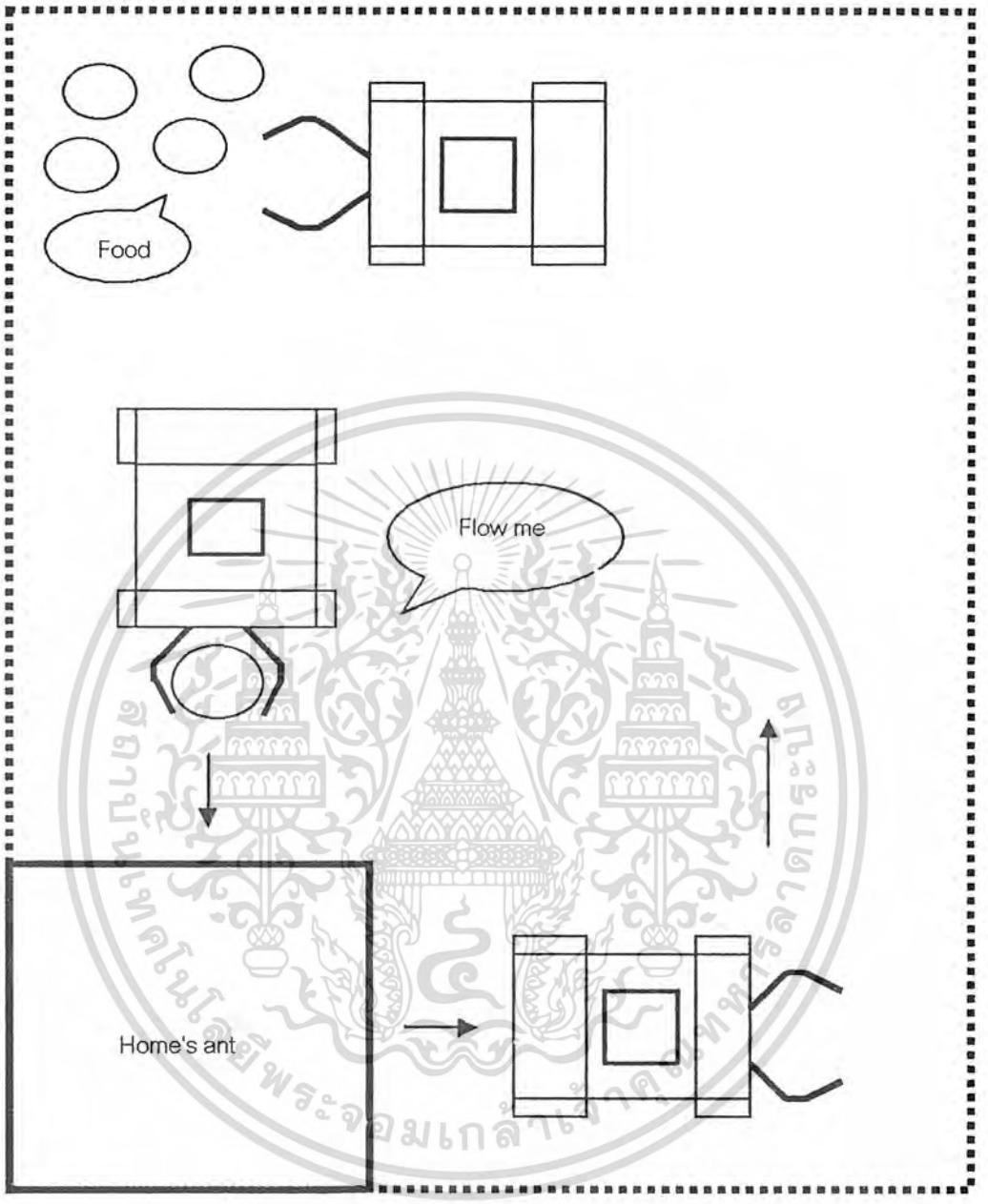
ลักษณะการทำงานของ Ant robot



TOP VIEW

(1-1) ภาพแสดงการทำงานของหุ่นยนต์มด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(1-2) แสดงการทำงานแบบช่วยกันทำงานของหุ่นยนต์มด

เราจะสามารถสังเกตได้ว่า หุ่นยนต์มดมีการทำงานแบบช่วยกันทำงาน จากรูปที่ (1-1) และ (1-2) เมื่อหุ่นยนต์ตัวที่นำอาหารมาเก็บ เสร็จเรียบร้อยแล้วหุ่นยนต์ตัวต่อไปที่กำลังเก็บอาหารอยู่ก็จะนำอาหารที่เก็บได้ไปเก็บไว้ที่รังแทน ส่วนหุ่นยนต์ตัวที่ทำภารกิจเสร็จแล้วก็จะไปเก็บอาหารมาใหม่โดยสามารถที่จะทราบตำแหน่งได้จากหุ่นยนต์ตัวที่กำลังเก็บอาหารอยู่ก็จะส่งสัญญาณไปบอกให้ทราบถึงบริเวณที่จะต้องมาเก็บอาหารให้มาเก็บอาหารต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

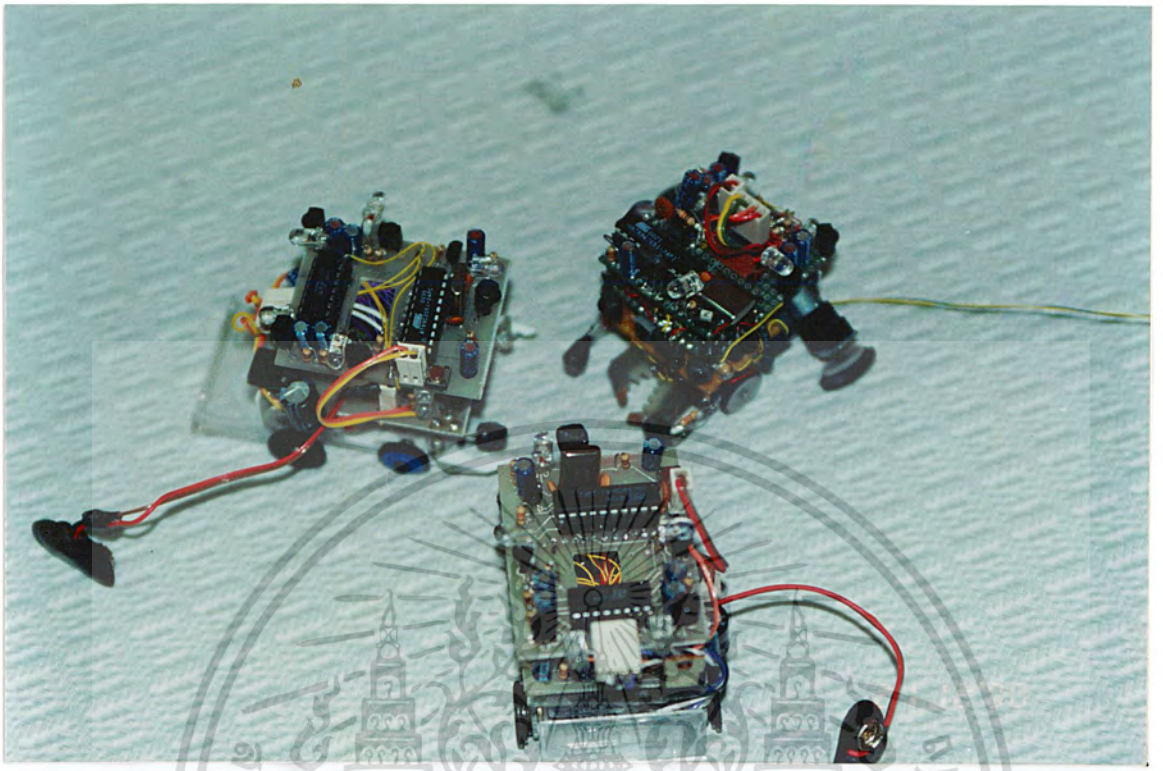
ความหมายของ Hardware และ Solfware

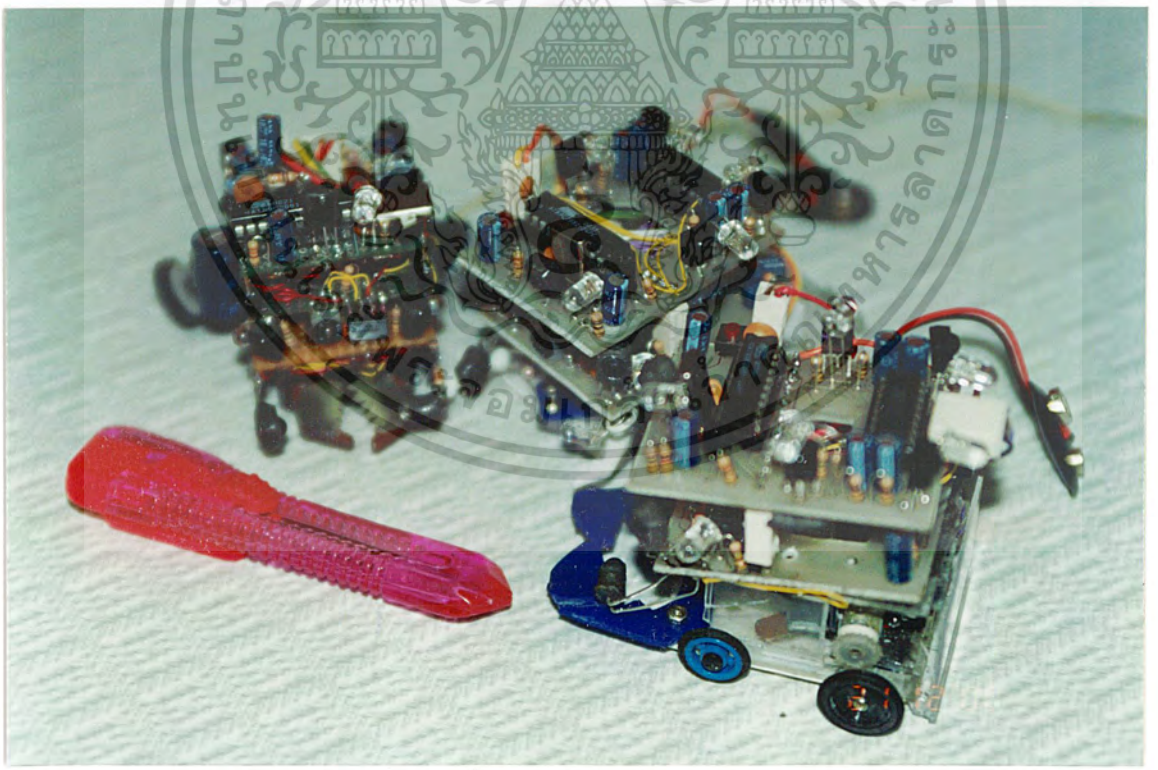
Hardware จัดเป็นส่วนที่เรามองเห็นได้สัมผัสได้ซึ่งแสดงผลออกมาตามโปรแกรมที่เรากำหนดไว้ซึ่งในส่วนของฮาร์ดแวร์ก็จะประกอบด้วย ส่วนของ DC motor ที่ใช้ในการขับเคลื่อนโดยควบคุมจาก Microcontroller และในส่วนของกลไกที่ใช้ในการหนีบวัตถุต่าง ๆ โดยจะใช้มอเตอร์ทำการหมุนในการควบคุมการหนีบและคลาย ที่ 45 ~ 90 องศา ซึ่งจะกล่าวโดยละเอียดในบทต่อไป และในส่วนของวงจรอิเล็กทรอนิกส์ ที่เป็นเสมือนสะพานที่ใช้เชื่อมต่อระหว่าง ฮาร์ดแวร์และ ซอฟต์แวร์ ให้ทำงานสัมพันธ์กัน อย่างมีประสิทธิภาพ ซึ่งอุปกรณ์ต่าง ๆ ในวงจรอิเล็กทรอนิกส์ จะขอกล่าวโดยละเอียดในบทต่อไปเช่นกัน ตลอดจนถึงข้อมูลทางเทคนิคที่ใช้ในการสร้างหุ่นยนต์ IC เบอร์ต่าง ๆที่จำเป็นต้องใช้ในการควบคุมนอกเหนือจาก CPU และที่สำคัญในการสร้างเราต้องพิจารณาถึงขอบเขตและรูปร่างของหุ่นยนต์ให้มีขนาดเล็กที่สุดเท่าที่จะทำได้เพราะในปัจจุบันหุ่นยนต์จะถูกพัฒนาให้มีขนาดที่เหมาะสมกับงานเพราะเราอาจคำนึงถึงพื้นที่ที่ใช้ในการใช้งาน

Software ในส่วนของซอฟต์แวร์นี้จัดเป็นส่วนที่สำคัญอีกส่วนหนึ่งที่ขาดไม่ได้อย่างแน่นอน เพราะถ้าไม่มีซอฟต์แวร์ หุ่นยนต์ก็จะไม่สามารถทำงานได้เลย ซอฟต์แวร์ก็คือชุดคำสั่งที่ใช้ในการสั่งงานให้หุ่นยนต์ทำงานตามคำสั่งนั้นๆ โดยเราต้องทำการเขียนโครงสร้างการทำงานก่อนว่าเราต้องการให้มันมีการทำงานอย่างไรชุดคำสั่งก็เป็นโปรแกรมภาษา Assembly ที่ใช้อัดลงใน Microcontroller ทำให้ CPU ตัวนี้เปรียบเสมือนสมองของหุ่นยนต์ที่ใช้ในการปฏิบัติการต่าง ๆ ตามที่เรากำหนด ถ้าขาด CPU ก็ทำงานไม่ได้เช่นกัน ในส่วนของซอฟต์แวร์นี้ เราจะต้องทำการเขียนโปรแกรมให้รองรับกับการใช้งานอย่างเหมาะสม และถูกต้อง โดยจะกล่าวให้ทราบรายละเอียดเกี่ยวกับซอฟต์แวร์ว่า ซอฟต์แวร์ที่เราทำการเขียนขึ้นนั้นต้องสามารถควบคุมในส่วนของภาคใดบ้าง เช่น ในส่วนของการควบคุม DC Motor กลไกการหนีบ ชุดSensor เราต้องทำการเขียนโปรแกรมรองรับกับส่วนที่กล่าวมาข้างต้นนี้อย่างถูกต้องและสมบูรณ์ ซึ่งจะขอกล่าวรายละเอียดส่วนต่าง ๆ เกี่ยวกับซอฟต์แวร์ในบทต่อไปเช่นกัน

โดยต่อไปนี้จะขอกล่าวถึงรายละเอียดที่สำคัญตามลำดับ โดยจะขอกล่าวถึงส่วนของฮาร์ดแวร์ก่อนว่าประกอบด้วยอะไรบ้างที่มีส่วนสำคัญและจะขอกล่าวถึง Block diagram การทำงานของอุปกรณ์ฮาร์ดแวร์ ที่มี CPU เป็นตัวควบคุมหลักทำให้สามารถทำงานได้อย่างสมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





รูปที่ (1-3) แสดงรูป Ant robot

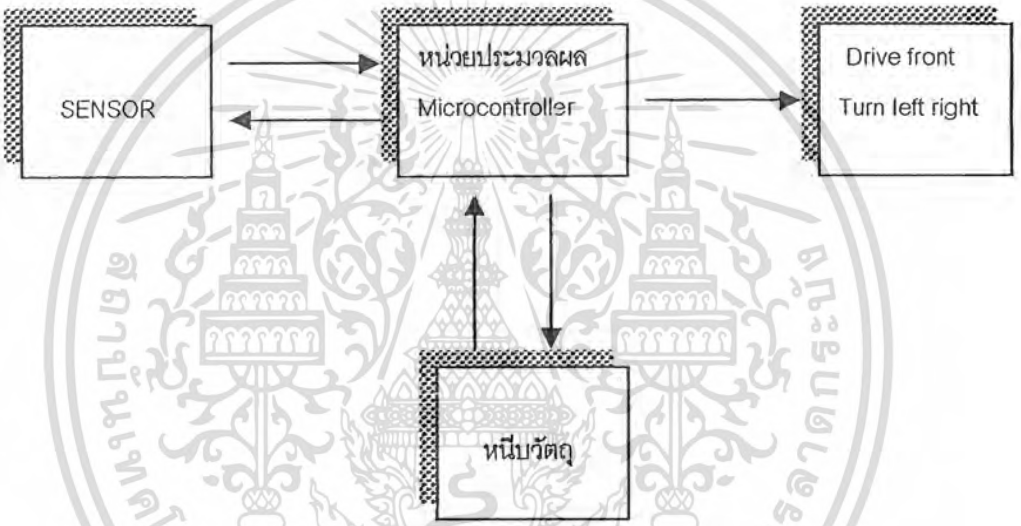
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ส่วนประกอบทาง Hard ware

ในส่วนของฮาร์ดแวร์จะขอกล่าวถึง Block diagram เป็นลำดับแรกเพื่อเข้าใจหลักการทำงานที่เป็นลำดับขั้นตอน เพื่อง่ายต่อการทำความเข้าใจในส่วนของฮาร์ดแวร์

BLOCK DIAGRAM



(2-1) Blockdiagram แสดงการทำงานพื้นฐานเบื้องต้น

ในการที่เราจะทำความเข้าใจ เกี่ยวกับลักษณะการทำงานของหุ่นยนต์ การทำความเข้าใจเกี่ยวกับ Blockdiagram เป็นสิ่งที่สำคัญเพราะ จะทำให้เราสามารถที่จะมองเห็นภาพรวมการทำงานของหุ่นยนต์ได้ดีขึ้นซึ่งในแต่ละ Block มีการทำงานที่สัมพันธ์กันดังต่อไปนี้

Sensor ในส่วนของ Sensor เปรียบเสมือนกับประสาทการรับรู้ทั้งหมดของหุ่นยนต์ที่ทำให้มันสามารถทำงานได้ เป็นส่วนแรกของการทำงานทั้งหมด เพราะเมื่ออุปกรณ์ Sensor ทำการตรวจจับวัตถุหรือ สิ่งกีดขวางได้มันจะทำการส่งสัญญาณไฟฟ้าไปทำการประมวนผลที่ Microcontroller เพื่อให้มันทำงานตามที่เราต้องการ เช่น เมื่ออุปกรณ์ Sensor ทำการตรวจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จับสิ่งกีดขวางได้ มันจะทำการส่งสัญญาณไฟฟ้าไปที่ Microcontroller ทำการประมวลผลและสั่งให้หลบสิ่งกีดขวางตามโปรแกรมที่เรากำหนดไว้สังเกตได้จาก Block diagram นอกจากนี้ Sensor ในหุ่นยนต์ยังทำหน้าที่ส่งสัญญาณไปยังหุ่นยนต์ตัวอื่นๆ เพื่อให้ทราบตำแหน่งที่อยู่ในขณะนั้น

Microcontroller

เป็นส่วนที่จัดได้ว่าสำคัญที่สุดทำหน้าที่ในการประมวณผลที่ได้รับมาจากอุปกรณ์ Sensorจากนั้นจะทำการประมวลผลตามโปรแกรมที่เราได้ทำการ SET เอาไว้จากนั้นจะทำการส่งผลที่ได้ไปทำการสั่งงานให้อุปกรณ์ต่างๆทำงาน เช่น ควบคุมการหมุนของ Motor drive ซึ่งมีผลในการควบคุมการเดินการเลี้ยว และการถอยหลังของหุ่นยนต์ให้ถูกต้องซึ่ง Microcontroller ที่ใช้เป็น Microcontroller ในตระกูล MCS 51 เบอร์ 89c2051 มีทั้งหมด 20 ขา ใช้ภาษา Assembly ในการเขียนโปรแกรมเพื่อสั่งให้ทำงาน การเขียนโปรแกรมจะต้องทำการเขียนรูปแบบการทำงานขึ้นมาก่อน (flow chart) เพื่อที่เราสามารถที่จะมองเห็นภาพรวมของการทำงานได้อย่างถูกต้อง ส่วนรายละเอียดการใช้งานจะกล่าวโดยละเอียดในเรื่องของการใช้งาน นอกจากนี้ยังทำหน้าที่ในการสร้างสัญญาณต่างๆที่ใช้ในการควบคุม

ส่วนควบคุมการเดินและการเลี้ยว(Drive System)

ส่วนควบคุมการเดินและการเลี้ยวเป็นส่วนที่ควบคุมโดยการใช้ Dc motor ในการควบคุมการเดินโดยมี Microcontroller เป็นตัวสั่งการซึ่งจะทำงานสอดคล้องกับชุด Sensor ในการควบคุมการเลี้ยวเราใช้ Dc motor ในการควบคุมการเลี้ยวและการเดินจำนวน 2 ตัวอยู่ในส่วนด้านหลังของหุ่นยนต์นอกจากนี้ Dc motor ที่ใช้เป็น Motor ชนิดที่สามารถทดแรงในการทำงานได้ด้วย โดยเราทำการทดลองการใช้งานพบว่าสามารถที่จะลากหรือดันวัตถุที่มีน้ำหนักมากกว่าตัวของมันหลายเท่าตัวโดยคิดเป็นอัตราส่วนของน้ำหนักตัว และการยึดเกาะพื้นผิวของล้อ เนื่องจากพื้นผิวที่มีค่าของความเสียดทานสูงหุ่นยนต์มดสามารถที่จะลากวัตถุได้ดีขึ้น

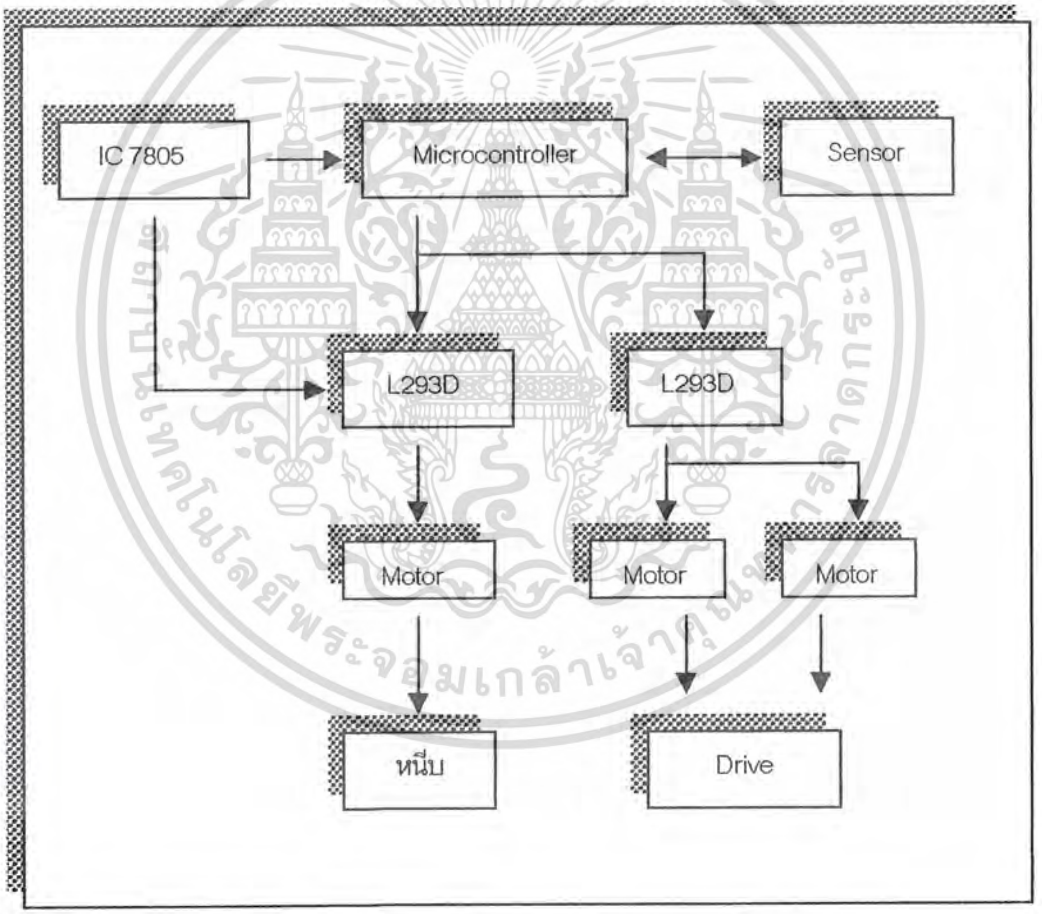
ส่วนกลไกที่ควบคุมการหนีบ

ในส่วนของกลไกการหนีบวัตถุเราทำการควบคุมโดยใช้ Dc motor เช่นกันโดยมีการทำงานที่สอดคล้องกับการเคลื่อนที่ซึ่งเราใช้ Motor ที่ทดแรงในการหนีบเช่นกัน เพื่อป้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กันไม่ให้การหนีบวัตถุหลุดก่อนจะถึงรัง ส่วนกลไกที่ใช้ในการหนีบนั้น จะกล่าวถึงโดยละเอียดในบทต่อไปจาก Blockdiagram เราจะเห็นว่าในส่วนของ Sensor มีการติดต่อกับ Microcontroller ซึ่งทำการควบคุมกลไกการหนีบตามที่เรากำหนดไว้ ซึ่งจะเห็นว่าในแต่ละส่วนมีการทำงานที่ติดต่อถึงกันและต้องอาศัยผลของการทำงานในแต่ละภาคเพื่อประสิทธิภาพในการทำงานที่สมบูรณ์แบบและถูกต้อง หากว่าการทำงานในภาคใดเสียไปก็จะทำให้ภาคต่างๆที่เกี่ยวข้องทำงานไม่ได้หรือทำงานได้ไม่เต็มที่ ดังนั้นทุกๆภาคที่กล่าวมาต้องมีการทำงานที่สมบูรณ์แบบและถูกต้องตามที่เรากำหนดเอาไว้

Blockdiagram รายละเอียดในการควบคุม



รูปที่ (2-2) เป็นรายละเอียดการควบคุมต่างๆของ Ant robot

เพื่อเป็นการมองภาพรวมการทำงานของ Ant robot สามารถแสดงได้จาก Block Diagramรูปที่ (2-2) จากรูปจะเห็นว่าจะมี IC 7805 ทำหน้าที่เป็น voltage Regulator จ่ายแรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดัน 5V ไปที่ Microcontroller ซึ่งในหุ่นยนต์มด 1 ตัวจะใช้ Voltage Regulator 7805 เพียง 1 ตัว ทำการจ่ายแรงดันให้กับภาคต่างๆ และ L293D ทั้ง 2 ตัว จากนั้น L293D ก็จะไปทำการควบคุม DC Motor ทั้ง 3 ตัว โดยแบ่งเป็น 1 ตัว ที่ใช้ในการหนีบวัตถุและ 2 ตัวที่ใช้ในการ Drive และการควบคุมที่เหลือก็คือ การควบคุมภาค Sensor ที่ต้องอาศัยการส่งสัญญาณออกไป ซึ่งสัญญาณที่ส่งออกไปจะอยู่ในย่าน Infrared และรับสัญญาณที่ได้เข้ามาทำการประมวลผลที่ Microcontroller ซึ่งจะไปทำการควบคุมในส่วนต่างๆต่อไป ส่วนในเรื่องของรายละเอียดเกี่ยวกับการทำงานของ L293d และภาค Voltage Regulator รวมถึงรายละเอียดในส่วนต่างๆจะขอกล่าวถึงในบทต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
รายละเอียดเกี่ยวกับ Hardware

ในการทำงานในส่วนของ Hard ware ประกอบด้วยรายละเอียดในส่วนต่างๆที่เราจะต้องทำความเข้าใจและต้องเข้าใจเกี่ยวกับการทำงานของวงจรไฟฟ้าพอสมควรโดยจะขอกล่าวรายละเอียดทาง Hard ware ดังนี้

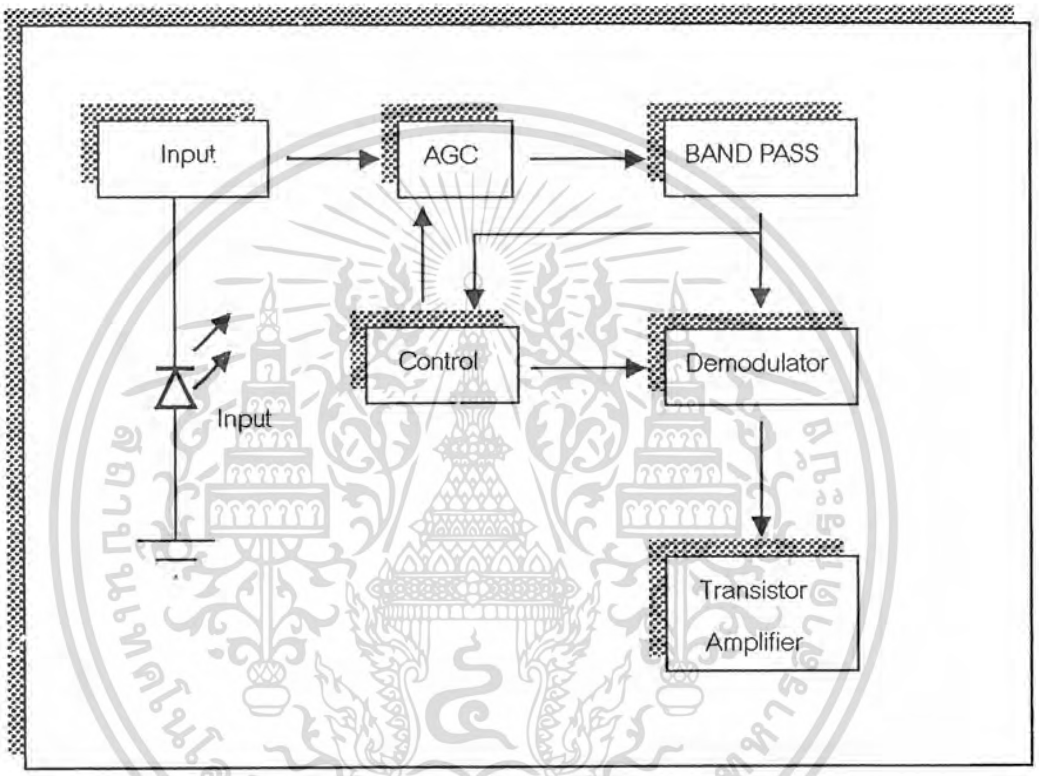
SENSOR

เปรียบเสมือนประสาทการรับรู้ของหุ่นยนต์ซึ่งถ้าขาดในส่วนของภาค Sensor นี้ก็จะทำให้ ant robot ทำงานไม่ได้เลย นั่นคือมันจะไม่ตอบสนองต่อสิ่งรอบๆ ข้างซึ่งหน้าที่ของ Sensor ก็คือตรวจจับสิ่งต่างๆ รอบๆ ข้างและจะทำงานตาม Blockdiagram ที่อธิบายไว้แล้วในตอนต้น หุ่นยนต์มดจะใช้ Sensor ทั้งหมด 7 ชุดซึ่งจะประกอบด้วย 4 ชุดที่ทำหน้าที่ ในการตรวจจับสิ่งกีดขวางและอีก 1 ชุดทำหน้าที่ในการตรวจจับวัตถุที่จะทำการหนีบส่วนอีก 2 ชุดที่เหลือจะทำหน้าที่ในการส่งสัญญาณในการติดต่อกับหุ่นยนต์ตัวอื่นๆ ที่อยู่รอบๆ ข้างซึ่งจะมีลักษณะการทำงานคือ จะมีชุดที่ทำหน้าที่ในการส่งสัญญาณ Iinfrared โดยทำการ modulate กับสัญญาณพัลส์ที่ความถี่ 38 kHz จากนั้นทำการส่งสัญญาณออกไปยังหุ่นยนต์มดตัวที่อยู่รอบๆ ให้ทราบ สาเหตุที่เราต้องทำการ Modulate กับสัญญาณพัลส์ก็เนื่องจากว่าในการส่งสัญญาณ Infrared โดยทั่วไปมักจะมีการรบกวนของแสงต่างๆ จากสิ่งแวดล้อมภายนอกที่ประกอบด้วยรังสีต่างๆมากมายทำให้การส่งสัญญาณออกไป และหลังจากรับสัญญาณเข้ามาทำการประมวลผลแล้วก็จะทำให้เกิดความผิดพลาดขึ้นได้ในการประมวลผลเพราะบางครั้งทางภาครับอาจจะแยกไม่ถูกว่าสัญญาณรบกวน หรือสัญญาณที่จะต้องนำเอาไปประมวลผล ดังนั้นเราจะต้องทำการแยกสัญญาณที่เราจะทำการส่งออกไปกับสัญญาณรบกวนต่างๆ ที่อยู่รอบๆ ออกจากกันให้อุปกรณ์ที่ทำหน้าที่ในการรับสัญญาณที่จะนำไปทำการประมวลผลไม่เกิดความสับสน สามารถที่จะแยกได้ระหว่างสัญญาณรบกวนและสัญญาณที่จะนำไปประมวลผลโดยเราทำการ Modulate หรือเรียกว่า (รวมสัญญาณ) ระหว่างสัญญาณ Infrared กับสัญญาณพัลส์ซึ่งจะทำการ Modulate กันที่ความถี่ 38 kHz ซึ่งเป็นความถี่ที่เหมาะสมสำหรับการใช้งานโดยเราสามารถที่จะใช้ Microcontroller ทำหน้าที่ในการผลิตสัญญาณพัลส์ส่งไปทำการ Modulate ได้โดยที่อุปกรณ์รับสัญญาณ Infrared จะเป็นแบบ Infrared module ซึ่งมันจะรับสัญญาณเฉพาะที่ความถี่ 38kHz เท่านั้นทำให้สามารถที่จะกำจัดตัวรบกวนหรือสัญญาณรบกวนต่างๆได้เป็นอย่างดีเพราะว่าในย่านความถี่แสงในบรรยากาศที่มีอยู่มากมายจะไม่มีผลกับ Infrared module เพราะว่ามีความถี่ไม่ตรงกันกับภาครับสัญญาณ อีกทั้งเราทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การ Modulate กับสัญญาณพัลส์จึงมีความแน่นอนในการส่งสัญญาณ และที่ภาครับจะต้องมีตัวถอดหรัสสัญญาณ (Demodulate) ที่เราได้ทำการ Modulate เอาไว้เพื่อกู้สัญญาณที่ถูกต้องกลับคืนมา จากนั้นก็จะนำผลที่ได้ไปทำการประมวลผลต่อไป

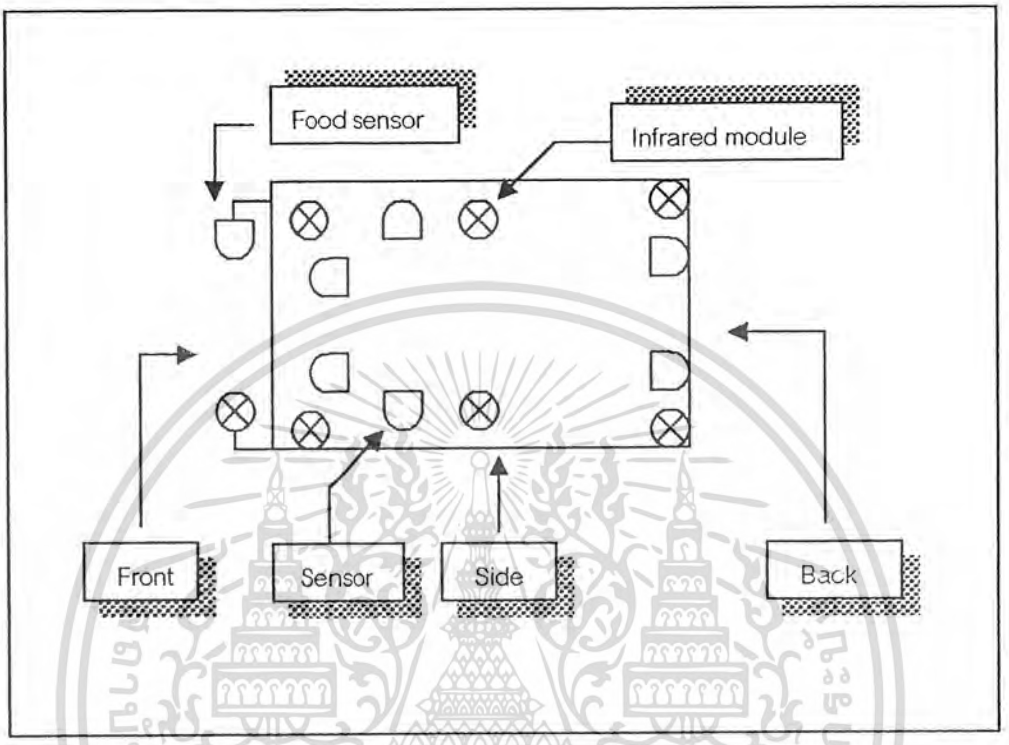
(3-1) Blockdiagram แสดงการทำงานของ infrared module Tsop1838



จากรูปที่ 3-1 แสดงการทำงานของ Infrared module เบอร์ Tsop1838 ที่แสดงดังรูปนี้เป็น Infrared module ที่ใช้ใน Ant Robot ทั้งหมดรับสัญญาณที่ความถี่ที่ 38 kHz เท่านั้น จากรูปเราจะเห็นว่าโครงสร้างภายในประกอบด้วยภาคต่างๆ เริ่มจากในส่วนของ input ที่รับสัญญาณเข้ามาและทำการส่งต่อไปยังภาค AGC (Automatic gain control) ที่ทำหน้าที่ในการควบคุมเกณฑ์การขยายของสัญญาณ จากนั้นก็ทำการส่งสัญญาณต่อไปที่ Band pass filter ที่ทำหน้าที่ในการกรองเอาเฉพาะความถี่ที่ต้องการโดยใช้ Band pass filter จากนั้นทำการส่งสัญญาณไปที่ Demodulator เพื่อทำให้สัญญาณที่เราทำการ Modulate ในตอนแรกกลับมาเหมือนเดิมเพื่อทำให้การประมวลผลไม่ผิดพลาดก่อนจะทำการส่งไปที่ Transistor amplifier เพื่อทำการขยายสัญญาณที่เราทำการ Demodulate แล้วเพื่อต้องการให้สัญญาณที่ได้มีความสมบูรณ์ที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นนำสัญญาณที่ได้ไปทำการประมวลผลที่ Microcontroller เพื่อทำการวิเคราะห์ว่าสัญญาณที่ได้รับมาควรจะทำอย่างไรต่อไป



(3-2) รายระเอียดในการติดตั้ง Sensor

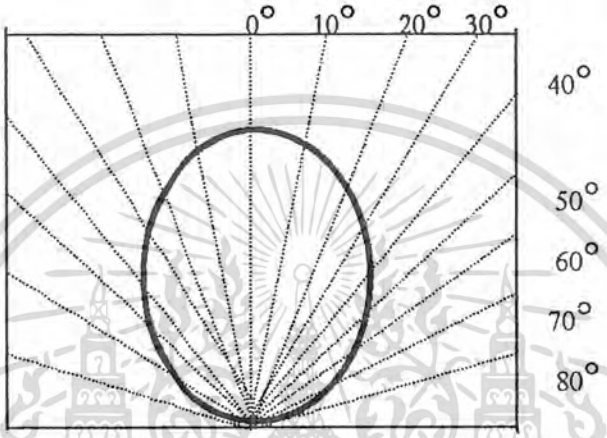
จากรูปเราจะสังเกตุได้ว่าจะมี Sensor อยู่ทางด้านข้างทั้งหมด 2 ชุด ด้านหน้า 2 ชุด ส่วน Sensor ทางด้านหน้าสุดทำหน้าที่ในการตรวจจับอาหารส่วน 2 ชุดสุดท้ายทางด้านหลังทำหน้าที่ในการส่งสัญญาณให้กับหุ่นยนต์ตัวอื่นๆ อุปกรณ์ที่ใช้เป็นตัวรับแสง Infraredใน ant robot เราใช้เป็น Infrared module ทั้งหมดแต่อย่างไรก็ตามเรายังพบปัญหาอีกอย่างนั่นคือมักจะเกิดการรบกวนกันของ Infrared แต่ละตัวเนื่องจากว่าการติดตั้ง Sensor แต่ละชุดมีความใกล้เคียงกันทำให้สัญญาณมีการรบกวนกัน เราสามารถที่จะแก้ปัญหาโดยทำการชีนอุปกรณ์ที่ใช้ในการรับแสง Infrared เพื่อ เป็นการลดการกวนของสัญญาณ นอกจากนี้เรายังพบว่าเกิดปัญหาที่ dc Motor เนื่องมาจากเกิดการดึงกระแสที่ใช้งานมากมีผลทำให้กระแสที่จ่ายให้กับอุปกรณ์ Sensor ไม่เพียงพอดังนั้นเราพอที่จะแก้ไขได้โดยเพิ่มกระแสที่แหล่งจ่าย

หากเราจะกล่าวว่าในการส่งสัญญาณ Infrared เพื่อการใช้งานเราสามารถที่จะกล่าวได้ว่า ลักษณะการใช้งานนั้นสามารถที่จะแบ่งได้เป็น 2 ลักษณะการใช้งานนั่นคือ

1. ลักษณะการใช้งานแบบอาศัยหลักการสะท้อนของแสง Active logic “1”
2. ลักษณะการใช้งานแบบอาศัยหลักการตัดลำแสง Active logic “0”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในลักษณะการทำงานในแบบแรกเป็นการใช้งานของ Sensor ชุดที่ใช้ตรวจจับสิ่งกีดขวางโดยที่ตัวส่งลำแสง Infrared ทำการส่งลำแสงออกไปหากว่ายังไม่มีสิ่งกีดขวางที่จะมาทำการสะท้อนลำแสง ตัวรับส่งจะมีสภาวะเป็น Logic "0" จนกระทั่งหุ่นยนต์มดเคลื่อนที่ไปพบกับสิ่งกีดขวางที่จะมาทำหน้าที่ในการสะท้อนสัญญาณไปยังตัวรับแสง ตัวรับแสงก็จะได้รับสัญญาณ Logic "1" จากนั้นก็จะนำสัญญาณที่ได้รับนี้ส่งไปทำการประมวลผลต่อไป

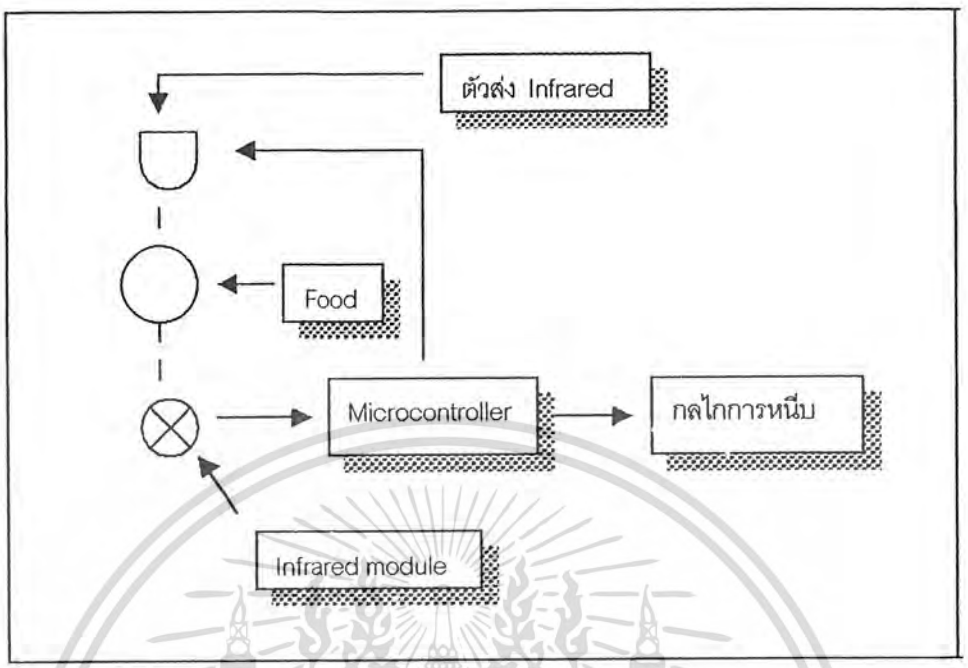


รูปที่(3-3)แสดงทิศทางในการส่งลำแสง Infrared และองศาของการกระจายลำแสง

สามารถที่จะสังเกตุได้ว่าทิศทางในการกระจายของลำแสง Infrared มีลักษณะของการกระจายลำแสงแบบแบ่งเป็นองศาโดยจะห่างกันทุกๆที่ 10 องศาด้วยลักษณะการกระจายลำแสงแบบทุกทิศทางทำให้ระบบของตัวส่ง Infrared มีความแน่นอนและแม่นยำในการตรวจสอบส่วนลักษณะของเส้นทึบสีดำเป็นระยะทางในการส่งลำแสงที่มีประสิทธิภาพโดยสามารถที่จะสังเกตุได้ว่าในกรณีที่มุมมีค่าสูงเท่าใดการส่งลำแสงจะมีประสิทธิภาพน้อยลงนั่นคือมันไม่สามารถที่จะส่งสัญญาณให้กับตัวรับแสงได้หากมุมในการส่งลำแสง Infrared มีค่ามากเกินไป

ส่วนการทำงานในลักษณะที่สองนี้เรานำมาใช้ในการตรวจจับวัตถุหรืออาหารที่จะทำการหนีบโดยเราทำการติดตั้ง Sensor ไว้ในบริเวณส่วนหน้าสุด โดยหลักการทำงานก็คือตัวส่งลำแสงจะทำหน้าที่ในการส่งสัญญาณออกไปให้กับ Infrared module ตลอดเวลาทำให้ที่ตัวรับแสงมีสถานะเป็น Logic"1" ตลอดเวลาแต่ในกรณีที่หุ่นยนต์มดเดินไปพบอาหารนั่นคืออาหารไปบดบังลำแสง Infrared ที่ทำการส่งให้กับ Infrared module หรือตัวรับแสงมีผลทำให้ตัวรับแสงมีสถานะเป็น logic"0" ทันทีส่งสัญญาณที่ได้ไปทำการประมวลผลและหน่วยประมวลผลก็จะสั่งให้กลไกการหนีบให้ทำงานต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(3-4) แสดงการทำงานของชุด Sensor อาหาร

จากรูปที่ 2-2 เราจะเห็นว่าตัวส่งลำแสงทำการส่งสัญญาณไปที่ตัวรับแต่มีอาหารมาบังลำแสงในขณะที่อาหารหรือวัตถุดิบมาบังลำแสงเอาไว้ทำให้ค่าที่ตัวรับแสงมีค่าเป็น"0" จากนั้น Infrared module ก็ทำการส่งสัญญาณที่ได้ไปประมวลผลที่ Microcontrollerจากนั้นก็ไปควบคุมกลไกการหนีบให้ทำงานตามโปรแกรมที่เรากำหนดเอาไว้จากรูปจะเห็นว่า Microcontroller ไปควบคุมที่ตัวส่ง Infrared ด้วยเนื่องมาจาก Microcontroller ต้องทำการสร้างสัญญาณพัลส์และทำการ Modulate กับสัญญาณ Infrared ก่อนทำการส่งไปยังตัวรับ โดยทั่วไปแล้วลักษณะการใช้งานของ Sensor ตามหลักการก็จะมีการใช้งาน 2 ลักษณะนี้

ส่วนหลักการทางทฤษฎีการทำงาน ก็คือ เมื่อ Infrared มีการนำกระแสเกิดขึ้น อีเล็กตรอนจะเคลื่อนที่ผ่านสารกึ่งตัวนำชนิดพิเศษทำให้เกิดพลังงานแสงในย่าน infrared โดยทั่วไปแล้วจะกำเนิด Infrared ในช่วงสองความยาวคลื่นคือ Infrared ที่สร้างมาจากสารแกลเลี่ยมอาเซไนด์จะให้ความยาวคลื่นประมาณ 900 นาโนเมตรและ Infrared ที่สร้างมาจากแกลเลี่ยมอลูมินั่มอาเซไนด์จะกำเนิดแสงที่มีความยาวคลื่นประมาณ 880 นาโนเมตรซึ่งรอยต่อ P-Nของ LED จะเป็นแหล่งกำเนิดโฟตอนสำหรับสารแกลเลี่ยมอาร์เซไนด์จะทำให้เกิดโฟตอน 88 ตัวต่ออิเล็กตรอน 100 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติของ LED Infrared

แรงดันตกคร่อมที่รอยต่อ P-N ของไดโอดจะต้องมีค่ามากว่าแรงดันเทรชโฮลด์จึงจะสามารถที่จะทำให้ Diode นำกระแสได้ สำหรับ Silicon Diode แรงดันที่ทำให้นำกระแสได้มีค่าประมาณ 0.6 Volt ส่วน LED ที่สามารถให้แสงในย่านมองเห็นได้ถ้าทำจากสารGap ซึ่งให้แสงสีเขียวจะมีค่าแรงดันในการเริ่มทำงานที่ประมาณ 2.1 ถึง 2.8 Volt ถ้าเป็น LED ที่ทำมาจาก AlGaAs จะให้แสงสีแดงมีแรงดันทำงานที่ 1.75 ถึง 2.5 Volt ส่วน LED ที่ให้ลำแสงใกล้กับย่าน Infrared ที่ทำมาจากสาร GaAs จะมีแรงดันทำงานที่ 1.5 Volt

พลังงานที่ได้จากการเปล่งแสงของ LED สามารถที่จะหาได้จากกระแสไบอัสตรงของไดโอดและจำเป็นอย่างยิ่งที่จะต้องระมัดระวังไม่ให้กระแสในส่วนนี้มีค่าสูงจนเกินไปซึ่งอาจมีผลทำให้ไดโอดเสียหายได้

Silicon LED ที่ทำมาจาก GaAs จะให้กำลังงานประมาณ 5 มิลลิวัตต์ ที่กระแสไบอัสตรง 100 มิลลิแอมป์ LED ที่ทำมาจาก AlGaAs จะให้กำลังงานเป็น 2 เท่าเมื่อเราทำการจ่ายกระแส

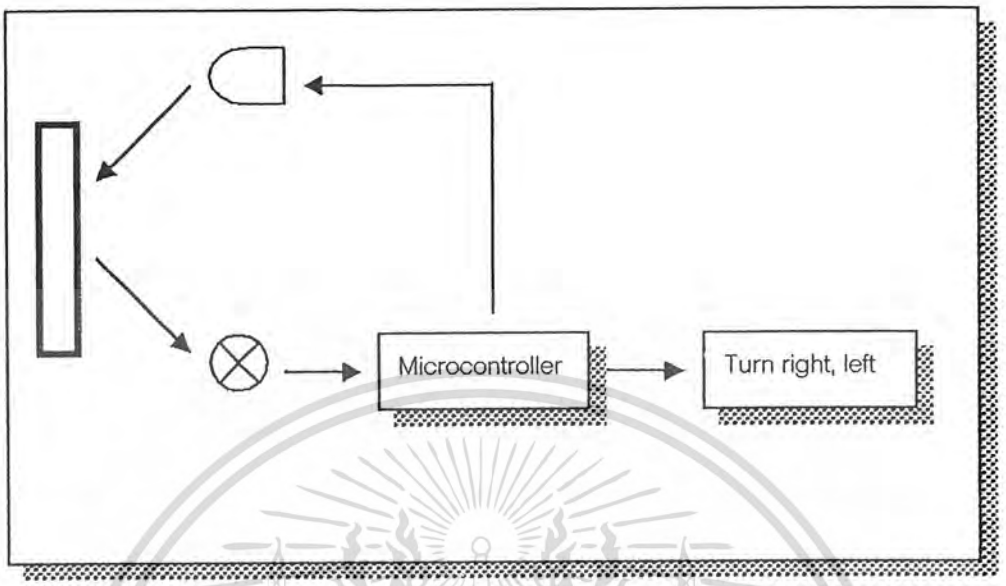
ไบอัสตรงที่ค่าเดียวกันข้อดีกว่าอีกประการหนึ่งของ LED ชนิด AlGaAs คือมีค่า Rest time และ full time ที่มีค่าเร็วกว่าประมาณ 0.5 ไมโครวินาที ในขณะที่ GaAs Silicon Diode มีค่า 1.5 ไมโครวินาที

ข้อดีอีกประการหนึ่งคือการเปล่งลำแสงของ LED ที่ความยาวคลื่น 880 นาโนเมตร จะมีค่าใกล้เคียงกับความยาวคลื่นที่ Silicon Phototransistor มีค่าความไวสูงสุดจึงนับได้ว่าเป็นการเหมาะสมอย่างยิ่งที่จะใช้ LED ที่มีความยาวคลื่น 880 นาโนเมตรแทนการใช้ LED ที่มีความยาวคลื่น 940 นาโนเมตร

นอกจากนี้ LED ที่ให้ความยาวของคลื่นแสงที่ 880 นาโนเมตร จะมีข้อดีคือ จะไม่ถูกดูดกลืนโดยละอองน้ำเหมือนกับ LED ที่ใช้ความยาวคลื่น 940 นาโนเมตร ที่จะถูกดูดกลืนจากละอองน้ำ ดังนั้นเราจึงสามารถที่จะนำไปใช้ในการตรวจจับละอองน้ำในอากาศได้

LED ชนิด 940 นาโนเมตร จึงไม่เหมาะสมกับการสื่อสารด้วยแสงภายนอกเนื่องมาจากการที่ LED ชนิด 940 นาโนเมตรจะถูกดูดกลืนจากละอองน้ำนั่นเอง ส่วน LED ชนิด Silicon ที่ทำมาจาก GaAs มักจะถูกใช้เป็นแหล่งกำเนิดแสงในย่านอินฟาเรด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



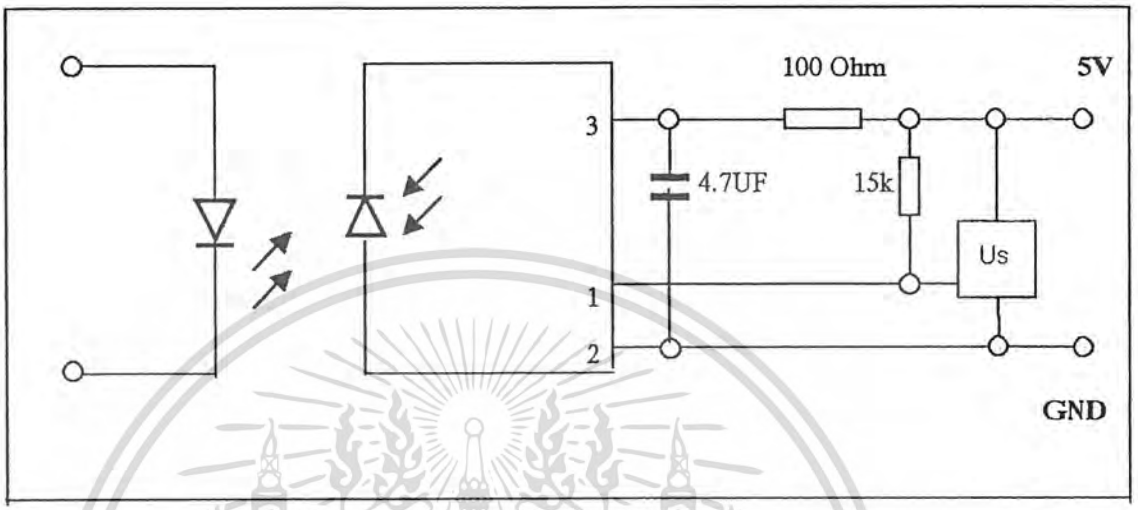
(3-5) แสดงการทำงานของชุด Sensor ที่ใช้ในการตรวจจับสิ่งกีดขวาง(refector sensor)

นอกจากนี้เรายังสามารถใช้ Sensor แบบอื่นๆได้อีกเช่นใช้ Micro Switch แทนการใช้ Sensor ตรวจจับสิ่งกีดขวางก็สามารถใช้งานได้แต่ข้อดีในการใช้ Sensor แบบ Infrared ก็คือมันสามารถที่จะทำการควบคุมระยะทางในการ Sensor ได้ไกลกว่า มีความไวในการตรวจจับสามารถที่จะควบคุมระยะการ Sensor ได้ ประหยัดพื้นที่ในการติดตั้ง การตรวจจับมีความแน่นอน ในปัจจุบัน Sensor ชนิดนี้เป็นที่นิยมใช้กันอย่างแพร่หลายแต่ก็ยังมีข้อเสียในการใช้งานนั่นก็คือ มักจะเกิดการรบกวนจากแสงภายนอก และตัวส่ง ลำแสงยังมีความไวต่อการเปลี่ยนแปลงของ กระแสที่ไบอัสเมื่อเกิดการดึงกระแส สูงๆดังนั้นเราควรคำนึงถึงการใช้งานที่เหมาะสมกับการใช้งาน Sensor และควรทราบการแก้ไขจุดบกพร่องต่างๆ เพื่อสามารถที่จะใช้งาน ได้อย่างสมบูรณ์แบบแต่อย่างไรก็ตามการใช้งานอุปกรณ์ Sensor ก็จะมีลักษณะการใช้งานโดยทั่วไปตามที่ได้กล่าวไว้แล้ว

ส่วนการต่อใช้งานอุปกรณ์ตรวจจับหรือ ชุด Sensor สามารถที่จะอธิบายได้ดังรูปที่ (3-6) ซึ่งเป็นการต่อวงจรใช้งาน ชุด Sensor ที่ประกอบด้วยภาครับและภาคส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



รูปที่ (3-6) เป็นการต่อใช้งานอุปกรณ์ชุด Sensor

จากรูปเป็นการต่อใช้งานที่สามารถใช้งานได้ โดยสัญญาณที่ถูกส่งออกที่ Output 5 v จะถูกส่งไปยัง Microcontroller เพื่อทำการประมวลผล จากรูปเราจะเห็นภาพของการทำงานโดยจะมีตังส่งและตัวรับสัญญาณ Infrared และมีอุปกรณ์ capacitor ทำหน้าที่ในการฟิลเตอร์จากรูปเป็นลักษณะการต่อใช้งานที่สามารถที่จะนำไปใช้งานได้เลย

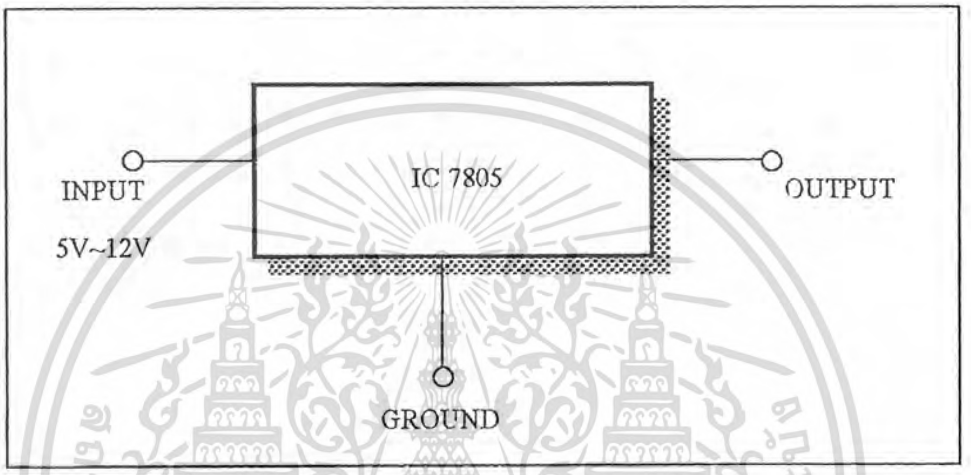
Voltage Regulator

ภาค Voltage Regulator ก็คือ ภาคที่ใช้ในการแปลงแรงดันไฟฟ้า จากปรกติเพื่อให้เป็น แรงดันใช้งานสำหรับอุปกรณ์ TTL เพราะอุปกรณ์ TTL ใช้แรงดันในการใช้งานที่ต้องอาศัยความแน่นอน นั่นคือ ใช้ แรงดัน 5 Volt ในการใช้งาน ดังนั้นเราจึงมีความจำเป็นอย่างยิ่งที่จะต้องทำการแปลงระดับแรงดันไฟฟ้า จากแหล่งจ่ายที่ใช้ตามปรกติให้เป็น 5 Volt ในการใช้งานใน หุ่นยนต์มดเราใช้ IC 7805 เพื่อเป็นตัว Regulate แรงดันไฟฟ้าที่ใช้จากปรกติ 9 Volt ให้เป็นแรงดัน 5 Volt สำหรับอุปกรณ์ TTL ซึ่งลักษณะของ IC 7805 ที่ใช้มันจะทำการแปลงแรงดัน 9 Volt ให้เหลือเพียง 5 Volt เพื่อการใช้งาน โดยทั่วไปแล้ววงจร Regulate มีอยู่หลายชนิด แต่ที่นิยมใช้กี่จะเป็นลักษณะของ IC เช่น เบอร์ 7805 จะแปลงระดับแรงดันไฟฟ้าได้ 5 Volt , 7809 ก็จะแปลงระดับแรงดันได้ 9 Volt แต่เราก็คงต้องคำนึงถึงลักษณะการใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

42687

งานที่เหมาะสมด้วยเช่นกัน ภาค Voltage Regulator จะจัดเป็นภาคแรกของวงจรในการใช้งานวงจรที่ประกอบด้วยอุปกรณ์ TTL เพื่อทำการแปลงแรงดัน DC Voltage สำหรับอุปกรณ์ต่าง ๆ ในวงจรที่ใช้ในการควบคุม ในปัจจุบันโดยส่วนมากในตลาดนิยมใช้อุปกรณ์ TTL กันอย่างแพร่หลาย ส่วนอุปกรณ์ ที่อยู่ในตระกูล CMOS ก็ยังมีใช้อยู่แต่ไม่ค่อยเป็นที่นิยมเท่าใดในปัจจุบัน



รูปที่ (3-7) เป็นรูปของอุปกรณ์ Regulator

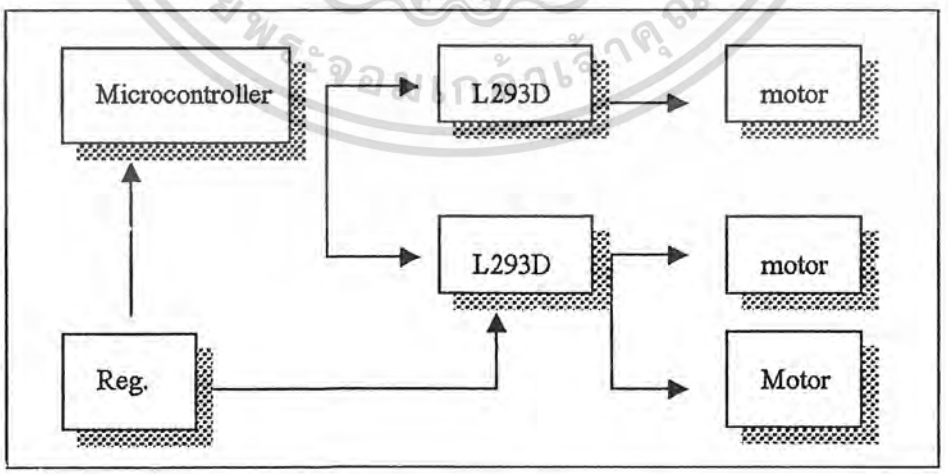
โดยทั่วไปแล้ว IC 7805 ต้องใช้แรงดันไฟฟ้า (input voltage) ที่ใช้ประมาณ 5 Volt~12Volt เพราะในขณะที่มันทำการ Regulate แรงดันไฟฟ้าจะเกิดความร้อนในตัวมันเอง อาจทำให้เกิดความเสียหายได้ ดังนั้น เราต้องคำนึงถึงการใช้งาน นั่นคือ คำนึงถึงว่าในวงจรกินโหลด กระแสมากน้อยเพียงใด เพราะถ้าในวงจรมีการใช้กระแสมากหรือต้องการ Voltage มาก ๆ IC 7805 ก็ต้องทำงานหนัก อาจทำให้เกิดความเสียหายได้ ดังนั้นเราอาจแก้ปัญหาโดยการติดแผ่นระบายความร้อนให้กับ IC 7805 หรือต้องลดภาระโหลดให้น้อยลง ในการทำงานของหุ่นยนต์มดนั้น มีการนำเอา IC 7805 มาใช้งานเพียง 1 ตัวเท่านั้น เพราะภาระทางโหลดในวงจรก็มีเพียงอุปกรณ์ TTL เช่น Microcontroller และอุปกรณ์ L293D ที่ใช้ในการควบคุม MOTOR และที่ L293D อาจต้องมีการใช้กระแสมาก เนื่องจากเป็นตัวที่ใช้ควบคุม MOTOR แต่เราพบว่าไม่มีปัญหาในการใช้งาน เมื่อต้องจ่ายกระแสให้กับ MOTOR ที่ละ2 ตัวพร้อมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุม MOTOR (L293D)

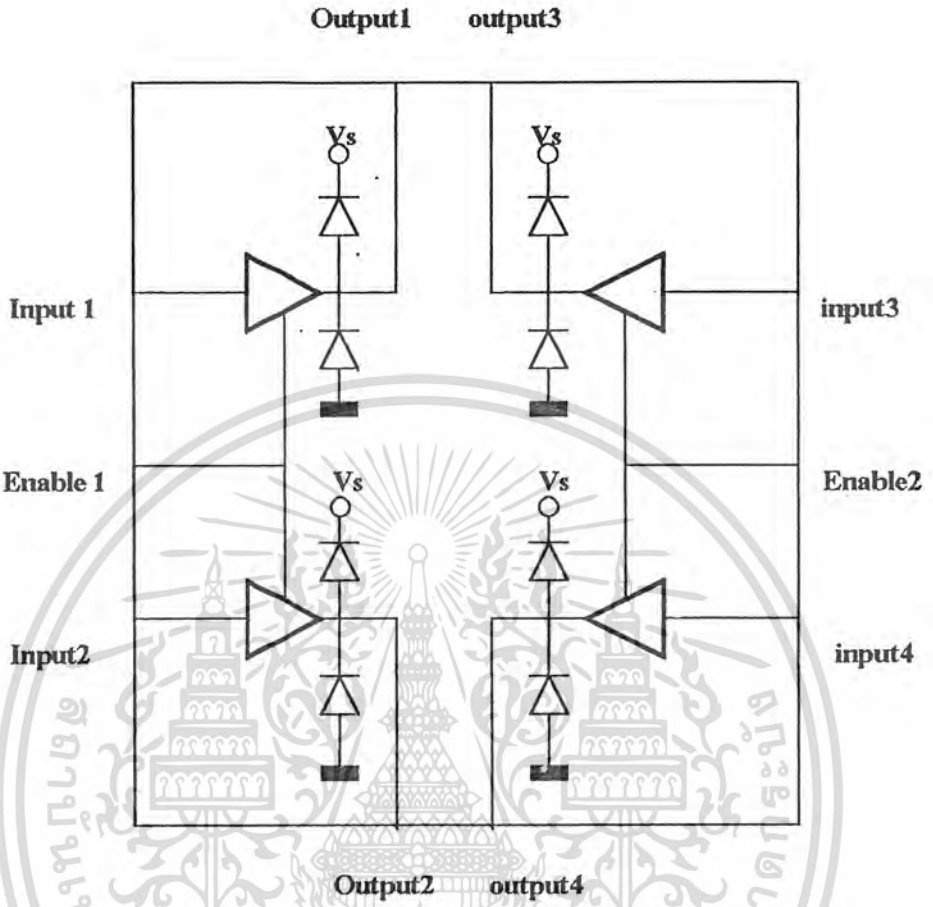
L293D เป็นอุปกรณ์ในตระกูล TTL ใช้แรงดันในการทำงานที่ 5 Volt input เป็นอุปกรณ์ที่ใช้ในการควบคุมทิศทางการหมุนของ DC MOTOR ซึ่งโดยทั่วไปแล้วทิศทางการหมุนของ MOTOR เราสามารถที่จะควบคุมได้โดยการกลับหรือสลับขั้วแรงดันที่จ่ายเข้าไปที่ MOTOR ดังนั้น MOTOR ก็จะทำการหมุนกลับทิศทางกัน เราจึงใช้ IC L293D ในการควบคุมทิศทางการหมุนของ MOTOR เพราะง่ายต่อการควบคุม และภายในตัวอุปกรณ์ L293D ยังมีภาคขยายสำหรับกระแสที่จ่ายไปยัง MOTOR อีกด้วย เราเพียงแต่ทราบสภาวะในการควบคุมอย่างไร แล้วเราก็ทำการควบคุมได้ L293D จะจ่าย Current Max ที่ output 1.2A สำหรับกระแสที่ Output ที่ประมาณ 1-1.2A ก็เพียงพอแล้วสำหรับ Drive MOTOR ใน ANT ROBOT 1 ตัว จะมีตัวควบคุม MOTOR L293D จำนวน 2 ตัว เนื่องจากว่า L296D 1 ตัวนั้นสามารถที่จะควบคุม MOTOR ได้ถึง 2 ตัว แต่ ANT ROBOT ใช้ MOTOR ในการควบคุมการเดินทั้งหมด 3 ตัว ดังนั้นจำเป็นที่จะต้องใช้ L293D ถึง 2 ตัวเพื่อควบคุม DC MOTOR 3 ตัว แต่ในการควบคุม L293D นั้นจะต้องใช้ MICRO CONTROLLER เนื่องจาก MICRO CONTROLLER เปรียบเสมือนสมองสั่งการไปยัง L293D ทั้ง 2 ตัวว่าจะให้ MOTOR ตัวใดทำงาน และหมุนไปในทิศทางใด ตามโปรแกรม และเงื่อนไขที่กำหนดเอาไว้ ดังนั้น เราจึงต้องทราบว่า L293D สามารถที่จะควบคุม DC MOTOR ได้อย่างไร

นอกจากนี้ยังมี IC อีกหลายเบอร์ในตระกูล L293D ที่สามารถที่จะนำไปประยุกต์ใช้ในงานที่เกี่ยวกับการควบคุม Motor ที่ต้องการกระแสสูงๆ ในการใช้งานแต่ที่สำคัยเพียงแค่เราต้องทราบสภาวะการควบคุมจึงจะสามารถที่จะใช้งานได้



รูปที่ (3-8) เป็นการต่อใช้งานอุปกรณ์ L293D กับอุปกรณ์ต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ (3-9) แสดงโครงสร้างภายในของ L293D

จากโครงสร้างภายในจะเห็นว่ามี OP AMP ทำหน้าที่ในการขยายกระแสโดย IC L293d 1 ตัวจะสังเกตว่ามี OP AMP 2 ตัว ที่ทำหน้าที่ในการขยายกระแสเพื่อควบคุมทิศทางการหมุนของ MOTOR 2 ตัว และจะสังเกตุพบว่าจะมี DIODE ที่ทำหน้าที่ในการป้องกันแรงดันไหลย้อนกลับ เนื่องจาก MOTOR เป็นการใช้ขดลวดพันกัน เพื่อทำให้เกิดการเหนี่ยวนำทางไฟฟ้า (Inductor) ซึ่งในความเป็นจริง อาจเกิดแรงดันไหลย้อนกลับ (Electromagnetic force) ที่อาจก่อความเสียหาย แก่วงจรภายในได้ ดังนั้นจึงใช้ Diode ทำหน้าที่ในการป้องกันแรงดันไหลย้อนกลับ โดยใช้หลักการของ Diode เมื่อเกิดแรงดันที่ไหลย้อนกลับมาจาก MOTOR แรงดันส่วนนั้นจะถูก Detect จาก Diodeลงground เพื่อป้องกันความเสียหายที่อาจจะเกิดขึ้นกับวงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรโดยทั่ว ๆ ไปส่วนใหญ่ที่ใช้ MOTOR หรืออุปกรณ์ Inductor ควรจะคำนึงถึงแรงดันที่ไหลย้อนกลับ เพราะอาจทำความเสียหายแก่วงจรได้ เราอาจแก้ไขโดยใช้ Diode ทำการป้องกันได้ดังรูป เนื่องจากอุปกรณ์ Motor เป็นอุปกรณ์ที่ใช้หลักการเหนี่ยวนำจากขดลวดทองแดงกับแกนแม่เหล็กทำให้เกิดสนามไฟฟ้าซึ่งการทำงานในส่วนนี้อาจจะเกิดแรงดันไหลย้อนกลับมาเราจึงต้องทำการป้องกันเพื่อไม่ให้เกิดความเสียหาย

Inputs	Function	
Vinh=H	C=H D=L	turn Right
	C=L D=H	turn left
	C=D	Motor stop
Vinh=L	C=X D=X	Free running Motor stop

รูปที่ (3-10) ตารางแสดงสถาวะการควบคุม L293D

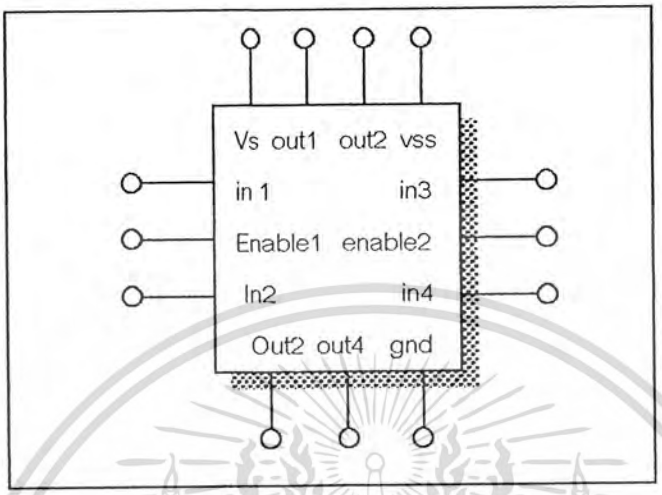
จากการแสดงการควบคุมนี้ จะเห็นว่าลักษณะการควบคุมไม่มีความซับซ้อนมากนัก เพียงแต่ทำการจ่ายแรงดัน (H=5V) หรือ (L=0V) ตามตารางที่บอกไว้ก็จะสามารถที่จะควบคุมทิศทางการหมุนของ MOTOR ได้ ในการใช้งานที่ต้องการกระแสที่ Output มากๆ อาจใช้ IC L293D ที่สามารถที่จะจ่ายกระแสได้มากกว่า 1.2 A เช่นเบอร์ L293DD ที่สามารถที่จะจ่ายกระแสไปยังโหลดได้มากกว่า แต่จะมีขาใช้งานมากกว่า คือจะเพิ่มขา Ground อีก 4 ขา

ส่วนการต่อใช้งานรวมกับ Motor จะมีลักษณะดังต่อไปนี้คือ

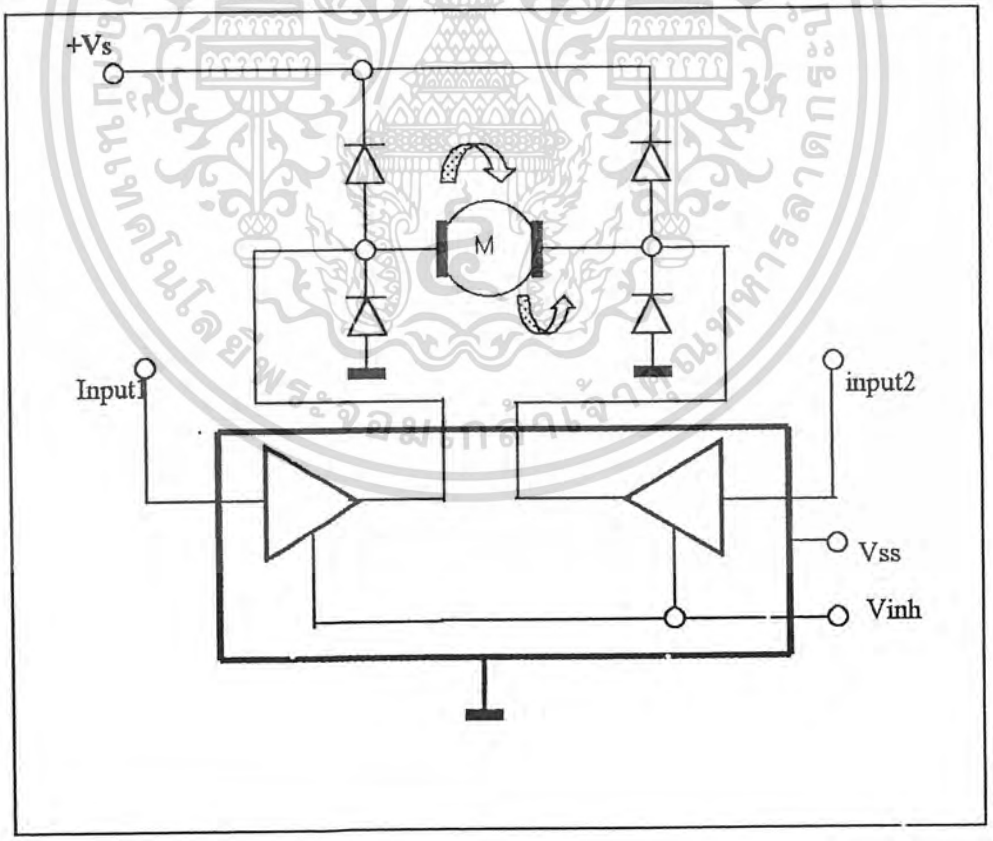
1. ทำการต่อ Out1 และ out2 ไปยัง Motor ตัวที่1
2. ป้อนแรงดันที่ Enable1 และ Enable2 =5v (vinh)
3. จ่ายแรงดันให้กับ Vsและ Vss

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ทำการควบคุมโดยอาศัยตารางคุณสมบัติการควบคุมโดยที่ C และ D เป็นINPUT 1และ2
5. ดูรูปที่ (3-12) ประกอบ



รูปที่ (3-11) แสดงขาต่างๆที่ต่อใช้งานกับ Motor

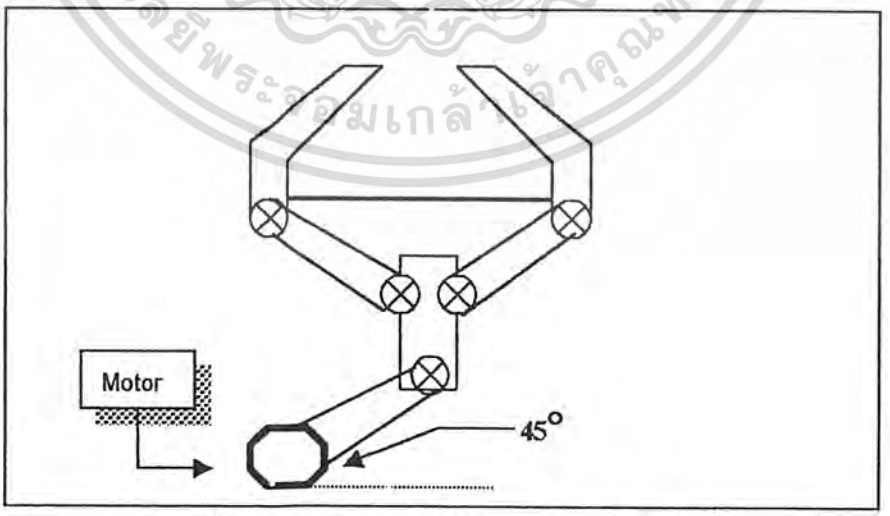


รูปที่ (3-12) เป็นลักษณะของการต่อใช้งาน L293D

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของหุ่นยนต์มด (ANT ROBOT) ส่วนสำคัญส่วนหนึ่งที่ต้องกล่าวถึงก็คือ กลไกที่ใช้ในการหนีบ เป็นส่วนหนึ่งที่ทำให้หุ่นยนต์มีความสมบูรณ์ เพราะต้องใช้ส่วนกลไกการหนีบทำหน้าที่ในการขนถ่ายอาหารไปไว้ที่รัง ดังนั้นส่วนที่ต้องมีความมั่นคงพอสมควรเพื่อที่จะหนีบอาหารไปไว้ที่รัง

กลไกการหนีบประกอบด้วยส่วนสำคัญหลายส่วนประกอบกัน โดยใช้จุดหมุนช่วยในการหนีบจับวัตถุของแกนแต่ละแกน สามารถที่จะแบ่งแกนได้หลัก ๆ คือ แกนที่ใช้ในการหนีบที่ประกอบด้วยด้านซ้ายและขวา 2 ด้าน แกนบังคับ แกนหนีบที่ติดต่ออยู่กับแกนส่วนกลาง และแกนส่วนกลางที่ทำหน้าที่ในการบังคับแกนหนีบทั้ง 2 ข้าง ที่ต่ออยู่กับแกนของ MOTOR แกนส่วนกลางนี้จะทำงานสัมพันธ์กับการหมุนของ MOTOR โดยเป็นสัดส่วนขององศาที่ชดเชยกัน คือ แกน MOTOR ถูกควบคุมให้สามารถหมุนได้ที่ 45 องศา ~ 90 องศา โดยประมาณ โดยในขณะที่แกน MOTOR ทำการหมุนมาที่ 45 องศา มีผลทำให้แกนส่วนกลางถูกดึงลงมาโดยแกนทั้งหมดถูกเชื่อมต่อถึงกันโดยให้มีจุดหมุนทั้งหมด 5 จุด เพื่อทำให้แกนทั้งหมดทำงานสัมพันธ์กัน เมื่อแกนส่วนกลางถูกดึงลงมาที่ 45 องศา โดยเทียบกับตำแหน่งการติดตั้ง MOTOR จะมีผลทำให้แกนทั้งหมดถูกดึงลงมาด้วย โดยทำงานสัมพันธ์กันได้ เนื่องจากการใช้จุดหมุน 5 จุด ทำการยึดแกนทั้งหมดเอาไว้ เมื่อแกนกลางถูกดึงลงมาที่ 45 องศา มีผลทำให้เกิดการหนีบวัตถุโดยจะมี Microcontroller ทำการควบคุมตำแหน่งองศาการหมุนของ MOTOR และการหนีบวัตถุค้างเอาไว้ไม่วางวัตถุจนกว่ามันจะไปที่รังจึงจะคลายแกนหนีบ และไปเก็บอาหารมาใหม่ ในส่วนของกลไกการหนีบนี้จะทำงานได้ในขณะที่หุ่นยนต์มดยังคงเดินอยู่ เมื่อหุ่นยนต์มดพบอาหารมันจะหนีบอาหาร

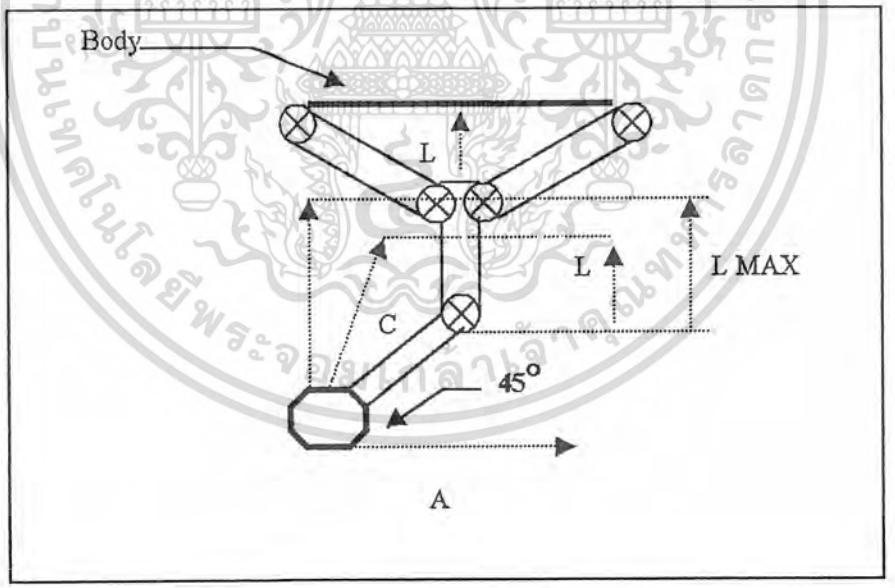


รูปที่ (3-13) แสดงการติดตั้งชุดกลไกการหนีบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปแสดงให้ทราบถึงตำแหน่งของการติดตั้ง MOTOR และแกนทั้ง 4 แกน โดยแกนหนีบ 2 แกน จะติดกับจุดหมุน 4 จุด และเชื่อมต่อกับแกนกลาง ซึ่งมีหน้าที่บังคับการหนีบและคลายวัตถุ และแกนกลางยึดกับแกนของ MOTOR โดยมีจุดหมุนอีก 1 จุด โดยแกนของ MOTOR มีลักษณะ Slope 45 องศา เมื่อเทียบกับตำแหน่งการติดตั้ง MOTOR

ส่วนลักษณะการคลายวัตถุของแกนหนีบ ก็คือ เมื่อแกนของ MOTOR หมุนมาที่ 45 องศา มันมีผลทำให้แกนทั้งหมดมีลักษณะที่หนีบวัตถุ ดังนั้น เมื่อ Microcontrollerสั่งให้คลายวัตถุแล้ว MOTOR จะทำการหมุนในทิศทางทวนเข็มนาฬิกา บังคับให้แกนของ MOTOR หมุนในองศาจากจุดเริ่มต้นที่ 45 องศา ไปที่ 70 องศา ≈ 90 องศา มีผลทำให้แกนกลางเลื่อนขึ้นข้างบนไปดันให้แกนหนีบวัตถุคลายวัตถุออก ทั้งนี้อัตราส่วนในการเคลื่อนที่ของแกนกลางจะไปสุดที่ขอบของ BODY โดยมีระยะทางที่เท่ากับองศาที่เปลี่ยนไป หรือถ้าหากว่า แกนกลางของ MOTOR ทำการหมุนองศาสูงสุดที่กำหนดไว้ที่ 90 องศา จุดหมุนที่ติดอยู่กับแกนกลาง 2 จุดจะทำให้ไม่เกิดปัญหาในการหนีบและคลาย ส่วนหลักการการคิดก็เพียงแค่อาศัยหลักการทางคณิตศาสตร์ เพียงเล็กน้อย คือ



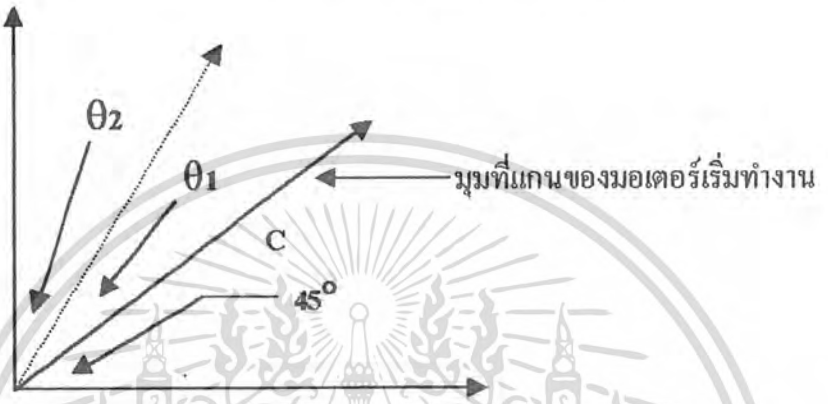
รูปที่ (3-14) แสดงการพิจารณามุมและค่าของระยะทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

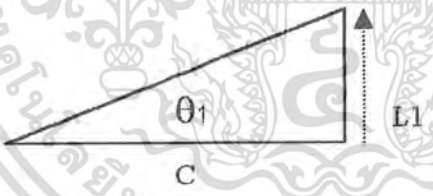
สมการแสดงความสัมพันธ์

เนื่องจากมุมเริ่มต้นที่ 45 องศา และมุมสูงสุดที่กำหนดไว้ 90 องศา

$\therefore$ 90 – 45 = 45 องศา ⟶ ทำให้จะทราบว่าทิศทางการเคลื่อนที่ของแกน MOTOR จะเป็นมุมแหลมในช่วง $\approx$ 45 องศาเมื่อมุมเริ่มต้นคือมุม 45°



ดังนั้นที่ L1 ระยะทางที่เพิ่มขึ้น L1 ต้องการทราบมุม



$$\text{Tan } \theta_1 = \frac{L1}{C} \quad \text{----------1}$$

และมุมที่เพิ่มขึ้นคือ θ_1 ต้องการทราบระยะทางที่เปลี่ยนไป

$$L1 = C \tan \theta_1$$

ดังนั้นระยะทางที่เพิ่มขึ้นจากการเปลี่ยนแปลงของมุมจะมีค่าเท่ากับระยะทางที่แกนกลางเคลื่อนที่ขึ้นหรือลง

เมื่อผลรวมของมุมที่กำหนด คือ 90 องศา

$$\therefore \quad 90^\circ = \theta_1+\theta_2+\theta_3+\ldots\ldots\ldots\ldots \theta_N+45^\circ$$

$$\theta_1+\theta_2+\theta_3+\ldots\ldots\ldots\ldots \theta_N = 90^\circ - 45^\circ$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสมการที่ 2 ก็คือมุมที่แกนของมอเตอร์จะเคลื่อนที่ได้จนสุดที่ 45°
ดังนั้นมุมที่เหลือคือมุมจากการเคลื่อนที่ของแกนกลางในขณะที่คลายวัตถุออกจนสุด

$$\text{Tan } \theta t = \frac{L}{C}$$

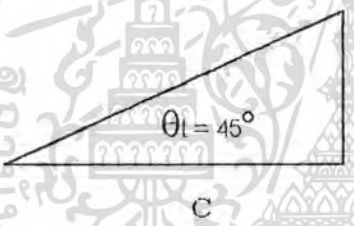
และจากสมการที่ 1 $\theta T = 45°$ แทนค่าจะได้

$$\text{Tan } 45° = \frac{L}{C}$$

$$1 = \frac{L}{C}$$

$$\therefore \quad L = C$$

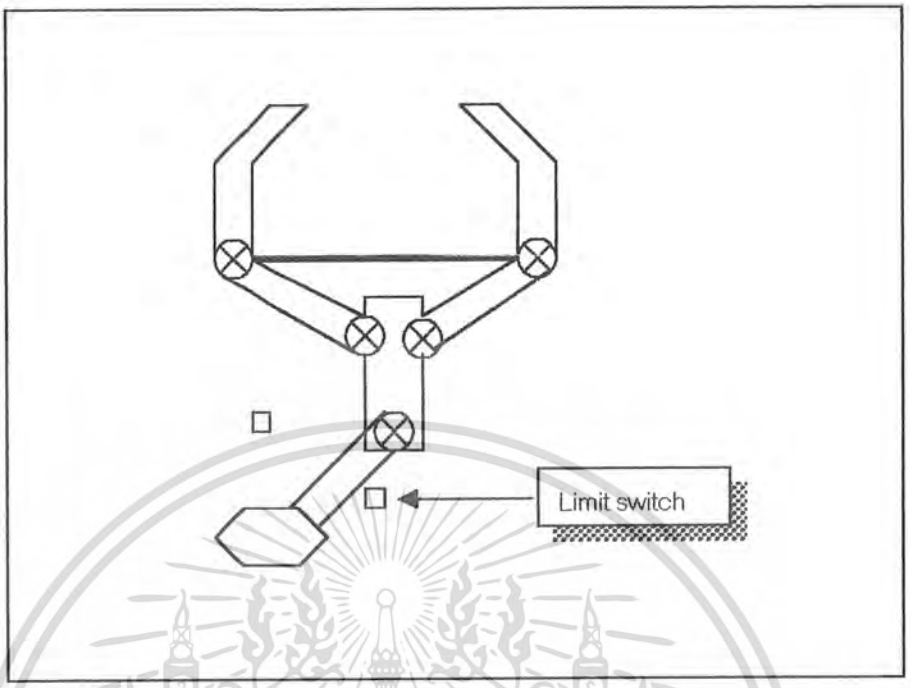
ดังนั้นจะได้ระยะทางการเคลื่อนที่ของแกนกลางในขณะที่คลายวัตถุออกจนสุด = C



ดังนั้นทำให้ทราบว่าในกรณีเดียวกันกับที่แกนหนีบทำการหนีบวัตถุแกนกลางก็จะเคลื่อนที่ เท่ากับระยะ C แต่ในการทำงานจริง ๆ แล้ว แกนของ MOTOR จะหมุนไม่ถึง 90 องศา เพราะเพียงแค่ 70 องศา ก็เพียงพอที่จะหนีบวัตถุแล้ว

ในการทำงานของชุดกลไกการหนีบอาหารนั้นเมื่อ MOTOR ทำการหมุนเพื่อต้องการให้มีการหนีบหรือคีบวัตถุ MOTOR จะทำการหมุนตลอดเวลา ถ้าเราไม่ป้องกันในส่วนนี้เอาไว้ อาจทำให้เฟืองภายในของ DC MOTOR เสียหายได้ ดังนั้นเราจึงต้องทำการป้องกันโดยอาจใช้การป้องกันทาง HARD WARE และการป้องกันทาง SOLF WARE มาช่วย ในการป้องกันแบบ HARD WARE ก็คือ เราอาจจะนำ MICRO SWITCH มาติดตรงองศาที่เราต้องการกำหนดให้ MOTOR หยุดทำงาน แล้วค่อยใช้ MICROCONTROLLER ควบคุม หรือเราอาจใช้การป้องกันทาง SOLF WARE โดยใช้การเขียน PROGRAM ป้องกัน กล่าวคือ เราทำการเขียน PROGRAM หน่วงเวลาให้ MOTOR ทำงานตามเวลาที่เรากำหนด แต่ในการใช้ PROGRAM ทำการป้องกันจะสะดวกกว่าการใช้ อุปกรณ์ทาง HARD WARE ป้องกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ (3-15) การใช้ LIMIT SWITCH ควบคุมองศาการหมุนของแกน MOTOR

หน่วยประมวลผล (MICROCONTROLLER)

ในการควบคุมหุ่นยนต์จะขาด MICROCONTROLLER ไม่ได้ เพราะเปรียบได้กับเป็นมันสมองที่ใช้ในการคิดและตัดสินใจตามที่เรากำหนดไว้โดย MICROCONTROLLER จะประกอบด้วยระบบของหน่วยความจำภายในที่ครบถ้วน กล่าวคือ มี ROM และ RAM ใน SINGLE SHIP ตัวเดียวทำให้มีความสะดวกในการใช้งาน อีกทั้ง PROGARM ที่ใช้ก็สามารถทำความเข้าใจได้ การใช้งาน MICROCONTROLLER จะต้องทำการวางแบบแผนของการเขียน PROGARM ให้ MICROCONTROLLER คิดและตัดสินใจตามที่เราต้องการอย่างถูกต้อง MICROCONTROLLER ที่ใช้ อยู่ในตระกูลของ MCS 51 แต่จะมี PORT และขาในการใช้งานน้อยกว่า MCS 51 เป็น MICROCONTROLLER ที่มีขาในการใช้งานจำนวน 20 ขา แบ่งเป็น PORT ต่างๆให้เลือกใช้งานได้ 2 PORT คือ PORT P1 และ PORT P3 และประกอบด้วยขาอื่นๆที่ใช้งานร่วมกันการใช้งานเหมือนกับการใช้งาน MICROCONTROLLER เบอร์อื่น ๆ ใช้ภาษา ASSAMBY ในการ PROGRAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติของ 89C2051

1. Flash memory มีขนาด 2 Kbytes
2. สามารถที่จะทำการโปรแกรมได้ถึง 1000 ครั้ง
3. ใช้ความถี่ 24 MHz
4. มีขนาด 8 bit สามารถที่จะติดต่อ RAM ภายในที่ 256 ตำแหน่งหรือ 128 Byte
5. Timer/ Counter มีขนาด 16 bit
6. สามารถที่จะต่อ LED ใช้งานทาง Output ได้เลย
7. ภายใน Chip มีวงจร Comparetor
8. ใช้กำลังงานต่ำ

รายละเอียดเกี่ยวกับขาการใช้งาน MICROCONTROLLER มีดังนี้

1. RST เป็นสัญญาณ RESET MICROCONTROLLER จะถูก RESET เมื่อสัญญาณที่ขานี้มีค่าเป็น ลอจิก 1
2. (RXD) P3.0 INPUT ของ PORT
3. (TXD) P3.1 OUTPUT ของ PORT
4. XTAL2 เป็น OUTPUT ของภาคขยายสัญญาณแบบ invert ของวงจรผลิตสัญญาณนาฬิกา
5. XTAL1 เป็น INPUT ของภาคขยายสัญญาณแบบ invert ของวงจรผลิตสัญญาณนาฬิกา
6. (INTO) P3.2 สัญญาณอินเตอร์รัพต์จากภายนอกตัวที่ 0
7. (INT1) P3.3 สัญญาณอินเตอร์รัพต์จากภายนอกตัวที่ 1
8. (TO) P3.4 อินพุทจาภายนอของตัวตั้งเวลา 0
9. (T1) P3.5 อินพุทภายนอกของตัวตั้งเวลา 1
10. GND
11. P3.7 สัญญาณการอ่านข้อมูลจากภายนอกเข้ามา
12. P1.0 (AINO)
13. P1.1 (AIN1)
14. P1.2
15. P1.3
16. P1.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

17. P1.5
18. P1.6
19. P1.7
20. VCC จ่ายแรงไฟ 5V

P1.0 ถึง P1.7 เป็นขาที่สามารถที่จะใช้เป็นอินพุทหรือสามารถที่จะกำหนดเป็นเอาต์พุทที่เป็นแบบ 2 ทิศทางมีขนาด 8 bit ที่มี Pull up อยู่ภายใน เมื่อเราต้องการกำหนดให้เป็นอินพุทเราต้องทำการ Set ค่าลอจิก 1 ส่งออกไปที่ port นี้ก่อนเพื่อทำให้เกิดการ Pull up ภายในและเมื่อมีสัญญาณอินพุทเป็น 0 เข้ามาจะทำให้ Port จ่ายกระแสออกไปนอกจากนี้แล้ว Port P1 ยังทำหน้าที่ในการรับตำแหน่งทางด้านต่ำในช่วงของการโปรแกรม EPROM อีกด้วย

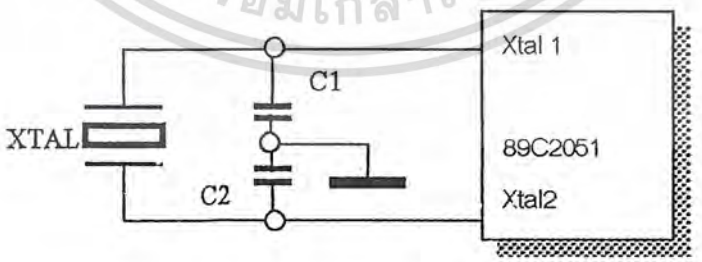
แสดงการต่อใช้งาน XTAL

XTAL เป็นส่วนที่ใช้ในการผลิตสัญญาณความถี่ให้กับ Microcontroller จุดประสงค์ก็คือต้องการให้ Microcontroller มีจังหวะในการทำงานที่ถูกต้องและสมบูรณ์

ในการต่อใช้งานขา XTAL1 และ XTAL2 เพื่อสร้างวงจรผลิตสัญญาณป้อนให้กับ Microcontroller ซึ่งลักษณะการใช้งานมี 2 ลักษณะคือ

1. การใช้คริสตอลผลิตความถี่ป้อนให้กับ Controller
2. การใช้สัญญาณความถี่จากภายนอกป้อนให้กับ Controller

รูปที่(3-16) ลักษณะการต่อใช้งาน XTAL



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

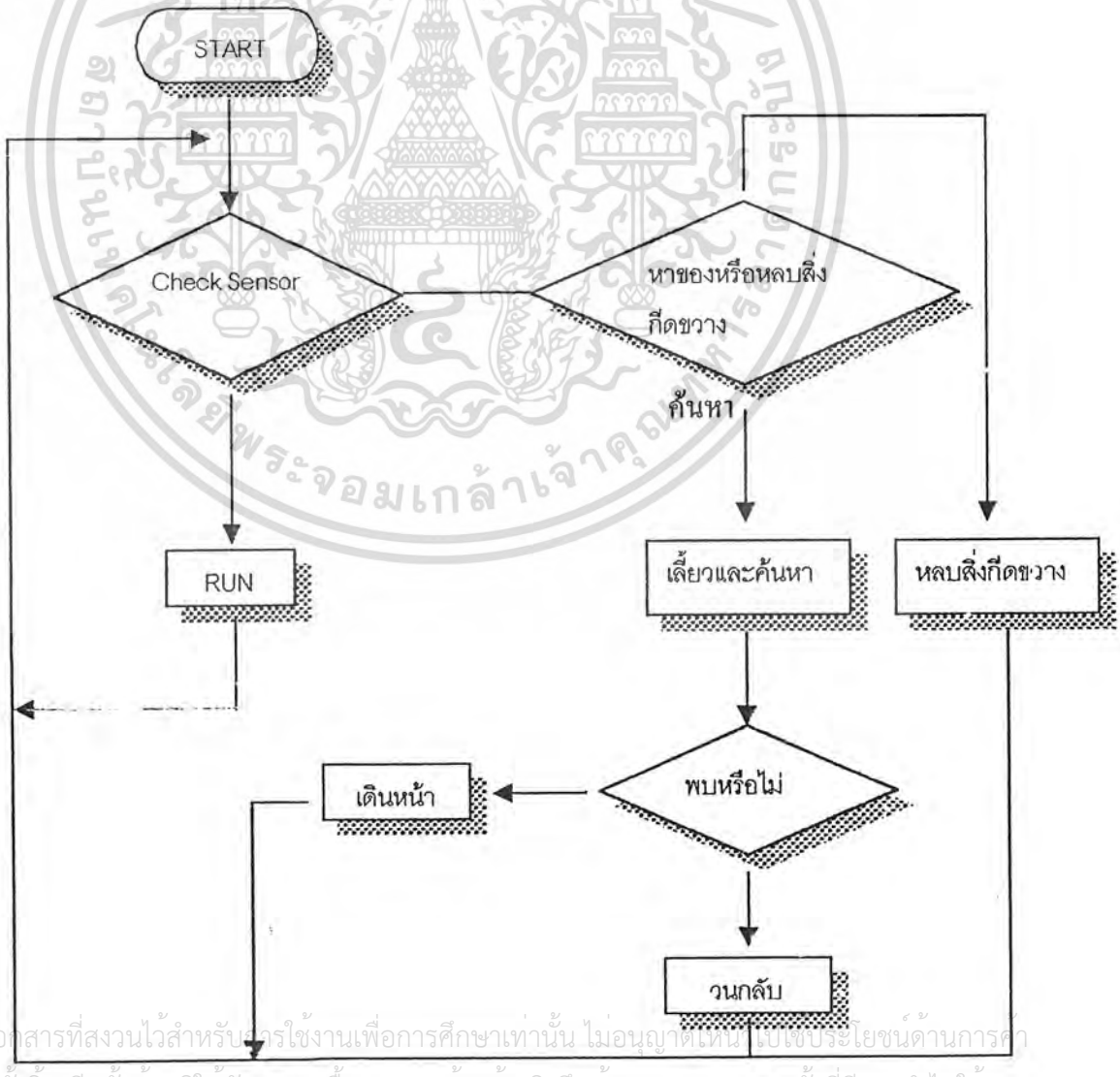
บทที่4

ลายละเอียดทางด้าน Solf ware

Solf Ware ก็คือชุดคำสั่งที่เราทำการกำหนดให้ Hard Ware ทำงานตามที่เราต้องการโดยชุดคำสั่งจะมีพื้นฐานมาจาก MCS 51 โดยทำการเขียนเป็นภาษา Assembly ก่อนที่เราจะทำการเขียนโปรแกรมเราต้องทำการกำหนดรูปแบบในการทำงานก่อนว่าต้องการให้มีลักษณะการทำงานอย่างไร แต่อย่างไรก็ตามในการเขียนโปรแกรมเราต้องมีพื้นฐานความรู้ทาง Hard Ware พอสมควรเนื่องจากเราจะต้องทำการอ้างถึงหน่วยความจำภายใน และรีจีสเตอร์ต่างๆ

กล่าวถึงการเขียน Flow Chart ในการทำงานของหุ่นยนต์มดจะสามารถที่จะแบ่งสภาวะการทำงานตาม Flow Chart ได้ดังนี้

ลักษณะที่ 1 การทำงานของหุ่นยนต์มดโดยทั่วไป

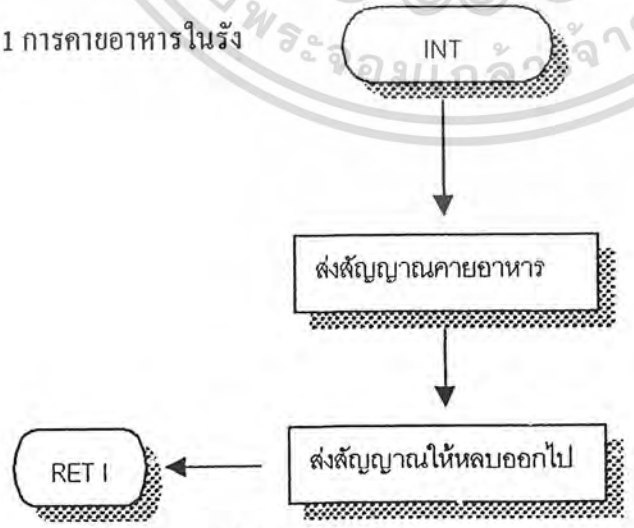


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะที่ 2 Flow Chart แสดงการคีบอาหาร

ลักษณะที่3 การนำอาหารไปที่รังมด

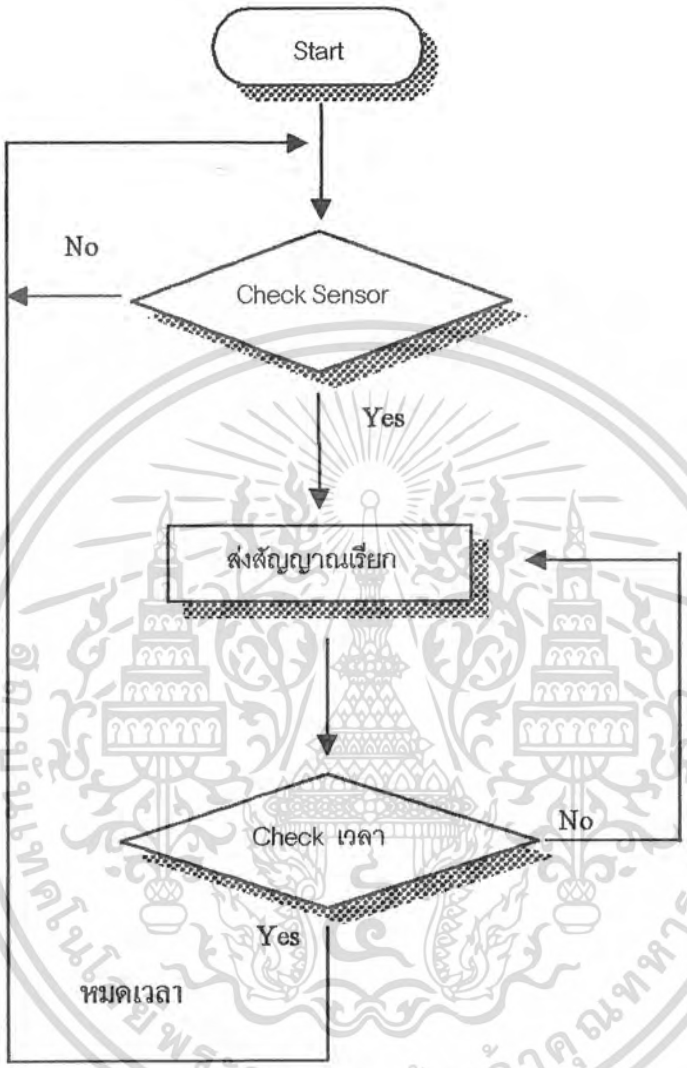
3.1 การคาขอาหารในรัง



3.2 การที่รังทำการส่งสัญญาณออกมาเรียกมด

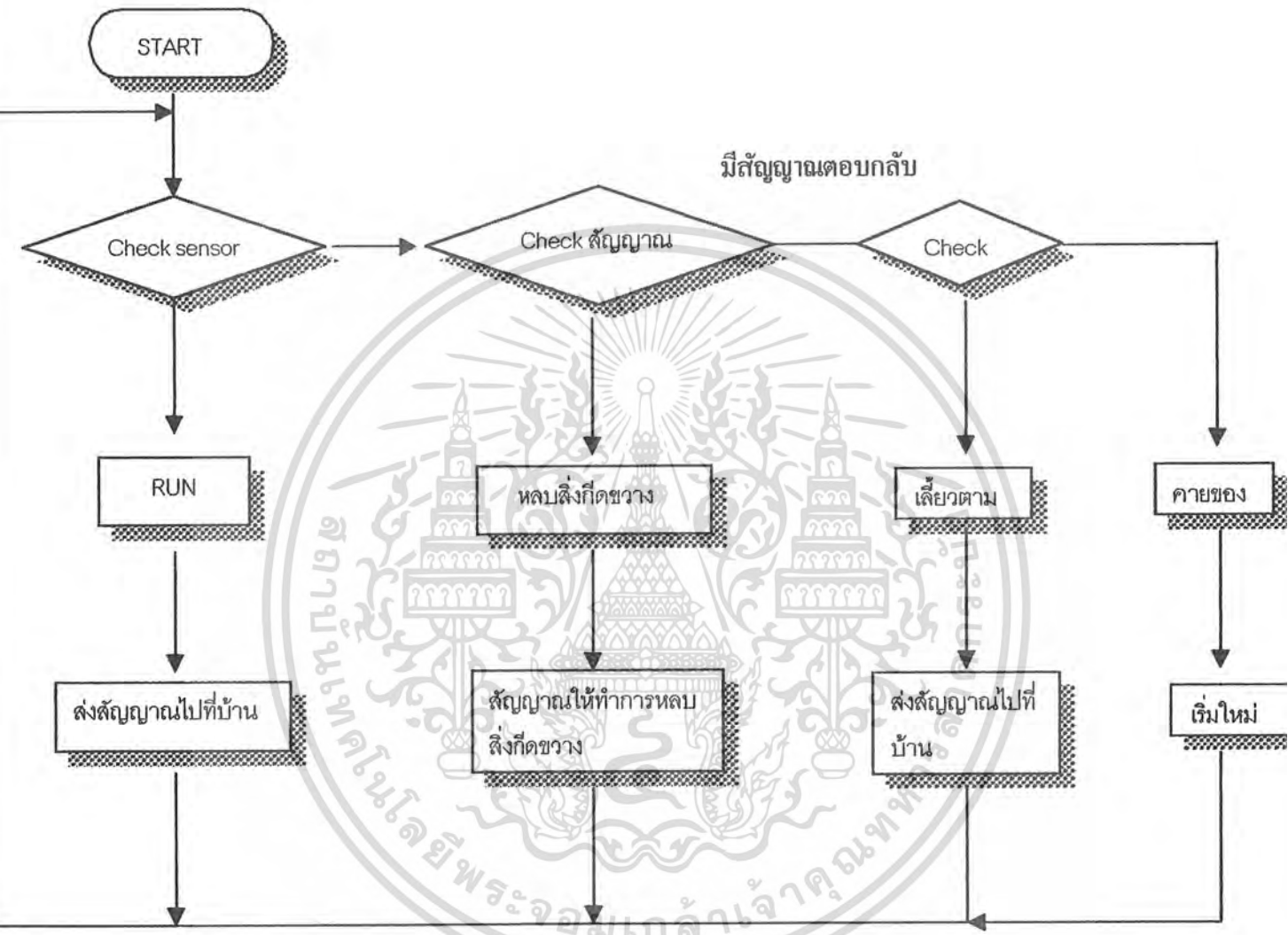
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การที่รังทำการส่งสัญญาณออกมาเรียกมด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะที่ 4 หลังจากที่หุ่นยนต์หนีบอาหารได้แล้วทำการส่งสัญญาณต่างๆออกไป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงโปรแกรมที่ใช้ในการปฏิบัติการของหุ่นยนต์มด

```
;BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
 SET_REG          EQU   31H
 RE_REG           EQU   32H
 DISTANCE_REG     EQU   35H

 X_THING1         EQU   40H
 X_THING2         EQU   41H
 FOUND1           EQU   42H
 FOUND2           EQU   43H
 FOUND3           EQU   44H
 FOUND4           EQU   45H
 FOUND5           EQU   46H
 LO30             EQU   47H
 LO50             EQU   48H
 LOOP_REV         EQU   4AH
 LOOP_B           EQU   4BH

 SUB_G1           EQU   4EH
 SUB_G2           EQU   4FH
 GRIP_REG         EQU   50H
 STOP_GRIP        EQU   51H
 RELEASE_OK       EQU   52H
 WAIT_RE          EQU   53H
 BUSY_REL_REG     EQU   54H

 XR0              EQU   55H
 XR1              EQU   56H
 TRAIN_SENT1      EQU   5AH
 TRAIN_SENT2      EQU   5BH
 CYCS_TIME        EQU   5CH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
RETURN_REG        EQU   60H
TIME_SENT_X       EQU   61H
BUSY2             EQU   65H
BUSY22            EQU   66H
BUSY222           EQU   67H
TURN_REG          EQU   68H
STEP_REG          EQU   69H
 LOOP_REVERT       EQU   10D
LOOP_AROUND       EQU   10D
LOOP_BUSY         EQU   30D


LOOP_G            EQU   03D
STOP_G            EQU   03D
HIGHT             EQU   06D
LLOW              EQU   02D

;*****************************************
     ORG    0000H
            AJMP   MAIN
;-----------------------------------------
            ORG    0003H
       CLR          EX0
                    AJMP  GRIP
;-----------------------------------------
              ORG    000BH
                RETI
;-----------------------------------------
              ORG    00013H
                   RETI
;-----------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                ORG     001BH
                  RETI
;-------------------------------------------------
                ORG     0023H
                  RETI
;-------------------------------------------------
  ORG     0030H
MAIN:           MOV   P1,#00000011B
                MOV   P3,#11001111B
                ACALL RELEASE_SUB
SETUP_MAIN:
  MOV           R1,#00H
  MOV           BUSY2,#002D
  MOV           BUSY22,#003D
  MOV           BUSY222,#010D
  MOV           TURN_REG,#003D
  MOV           DISTANCE_REG,#00D
  CLR   00H
  CLR   02H
  CLR   04H
  CLR   05H
  CLR   10H
  CLR   11H
  CLR   15H
  CLR   21H
  MOV           SET_REG,#030D
SET_UP: ACALL PULSE
  DJNZ  SET_REG,SET_UP
  SETB          PX0             ;IP.0
  SETB  IT0             ;TCON
  MOV           IE,#10000001B ;SET   /EA,EX0   OF   IE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    ;ENABLE     INT
                    ;INITIAL    EX0
  ACALLFORWARD
;************************************************

MAIN_START1:
  MOV   R3,#011D
CHECK11:        MOV   R2,#250D
CHECK1:         JNB     P3.1,FOUND_R
  JNB           P3.1,FOUND_R
  JNB           P3.0,FOUND_L
  JNB           P3.0,FOUND_L
  JNB           P3.3,FOUND_F
  JNB           P3.3,FOUND_F
  JNB           P1.1,FOUND_R_HEAD
  JNB           P1.1,FOUND_R_HEAD
  JNB           P1.0,FOUND_L_HEAD
  JNB           P1.0,FOUND_L_HEAD
  DJNZ          R2,CHECK1
  DJNZ          R3,CHECK11
  AJMP          RUN1
;-------------------------------------------------------
FOUND_R:        AJMP   SOMETHING_R
FOUND_L:        AJMP   SOMETHING_L
FOUND_F:        AJMP   SOMETHING_F
FOUND_R_HEAD:       AJMP   SOMETHING_R_HEAD
FOUND_L_HEAD:       AJMP   SOMETHING_L_HEAD


;************************************************************
;CHECK SIGNALS
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;*************************************************************
;-------------------------------------------------------
;SOMETHING_F                    OK
;-------------------------------------------------------
SOMETHING_F:
        MOV         FOUND1,#0100D
        MOV         FOUND2,#0195D
        MOV         LO30,#0195D
        MOV         LO50,#0130D
        MOV         FOUND3,#0195D
FF:     MOV         X_THING2,#003D
SF2:    MOV         X_THING1,#0250D
SF1:    JB          P3.3,FOUND_F1       ;(2)
        DJNZ        X_THING1,SF1;(2)
        DJNZ        X_THING2,SF2;(2)
        AJMP        ESCAPE_F        ; >3000uS
;+++++++++++++++++++++++++++++++++++++++++++++++++++++++
SHORT_F:            JNB     10H,GO_M1F
        JB          11H,GO_M2FL
        AJMP        CH2_R
GO_M2FL:            AJMP    CH2_L
GO_M1F:             AJMP    MAIN_START1
;+++++++++++++++++++++++++++++++++++++++++++++++++++++++
FOUND_F1:           JNB     P3.3,SHORT_F ;<400uS ,15CYCS
        DJNZ        FOUND1,FOUND_F1
FOUND_F2:           JNB     P3.3,OK_F120  ; 400-1170 us ,15-45 CYCS
        DJNZ        FOUND2,FOUND_F2
FOUND_F3:           JNB     P3.3,OK_F2    ; 1170-1950 us ,45-75 CYCS
        DJNZ        FOUND3,FOUND_F3
        AJMP        SHORT_F
OK_F120:            JB      P3.3,OK_F20
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        DJNZ   LO30,OK_F120 ; 20-30 CYCS
OK_F140:JB     P3.3,OK_F40
        DJNZ   LO50,OK_F140 ;40-50 CYCS
        AJMP   FF
;-----------------------------------------
OK_F20:         AJMP   SIGNAL_F1    ; GO
OK_F40:         AJMP   SIGNAL_F2    ; HOME
OK_F2:          AJMP   RELEASE_F    ; RELEASE
;-----------------------------------------------------
;SOMETHING_R_HEAD                   OK
;-----------------------------------------------------
SOMETHING_R_HEAD:
        MOV    FOUND1,#0100D
        MOV    LO30,#0195D
        MOV    LO50,#0130D
        MOV    FOUND2,#0195D
HR_HR:  MOV          X_THING2,#03D
S_HR2:  MOV          X_THING1,#0250D
S_HR1:  JB     P1.1,FOUND_HR1      ;(2)
        DJNZ   X_THING1,S_HR1      ;(2)
        DJNZ   X_THING2,S_HR2      ;(2)
        AJMP   ESCAPE_HR
;++++++++++++++++++++++++++++++++++++++++++++++
SHORT_HR:       JNB    10H,GO_M1HR
        JB     11H,GO_M2HRL
        AJMP   CH2_R
GO_M2HRL:       AJMP   CH2_L
GO_M1HR:        AJMP   MAIN_START1
;++++++++++++++++++++++++++++++++++++++++++++++
FOUND_HR1:      JNB    P1.1,SHORT_HR
        DJNZ   FOUND1,FOUND_HR1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
FOUND_HR2:   JNB    P1.1,OK_HR120
             DJNZ   FOUND2,FOUND_HR2
             AJMP   SHORT_HR
OK_HR120:JB  P1.1,OK_HR20
             DJNZ   LO30,OK_HR120        ; 20-30 CYCS
OK_HR140:JB  P1.1,OK_HR40
             DJNZ   LO50,OK_HR140        ;40-50 CYCS
             AJMP   HR_HR
;-------------------------------------------------------
OK_HR20:AJMP SIGNAL_HR1  ; GO
OK_HR40:AJMP SIGNAL_HR2  ; HOME
;-------------------------------------------------------
;SOMETHING_L_HEAD                OK
;-------------------------------------------------------
SOMETHING_L_HEAD:
             MOV    FOUND1,#0100D
             MOV    LO30,#0195D
             MOV    LO50,#0130D
             MOV    FOUND2,#0195D
HL_HL:  MOV          X_THING2,#03D
S_HL2:  MOV          X_THING1,#0250D
S_HL1:  JB     P1.0,FOUND_HL1    ;(2)
             DJNZ   X_THING1,S_HL1      ;(2)
             DJNZ   X_THING2,S_HL2      ;(2)
             AJMP   ESCAPE_HL
;++++++++++++++++++++++++++++++++++++++++
SHORT_HL:JNB 10H,GO_M1HL
             JB11H,GO_M2HLL
             AJMP   CH2_R
GO_M2HLL:    AJMP   CH2_L
GO_M1HL:     AJMP   MAIN_START1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;++++++++++++++++++++++++++++++++++++++++
FOUND_HL1:   JNB    P1.0,SHORT_HL
             DJNZ   FOUND1,FOUND_HL1
FOUND_HL2:   JNB    P1.0,OK_HL120
             DJNZ   FOUND2,FOUND_HL2
             AJMP   SHORT_HL
OK_HL120:JB  P1.0,OK_HL20
             DJNZ   LO30,OK_HL120        ; 20-30 CYCS
OK_HL140:JB  P1.0,OK_HL40
             DJNZ   LO50,OK_HL140        ;40-50 CYCS
             AJMP   HL_HL
;-----------------------------------------------------
OK_HL20:     AJMP   SIGNAL_HL1  ; GO
OK_HL40:     AJMP   SIGNAL_HL2  ; HOME
;-----------------------------------------------------
;SOMETHING_R                OK
;-----------------------------------------------
SOMETHING_R: MOV    FOUND1,#0100D
             MOV    LO30,#0195D
             MOV    LO50,#0130D
             MOV    FOUND2,#0195D
R_R:         MOV          X_THING2,#03D
SR2:         MOV          X_THING1,#0250D
SR1:         JB     P3.1,FOUND_R1        ;(2)
             DJNZ   X_THING1,SR1;(2)
             DJNZ   X_THING2,SR2;(2)
             JNB    15H,NO_LOPT_R2
             AJMP   L2RL
NO_LOPT_R2:  AJMP   ESCAPE_R     ; >3000 uS
SHORT_R:     JNB    10H,GO_M1R
             JB     15H,P_LOPT_R2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 JB              11H,GO_M2RL
 AJMP            CH2_R
GO_M2RL:         AJMP   CH2_L
P_LOPT_R2:       AJMP   PASS_LOPT_R2
GO_M1R:          AJMP   MAIN_START1
FOUND_R1:        JNB    P3.1,SHORT_R
          DJNZ   FOUND1,FOUND_R1
FOUND_R2:        JNB    P3.1,OK_R120
          DJNZ   FOUND2,FOUND_R2
          AJMP   SHORT_R
OK_R120:         JB     P3.1,OK_R20
          DJNZ   LO30,OK_R120; 20-30 CYCS
OK_R140:         JB     P3.1,OK_R40
          DJNZ   LO50,OK_R140;40-50 CYCS
          AJMP   R_R
;-------------------------------------------------------
OK_R20:          AJMP   SIGNAL_R1     ; GO
OK_R40:          AJMP   SIGNAL_R2     ; HOME
;-------------------------------------------------------
;SOMETHING_L          OK
;-------------------------------------------------------
SOMETHING_L: MOV    FOUND1,#0100D
          MOV    LO30,#0195D
          MOV    LO50,#0130D
          MOV    FOUND2,#0195D
L_L:      MOV    X_THING2,#03D
SL2:      MOV    X_THING1,#0250D
SL1:      JB     P3.0,FOUND_L1        ;(2)
          DJNZ   X_THING1,SL1;(2)
          DJNZ   X_THING2,SL2;(2)
          JNB    15H,NO_LOPT_L2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        AJMP   L2LR
NO_LOPT_L2:  AJMP   ESCAPE_L     ;>0.003S
SHORT_L:     JNB    10H,GO_M1L
        JB     15H,P_LOPT_L2
        JB     11H,GO_M2LL
        AJMP   CH2_R
GO_M2LL:     AJMP   CH2_L
P_LOPT_L2:   AJMP   PASS_LOPT_L2
GO_M1L:             AJMP   MAIN_START1
FOUND_L1:    JNB    P3.0,SHORT_L
        DJNZ   FOUND1,FOUND_L1
FOUND_L2:    JNB    P3.0,OK_L120
        DJNZ   FOUND2,FOUND_L2
        AJMP   SHORT_L
OK_L120:            JB     P3.0,OK_L20
        DJNZ   LO30,OK_L120 ; 20-30 CYCS
OK_L140:            JB     P3.0,OK_L40
        DJNZ   LO50,OK_L140 ;40-50 CYCS
        AJMP   L_L
;-------------------------------------------------------
OK_L20:      AJMP   SIGNAL_L1    ; GO
OK_L40:      AJMP   SIGNAL_L2    ; HOME
;-------------------------------------------------------
;*********************************************************
;OPERRATION   TABLE1
;*********************************************************
SIGNAL_R1:   MOV    STEP_REG,#07D
STEP_R:      ACALL TR
        DJNZ   STEP_REG,STEP_R
        MOV    P1,#00000011B
        AJMP   MAIN_START1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SIGNAL_R2:     JNB    02H,DONT_R2
               AJMP   SIGNAL_R1
DONT_R2:       MOV    STEP_REG,#10D
STEP_R_PULSE:  MOV    P1,#00000011B
               ACALL  PULSE
               DJNZ   STEP_REG,STEP_R_PULSE
               AJMP   MAIN_START1
;++++++++++++++++++++++++++++++++++++++++
ESCAPE_R:      CLR    00H
               JB     10H,NO_FDR
               JB     02H,PASS_FDR
               JB     04H,ESP_PASS_FDR
NO_FDR:        AJMP   MAIN_START2
ESP_PASS_FDR:  MOV    DISTANCE_REG,#12D
PASS_FDR:      INC    R1
               CJNE   R1,#LOOP_BUSY,NEXT_RIGHT
               ACALL  BACK
               AJMP   MAIN_START1
NEXT_RIGHT :   ACALL  TL
               AJMP   MAIN_START1
;----------------------------------------
SIGNAL_L1:     MOV    STEP_REG,#07D
STEP_L:        ACALL  TL
               DJNZ   STEP_REG,STEP_L
               MOV    P1,#00000011B
               AJMP   MAIN_START1
SIGNAL_L2:     JNB    02H,DONT_L2
               AJMP   SIGNAL_L1
DONT_L2:       MOV    STEP_REG,#010D
STEP_L_PULSE:  MOV    P1,#00000011B
               ACALL  PULSE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        DJNZ  STEP_REG,STEP_L_PULSE
        AJMP  MAIN_START1
;++++++++++++++++++++++++++++++++++++++++
ESCAPE_L:     SETB  00H
        JB    10H,NO_FDL
        JB    02H,PASS_FDL
        JB    05H,ESP_PASS_FDL
NO_FDL:       AJMP  MAIN_START2
ESP_PASS_FDL: MOV   DISTANCE_REG,#12D
PASS_FDL:     INC           R1
        CJNE          R1,#LOOP_BUSY,NEXT_LEFT
        ACALL         BACK
        AJMP  MAIN_START1
NEXT_LEFT :   ACALL         TR
        AJMP  MAIN_START1
;----------------------------------------
ESCAPE_F:     ACALLBACK
        AJMP  MAIN_START1
SIGNAL_F1:    MOV   STEP_REG,#07D
STEP_F_F:     ACALLFORWARD
        DJNZ  STEP_REG,STEP_F_F
        MOV   P1,#00000011B
        AJMP  MAIN_START1
SIGNAL_F2:    JNB   02H,DONT_F2
        AJMP  SIGNAL_F1
DONT_F2:      MOV   STEP_REG,#010D
STEP_F_PULSE: MOV   P1,#00000011B
        ACALLPULSE
        DJNZ  STEP_REG,STEP_F_PULSE
        AJMP  MAIN_START1
RELEASE_F:    JB    P3.2,NO_RELEASE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        AJMP   RELEASE
NO_RELEASE:  MOV   P1,#00000011B
        AJMP   SHORT_F
;-------------------------------------------------
ESCAPE_HR:    AJMP   SIGNAL_HL1
SIGNAL_HR1:   MOV    STEP_REG,#07D
STEP_HR_HR:   ACALLBL
        DJNZ   STEP_REG,STEP_HR_HR
        MOV    P1,#00000011B
        AJMP   MAIN_START1
SIGNAL_HR2:   JNB    02H,DONT_HR2
        AJMP   SIGNAL_HR1
DONT_HR2:     MOV    STEP_REG,#010D
STEP_HR_PULSE:      MOV    P1,#00000011B
        ACALLPULSE
        DJNZ   STEP_REG,STEP_HR_PULSE
        AJMP   MAIN_START1
;-------------------------------------------------
ESCAPE_HL:    AJMP   SIGNAL_HR1
SIGNAL_HL1:   MOV    STEP_REG,#07D
STEP_HL_HL:   ACALLBR
        DJNZ   STEP_REG,STEP_HL_HL
        MOV    P1,#00000011B
        AJMP   MAIN_START1
SIGNAL_HL2:   JNB    02H,DONT_HL2
        AJMP   SIGNAL_HL1
DONT_HL2:     MOV    STEP_REG,#010D
STEP_HL_PULSE:      MOV    P1,#00000011B
        ACALLPULSE
        DJNZ   STEP_REG,STEP_HL_PULSE
        AJMP   MAIN_START1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;-----------------------------------------
RUN1:           JNB     02H,YES
        MOV     SUB_G2,#0100D
RUN_GRIP22T:    MOV     SUB_G1,#0250D
RUN_GRIP2T:     JB      P3.2,RUN_NO_INT0
        DJNZ    SUB_G1,RUN_GRIP2T
        DJNZ    SUB_G2,RUN_GRIP22T
        SETB    P1.3
        ACALL   PULSE2
        CLR     P1.2
        CLR     P1.3
        AJMP    YES
RUN_NO_INT0:    ACALL   RELEASE_SUB
        MOV             IE,#10000001B
        CLR     02H
YES:            MOV             R1,#00H
        CLR     10H
        ACALL   FORWARD
        MOV     R0,DISTANCE_REG
        CJNE    R0,#00H,RUN_ESP
RUN_NORMAL:     CLR     04H
        CLR     05H
        MOV     DISTANCE_REG,#00D
        AJMP    MAIN_START1
RUN_ESP:        MOV     DISTANCE_REG,R0
        DJNZ    DISTANCE_REG,ESP_MODE
        AJMP    RUN_NORMAL
ESP_MODE:       AJMP    MAIN_START1
;-----------------------------------------
BACK:     MOV     LOOP_REV,#LOOP_REVERT
   MOV   LOOP_B,#LOOP_AROUND
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV   GRIP_REG,#LOOP_G
        MOV   STOP_GRIP,#STOP_G
        MOV   P1,#00000011B
              ACALL       FORWARD
GRIP_SUB:           SETB  P1.3
LOOP_GRIP:          ACALL   PULSE              ;  0.1 SEC
                    DJNZ          GRIP_REG,LOOP_GRIP
                          ; = 0.1 SEC * GRIP_REG
CHECK_GRIP:         ACALL   PULSE              ;  0.1 SEC
        JNB         P3.7,GRIP_PASS
        DJNZ        STOP_GRIP,CHECK_GRIP
                          ;  0.1 SEC *GRIP_REG
                                                   ;  + 0.1 SEC
*STOP_GRIP
        MOV   P1,#00000011B
        JNB   P3.2,GRIP_PASS
NO_REL:       ACALLRELEASE_SUB
        AJMP  NO_INT0
GRIP_PASS:    MOV   P1,#00000011B
        SETB  08H
        MOV    LOOP_REV,#LOOP_REVERT
GRIP_B:       ACALL   REVERT
                    DJNZ    LOOP_REV,GRIP_B
        JB    P3.2,NO_REL
        MOV   RETURN_REG,#040D
        JB    00H,RETURN_BL
RETURN_BR:    ACALLBR
        DJNZ  RETURN_REG,RETURN_BR
        AJMP  GOHOME
RETURN_BL:    ACALLBL
        DJNZ  RETURN_REG,RETURN_BL
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
GOHOME:     MOV   P1,#00000011B
            CLR         P1.3
            CLR   P1.2
            ACALL SENT_X
            DJNZ  R4,GOHOME   ;
            SETB      02H               ;GRIPED
            CLR   08H
            CLR   10H
            CLR   04H
            CLR   05H
            MOV         IE,#00H
            RETI

;*************************************************************
RELEASE:          ACALLFORWARD
            ACALLFORWARD
            ACALLRELEASE_SUB
            MOV   RELEASE_OK,#15D
            MOV   BUSY_REL_REG,#03D ;!
NEXT_RE :   ACALLREVERT
            JB    P3.2,NORMAL
            DJNZ  RELEASE_OK,NEXT_RE
 NORMAL:          MOV   WAIT_RE,#10D
WAIT:       ACALLREVERT                   ; 1 SEC
            DJNZ  WAIT_RE,WAIT
            MOV   RETURN_REG,#037D
            JB    00H,REL_RET_BL
REL_RET_BR: ACALLBR
            DJNZ  RETURN_REG,REL_RET_BR
            AJMP  LEAVEHOME
REL_RET_BL: ACALLBL
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        DJNZ  RETURN_REG,REL_RET_BL
LEAVEHOME:  MOV  P1,#00000011B
        CLR   02H
        AJMP  SETUP_MAIN
;-------------------------------------------------------------------
RELEASE_SUB: MOV  GRIP_REG,#LOOP_G
        MOV   STOP_GRIP,#STOP_G
              SETB  P1.2
        CLR   P1.3
LOOP_RE:      ACALL      PULSE                ;  0.1 SEC
        DJNZ        GRIP_REG,LOOP_RE  ; = 0.1 SEC * GRIP_REG
CHECK_RELEASE:      ACALL      PULSE                ;  0.1 SEC
        JNB         P3.7,RELEASE_PASS
        DJNZ        STOP_GRIP,CHECK_RELEASE      ; 0.1 SEC
                                ;  + 0.1 SEC * GRIP_REG
                                                ;      +  0.1  SEC
*STOP_GRIP
RELEASE_PASS :      CLR         P1.2
        CLR   P1.3
        RET
;*************************************************
;MAIN2
;*************************************************
MAIN_START2:
  JB      10H,MAIN_OPER2
  JB      00H,L_MAIN2
  AJMP  R_MAIN2
L_MAIN2:
  SETB  11H
TURN_R2:
  ACALL TL                    ;step1 seek
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  ACALL BR                    ;step2 seek
  ACALL TL                    ;step3 seek
  DJNZ  TURN_REG,TURN_R2
  MOV   TURN_REG,#03D
  SETB  10H
CH2_L:
  MOV   R3,#011D
CHECK2L:
  MOV   R2,#250D
CHECK2LL:
  JNB   P3.3,FOUND_FF2
  JNB   P3.3,FOUND_FF2
  JNB       P1.1,FOUND_R_HEAD2
  JNB       P1.1,FOUND_R_HEAD2
  JNB       P1.0,FOUND_L_HEAD2
  JNB       P1.0,FOUND_L_HEAD2
  JNB       P3.1,FOUND_RR2
  JNB       P3.1,FOUND_RR2
  JNB       P3.0,FOUND_LL2
  JNB       P3.0,FOUND_LL2
  DJNZ  R2,CHECK2LL
  DJNZ  R3,CHECK2L
  AJMP  RUN2
;-------------------------------------------------------------
MAIN_OPER2:   AJMP  OPERATE2
FOUND_FF2:    AJMP  SOMETHING_F
FOUND_R_HEAD2:      AJMP  SOMETHING_R_HEAD
FOUND_L_HEAD2:      AJMP  SOMETHING_L_HEAD
FOUND_RR2:    AJMP  SOMETHING_R
FOUND_LL2:    AJMP  SOMETHING_L
;-------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
R_MAIN2:
  CLR   11H
TURN_L2:        ACALL TR                        ;step1 seek
  ACALL BL                      ;step2 seek
  ACALL TR                      ;step3 seek
  DJNZ  TURN_REG,TURN_L2
  MOV   TURN_REG,#03D
  SETB  10H
CH2_R:
  MOV   R3,#011D
CHECK2R:
  MOV   R2,#250D
CHECK2RR:
  JNB   P3.3,FOUND_FF2
  JNB   P3.3,FOUND_FF2
  JNB           P1.1,FOUND_R_HEAD2
  JNB           P1.1,FOUND_R_HEAD2
  JNB           P1.0,FOUND_L_HEAD2
  JNB           P1.0,FOUND_L_HEAD2
  JNB           P3.0,FOUND_LL2
  JNB           P3.0,FOUND_LL2
  JNB           P3.1,FOUND_RR2
  JNB           P3.1,FOUND_RR2
  DJNZ  R2,CHECK2RR
  DJNZ  R3,CHECK2R
  AJMP  RUN2
;*************************************************************
;OPERRATION  TABLE2
;*************************************************************
OPERATE2:       JNB     11H,OPT_R2
                AJMP    OPT_L2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;-------------------------------------------------------
OPT_R2:         JB      00H,RL_MAIN2
RR_MAIN2:       AJMP    R_MAIN2
RL_MAIN2:       DJNZ    BUSY2,OPT2RL
LOOP2RL:        SETB    15H
                CLR     21H
                AJMP    SOMETHING_R
L2RL:           ACALL BR
                AJMP    LOOP2RL
PASS_LOPT_R2: ACALL BR
                MOV     BUSY2,#02D
                SETB    04H
                CLR     10H
                CLR     15H
                CLR     21H
                AJMP    MAIN_START1
OPT2RL:         ACALL BR
                AJMP    CH2_R
;-------------------------------------------------------
OPT_L2:         JNB     00H,LR_MAIN2
LL_MAIN2:       AJMP    L_MAIN2
LR_MAIN2:       DJNZ    BUSY2,OPT2LR
LOOP2LR:        SETB    15H
                SETB    21H
                AJMP    SOMETHING_L
L2LR:           ACALL BL
                AJMP    LOOP2LR
PASS_LOPT_L2: ACALL BL
                MOV     BUSY2,#02D
                SETB    05H
                CLR     10H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        CLR   15H
        CLR   21H
        AJMP  MAIN_START1
OPT2LR:             ACALLBL
        AJMP  CH2_L
;-----------------------------------------
RUN2:           DJNZ  BUSY22,RUNRUN2
        MOV   BUSY22,#02D
        AJMP  MAIN_START1
RUNRUN2:        ACALLFORWARD
        DJNZ  BUSY222,RUNRUN2
        MOV   BUSY222,#02D
        JB    11H,GO_L_MAIN2
        AJMP  CH2_R
GO_L_MAIN2:  AJMP  CH2_L
;*********************************************
;SUB.
;-----------------------------------
;FORWARD
;-----------------------------------
FORWARD:
 MOV     P1,#10010011B
 JB      02H,G2TF
 ACALL   PULSE
 RET
G2TF:   ACALL   PULSE2
 ACALLSENT_X
 ACALL   PULSE2
 RET
;-----------------------------------
;LEFT
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;-----------------------------------------
TL:              MOV   P1,#01010011B
  JB      02H,G2TL
  ACALL    PULSE
  RET
G2TL:   ACALL    PULSE2
  ACALL SENT_X
  ACALL    PULSE2
  RET
;-----------------------------------------
;RIGHT
;-----------------------------------------
TR:      MOV          P1,#10100011B
  JB      02H,G2TR
  ACALL        PULSE
  RET
G2TR:   ACALL        PULSE2
  ACALL SENT_X
  ACALL        PULSE2
  RET
;-----------------------------------------
;REVERT
;-----------------------------------------
REVERT:
  MOV   P1,#01100011B
  ACALL SENT1
  ACALL        PULSE2
               RET

;-----------------------------------------------------------------
;BACK-RIGHT                              OK.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;-------------------------------------------------------------------
BR:     JB      08H,BR_AF_G
  MOV   P1,#01000011B
        ACALL       PULSE
  MOV   P1,#01010011B
        ACALL       SENT1
  ACALL       PULSE2
  MOV   P1,#01000011B
  ACALL       PULSE
  RET
BR_AF_G:MOV   P1,#01000011B
  ACALL SENT_X
  ACALL    PULSE2
  MOV   P1,#01010011B
  ACALL SENT_X
  ACALL    PULSE2
  MOV   P1,#01000011B
  ACALL SENT_X
  ACALL    PULSE2
        RET
;-------------------------------------------------------------------
;BACK-LEFT                                          OK.
;-------------------------------------------------------------------
BL:     JB      08H,BL_AF_G
  MOV   P1,#00100011B
        ACALL       PULSE
  MOV   P1,#10100011B
        ACALL       SENT1
  ACALL    PULSE2
  MOV   P1,#00100011B
        ACALL    PULSE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  RET
BL_AF_G: MOV  P1,#00100011B
  ACALL SENT_X
  ACALL     PULSE2
  MOV   P1,#10100011B
  ACALL SENT_X
  ACALL     PULSE2
  MOV   P1,#00100011B
  ACALL SENT_X
  ACALL     PULSE2
  RET
;*****************************************************************************
SENT1:  MOV   TRAIN_SENT2,#11D
PULSE_TRAIN1:
  MOV   TRAIN_SENT1,#250D
PULSE_TRAIN0:
  SETB  P3.5
  NOP
  MOV   XR0,#06D
HI0:      DJNZ      XR0,HI0
  CLR   P3.5
  NOP
  MOV   XR1,#02D
LO0:      DJNZ            XR1,LO0
TRAIN_ON0:
  DJNZ  TRAIN_SENT1,PULSE_TRAIN0
  DJNZ  TRAIN_SENT2,PULSE_TRAIN1
; [1 +1+2 + 2 *(XR0) ] + [ 1 +1+ 2 + 2 *(XR1) + (2) ] * (TRAIN_SENT1)
;= [    H (16 )     +      L (10 )     ]   * (TRAIN_SENT1)
  RET
;-------------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;SENT_X
;-----------------------------------------------------------------
SENT_X: MOV   TIME_SENT_X,#03D
SENT:   ACALL PULSE_HI          ;RECEIVED  0(2)
  ACALL PULSE_LO          ;RECEIVED  1(2)
  ACALL PULSE_HI          ;RECEIVED  0(2)
  ACALL PULSE_LO          ;RECEIVED  1(2)
  ACALL PULSE_HI          ;RECEIVED  0(2)
  ACALL PULSE_LO          ;RECEIVED  1(2)
  ACALL PULSE_HI          ;RECEIVED  0(2)
  ACALL PULSE2
  DJNZ  TIME_SENT_X,SENT  ;(4200+20000)*2=75000
  RET
;-----------------------------------------------------------------
;PULSE  TRAIN
;-----------------------------------------------------------------
PULSE_HI:
  MOV   TRAIN_SENT1,#020D
PULSE_TRAIN:
  SETB  P3.5              ;(1)
  NOP
  MOV   XR0,#HIGHT        ;(2)
HI:     DJNZ    XR0,HI            ;(2)
  CLR   P3.5              ;(1)
  NOP
  MOV   XR1,#LLOW         ;(2)
LO:     DJNZ          XR1,LO            ;(2)
TRAIN_ON:
  DJNZ  TRAIN_SENT1,PULSE_TRAIN ;(2)
; [1 + 1+2 + 2 *(6) ] + [ 1 + 1+ 2 + 2 *(2) + (2) ]  * (TRAIN_SENT1)
; =  [    H (16 )     +     L (10 )     ]   * (TRAIN_SENT1)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  RET                    ;(2)
;-------------------------------------------------
PULSE_LO:
  CLR   P3.5             ;(1)
  MOV   TRAIN_SENT2,#030D ;(2)
P_CYCLES:
  MOV   CYCS_TIME,#011D          ;(2)
CYCS:  DJNZ  CYCS_TIME,CYCS      ;(2)
  DJNZ  TRAIN_SENT2,P_CYCLES     ;(2)
; { (2)+ (1)}+
; [(2)+(2)*(CYCS_TIME) +(2)] * (TRAIN_SENT2)} +{ (2) }
;  = 26 *(TRAIN_SENT2) +4
  RET                    ;(2)
;*************************************************************
; PULSE1                                  OK.
;-------------------------------------------------------------
PULSE:  MOV    R7,#09H          ;(1)
LOOP3:        MOV    R6,#14H          ;(1)  20
LOOP2:        MOV    R5,#0F8H         ;(1)  248
LOOP1:        DJNZ   R5,LOOP1         ;(2)
        DJNZ   R6,LOOP2         ;(2)
        DJNZ   R7,LOOP3         ;(2)
        RET
;1+R7(1+R6(1+R5(2)+2)+2)+2
;1+10 (1+20(1+248(2)+2)+2)+2
; = 0.09 sec. at 12 MHz
;*************************************************************
; PULSE2                                  OK.
;-------------------------------------------------------------
PULSE2:       MOV    R7,#02H          ;(1)
LOOP32:       MOV    R6,#14H          ;(1)  20
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LOOP22:     MOV    R5,#0F8H          ;(1)  248
LOOP12:     DJNZ   R5,LOOP12         ;(2)
            DJNZ   R6,LOOP22         ;(2)
            DJNZ   R7,LOOP32         ;(2)
            RET
;1+R7(1+R6(1+R5(2)+2)+2)+2
;1+2 (1+20(1+248(2)+2)+2)+2
; = 0.02 sec. at 12 MHz
;*****************************************************************
END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงโปรแกรมที่ควบคุมหุ่นยนต์มดในขณะที่ขนถ่ายอาหารไปที่บ้าน

```
; HOME3
;BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
  HIGHT          EQU   07D
  LLOW           EQU   01D
  XR0            EQU   30H
  XR1            EQU   31H
  SET_REG        EQU   35H
  RR1            EQU   3AH
  RR2            EQU   3BH
  TRAIN_SENT1 EQU   40H
  TRAIN_SENT2 EQU   41H
  CYCS_TIME   EQU   43H
  FOUND1              EQU   50H
  FOUND2              EQU   51H
        X_THING1      EQU   5AH
        X_THING2      EQU   5BH
;BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
              ORG     0000H
              AJMP    MAIN
;-----------------------------------------------
              ORG     0003H
    CLR         EX0
                  AJMP        RELEASE

;-----------------------------------------------
              ORG     000BH
                RETI
;-----------------------------------------------
              ORG     00013H
                  RETI
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;-----------------------------------------------
            ORG     001BH
                RETI
;-----------------------------------------------
            ORG     0023H
                RETI
;-----------------------------------------------
  ORG     0030H
MAIN:       MOV   P1,#11111000B
        SETB  P3.2
        SETB  P3.3
        CLR   P3.4
        SETB  PX0           ;IP.0
        SETB  IT0           ;TCON
        MOV   IE,#10000001B ;SET   /EA,EX0  OF  IE
                            ;ENABLE   INT
                            ;INITIAL     EX0
;*************************************************************
CHECK_LO:   CLR   P1.2
        SETB  09H
        JNB   P3.3,SOMETHING
        AJMP  CHECK_LO
;*************************************************************
SOMETHING:  MOV   FOUND1,#0100D
        MOV   FOUND2,#0195D
        MOV    X_THING2,#03D
SF2:        MOV   X_THING1,#0250D
SF1:        JB    P3.3,FOUND_F1       ;(2)
        DJNZ  X_THING1,SF1;(2)
        DJNZ  X_THING2,SF2;(2)
        AJMP  ESCAPE_F    ; >3000uS
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
FOUND_F1:     JNB    P3.3,SHORT_F ;<400uS ,15CYCS
              DJNZ   FOUND1,FOUND_F1
FOUND_F2:     JNB    P3.3,OK_F1    ; 400-1170 us ,15-45 CYCS
              DJNZ   FOUND2,FOUND_F2
              AJMP   CHECK_LO
;-------------------------------------------------------------------------
SHORT_F:      AJMP   CHECK_LO
ESCAPE_F:     AJMP   CHECK_LO
OK_F1:        SETB   P1.2
;-------------------------------------------------
              MOV    R3,#70D                ;5 S
;-------------------------------------------------
SIGNAL1:              JNB    09H,GO_CHECK_LO
              MOV    TRAIN_SENT1,#020D
              MOV    TRAIN_SENT2,#030D
              ACALL SENT2
              ACALL PULSE_X
              ACALL PULSE_X
              DJNZ   R3,SIGNAL1
GO_CHECK_LO: AJMP   CHECK_LO
;*********************************************************
SENT1:  MOV    TRAIN_SENT1,#250D
PULSE_TRAIN1:
  SETB   P3.4              ;(1)
  NOP
  MOV    XR0,#HIGHT    ;(2)
HI1:      DJNZ     XR0,HI1              ;(2)
  CLR    P3.4              ;(1)
  NOP
  MOV    XR1,#LLOW     ;(2)
LO1:      DJNZ           XR1,LO1                ;(2)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
TRAIN_ON1:
  DJNZ   TRAIN_SENT1,PULSE_TRAIN1        ;(2)
; [1 +1+2 + 2 *(XR0) ] + [ 1 +1+ 2 + 2 *(XR1) + (2) ] * (TRAIN_SENT1)
; =  [    H (18 )     +      L (8 )     ]   * (TRAIN_SENT1)
  RET                    ;6500
;*************************************************************
SENT2:  MOV    R0,#024D
SENT22: MOV    R1,TRAIN_SENT1
        MOV    R2,TRAIN_SENT2
  ACALL PULSE_HI              ;RECEIVED   0(2)
  ACALL PULSE_LO              ;RECEIVED   1(2)
  DJNZ   R0,SENT22            ;=50000
  RET
;-------------------------------------------------------------
;PULSE  TRAIN
;-------------------------------------------------------------
PULSE_HI:
PULSE_TRAIN2:
  SETB   P3.4                 ;(1)
  NOP                         ;(1)
  MOV    XR0,#HIGHT           ;(2)
HI:       DJNZ      XR0,HI              ;(2)
  CLR    P3.4                 ;(1)
  NOP                         ;(1)
  MOV    XR1,#LLOW            ;(2)
LO:       DJNZ           XR1,LO                  ;(2)
TRAIN_ON:
  DJNZ   R1,PULSE_TRAIN2      ;(2)
; [1 +1+2 + 2 *(XR0) ] + [ 1 +1+ 2 + 2 *(XR1) + (2) ] * (XR2)
; =  [    H (18 )     +      L (8 )     ]   * (XR2)
        RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;-------------------------------------------------------
PULSE_LO:
P_CYCLES:CLR P3.4                    ;(1)
  NOP                                ;(1)
  MOV   CYCS_TIME,#010D              ;(2)
CYCS:   DJNZ   CYCS_TIME,CYCS              ;(2)
  DJNZ   R2,P_CYCLES          ;(2)
          RET
; { (2)+ (2) } +
;{[(1)+(1)+(2)+(2)*(CYCS_TIME) +(2)] * (TRAIN_SENT2)}
; +{ (2) }
;  = 26 *(TRAIN_SENT2) + 6
;*********************************************
;INT0
;-------------------------------------------------------
RELEASE:      CLR    P3.4
RELEASE22T:   MOV    R7,#025D
RELEASE2T:    JB     P3.2,NO_INT0
          DJNZ   R7,RELEASE2T        ;100 us
          JNB    P3.2,OK_RELEASE
NO_INT0:      SETB   EX0
          RETI
OK_RELEASE:   CLR    09H
          SETB   P1.0
          MOV    R4,#15D
LOOP0:        MOV    TRAIN_SENT1,#020D
          MOV    TRAIN_SENT2,#060D
          ACALL SENT2
          ACALL PULSE_X
          DJNZ   R4,LOOP0
          MOV    R4,#0155D
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LOOP_WAIT:   ACALL SENT1          ;WAIT 1 SEC.
        DJNZ  R4,LOOP_WAIT
        CLR   P3.4
        CLR   P1.0
EXIT:        SETB  EX0
        RETI
;-------------------------------------------------------------
; PULSE1
;-------------------------------------------------------------
PULSE:  MOV    R7,#0100D              ;(1)
LOOP3:       MOV    R6,#14H              ;(1)  20
LOOP2:       MOV    R5,#0F8H             ;(1)  248
LOOP1:       DJNZ   R5,LOOP1             ;(2)
        DJNZ   R6,LOOP2             ;(2)
        DJNZ   R7,LOOP3             ;(2)
        RET
;1+R7(1+R6(1+R5(2)+2)+2)+2
;1+100 (1+20(1+248(2)+2)+2)+2
; = 1 sec. at 12 MHz
;-------------------------------------------------------------
PULSE_X:
             MOV   R7,#08D                ;(1)  8
LOOP32:      MOV    R6,#05H              ;(1)  5
LOOP22:      MOV    R5,#0F8H             ;(1)  248
LOOP12:      DJNZ   R5,LOOP12            ;(2)
        DJNZ   R6,LOOP22            ;(2)
        DJNZ   R7,LOOP32            ;(2)
        RET
;2500 *4 =20,000
;-------------------------------------------------------------
END
```

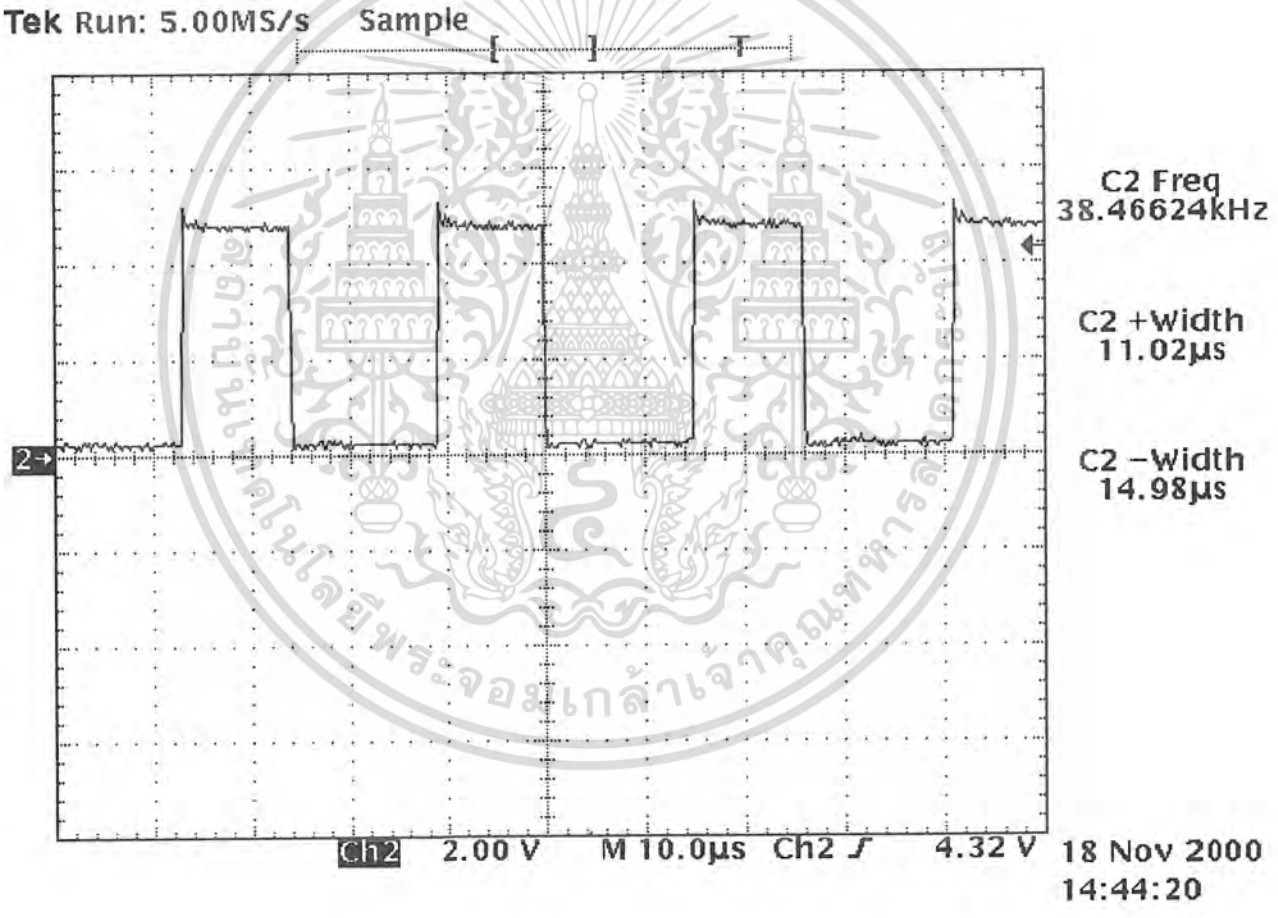
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลอง

สัญญาณของตัว หุ่นยนต์มด และ บ้านของมด นั้นจะมีการใช้ แสงอินฟาเรดในการติดต่อกันและตรวจสอบสิ่งกีดขวาง โดยสัญญาณที่ได้มีรายละเอียดดังต่อไปนี้

หุ่นยนต์มด มีด้วยกัน 2 สัญญาณคือ

1. สัญญาณตรวจสอบสิ่งกีดขวาง เป็นสัญญาณพัลส์ ความถี่ ประมาณ 38 กิโลเฮิร์ต ผลิตจากไอซี 555 โดยสามารถปรับขนาดความกว้างของพัลส์ได้ โดยการปรับค่าความต้านทานปรับค่าได้ R2 ของวงจรPage2 โดยสัญญาณที่วัดได้ มีลักษณะดังนี้

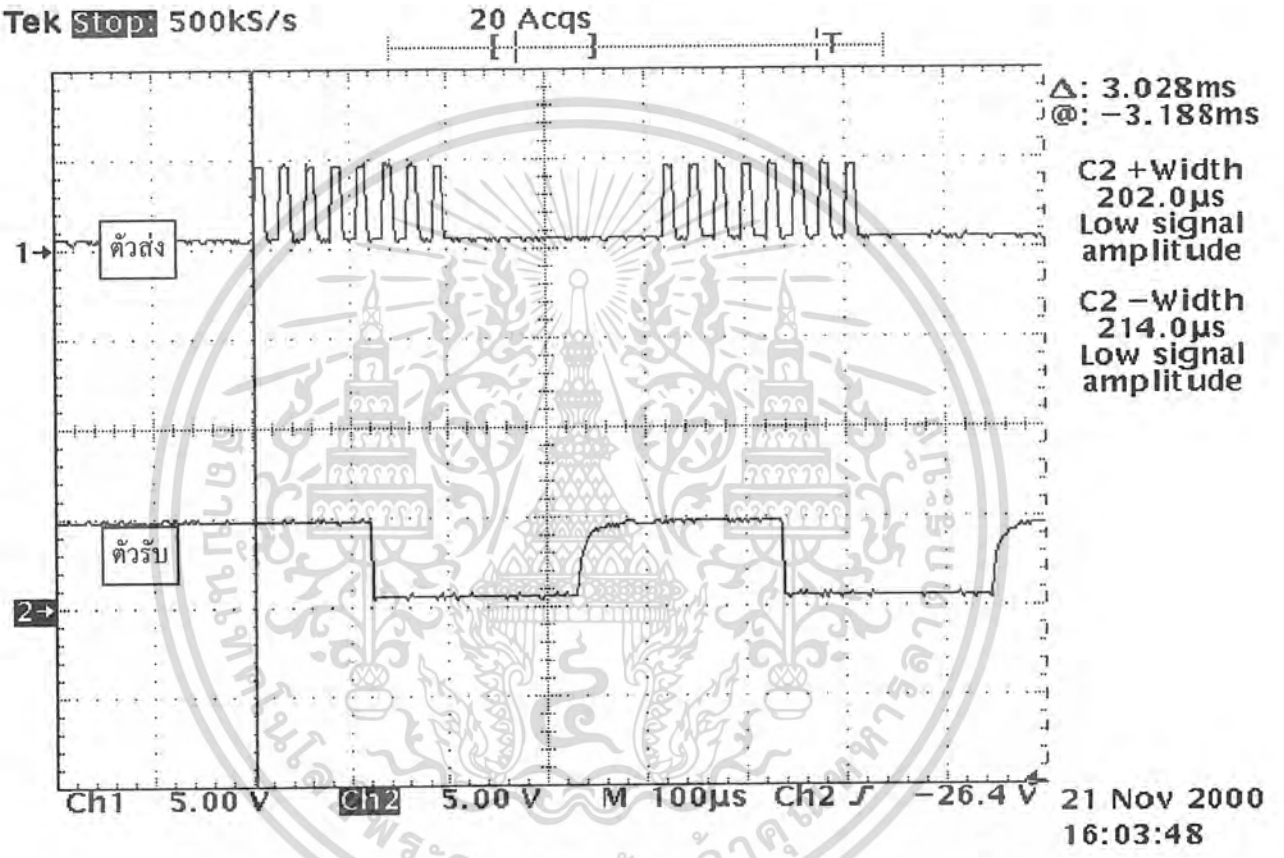


(7-1)ภาพแสดงสัญญาณ OUTPUT จาก ไอซี 555

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. สัญญาณติดต่อบ้านของมดและหุ่นยนต์มดตัวอื่นๆ เป็นสัญญาณที่มาจาก ไอซี 89C2051 แล้วส่งไปยังไอซี L293D เพื่อขยายกระแสก่อนส่งให้ไดโอดอินฟาเรด เป็นขบวนพัลส์ 20 พัลส์ โดยมีความถี่ 38 กิโลเฮิร์ต และเว้นระยะการส่ง 780 ไมโครวินาที แล้วส่งขบวนพัลส์ 20 พัลส์ มาอีก

ทางด้านตัวรับของบ้านของมดและหุ่นยนต์มดตัวอื่นๆ จะเป็นพัลส์ลบแล้วกลับมาเป็นพัลส์บวกมีคาบเวลา 780 ไมโครวินาที แล้วกลับเป็นพัลส์ลบอีก แล้วจึงกลับมาเป็นพัลส์บวก สามารถแสดงลักษณะตัวอย่างดังรูปข้างล่างซึ่งเป็นการ ส่งขบวนพัลส์ 8 พัลส์ โดยมีความถี่ 38 กิโลเฮิร์ต และเว้นระยะการส่ง 200 ไมโครวินาที



(7-2)ภาพแสดงลักษณะการ ส่งขบวนพัลส์ 8 พัลส์ ความถี่ 38 กิโลเฮิร์ต เว้นระยะการส่ง 200 ไมโครวินาที และสัญญาณด้านตัวรับ จาก TSOP1838

บ้านมด มีด้วยกัน 2 สัญญาณคือ

1 สัญญาณตรวจสอบสิ่งกีดขวาง เป็นสัญญาณพัลส์ ความถี่ 38 กิโลเฮิร์ต ผลิตจากไอซี 89C2051 เหมือนดังรูป(7-1)

2 สัญญาณเรียกหุ่นยนต์มด เป็นสัญญาณที่มาจาก ไอซี 89C2051 แล้วส่งไปยัง ทรานซิสเตอร์2N2222A เพื่อขยายกระแสก่อนส่งให้ไดโอดอินฟาเรด เป็นขบวนพัลส์ 20 พัลส์ โดยมีความถี่ 38 กิโลเฮิร์ต และเว้นระยะการส่ง 1560 ไมโครวินาที แล้วส่งขบวนพัลส์ 20 พัลส์ มาอีก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุป

การทำงานของตัวหุ่นยนต์มดและบ้านมด

การทำงานจะเป็นในลักษณะทำงาน โดยอัตโนมัติหลังจากจ่ายกระแสไฟฟ้าให้ และกดสวิทช์ Reset แล้ว ทั้งตัวหุ่นยนต์มดและบ้านมด

ปัญหาการทำงาน

1. ปัญหาสัญญาณรบกวนตัวส่งอินฟาเรดอันเนื่องมาจากตัวมอเตอร์ไฟฟ้า ลักษณะของตัวมอเตอร์ไฟฟ้าอย่างหนึ่งคือจะมีการดึงกระแสสูงในช่วง Start หรือ ช่วง เปลี่ยนทิศทางการหมุนของตัวมอเตอร์ไฟฟ้า ทำให้ตัวรับของตัวหุ่นยนต์มดตัวอื่นและบ้านหุ่นยนต์มด อ่านสัญญาณผิดพลาดได้ สามารถแก้ปัญหานี้ได้โดยการใช้โปรแกมตรวจสอบสัญญาณรบกวนนี้แล้วแก้ไขให้อยู่ในสภาวะเดิม

2. ในช่วง Startup ครั้งแรก ต้องมีการปรับแต่งสัญญาณอินฟาเรด ที่ส่งให้ ตัวส่งอินฟาเรด D3,D4 ของวงจรPage1 โดยปรับค่าความต้านทาน R2 แล้ววัดสัญญาณที่ส่งมาตัวส่งอินฟาเรด D3,D4 นี้ และตรวจสอบการตอบสนองของตัวรับอินฟาเรด U1,U2 ของวงจรPage1 ให้สัญญาณพัลส์ 0 โวล์ต ออกมาหรือไม่เมื่อมีสิ่งของมาสะท้อนสัญญาณอินฟาเรด

3. ปัญหาด้านการเคลื่อนที่ของล้อ พื้นที่มีลักษณะต้านการเคลื่อนที่ของล้อของตัวหุ่นยนต์มด จะทำให้หุ่นยนต์มดเคลื่อนที่ได้ดีกว่าพื้นลื่น ทำให้การหมุนหาสิ่งของที่เป็นอาหาร,การหลบสิ่งกีดขวาง หรือการหมุนตามสัญญาณเรียกจะหมุนต่างกันได้ตามลักษณะของพื้น

4. แหล่งจ่ายกระแสไฟฟ้า เนื่องจากใช้ มอเตอร์ไฟฟ้า 3 ตัว กระแสไฟฟ้าจึงสูงในการทำงาน (ประมาณ 300-800 มิลลิแอมป์) ทำให้สิ้นเปลืองพลังงานมาก การแก้ไขต้องมีการออกแบบกลไกให้ใช้ มอเตอร์ไฟฟ้า ให้น้อยลงกว่านี้ในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Features

- Compatible with MCS-51™ Products
- 2K Bytes of Reprogrammable Flash Memory
  - Endurance: 1,000 Write/Erase Cycles
- 2.7V to 6V Operating Range
- Fully Static Operation: 0 Hz to 24 MHz
- Two-level Program Memory Lock
- 128 x 8-bit Internal RAM
- 15 Programmable I/O Lines
- Two 16-bit Timer/Counters
- Six Interrupt Sources
- Programmable Serial UART Channel
- Direct LED Drive Outputs
- On-chip Analog Comparator
- Low-power Idle and Power-down Modes

8-bit Microcontroller with 2K Bytes Flash

AT89C2051

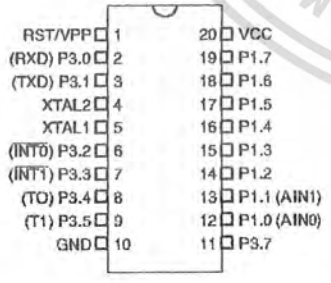
Description

The AT89C2051 is a low-voltage, high-performance CMOS 8-bit microcomputer with 2K 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 MCS-51 instruction set. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C2051 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications.

The AT89C2051 provides the following standard features: 2K bytes of Flash, 128 bytes of RAM, 15 I/O lines, two 16-bit timer/counters, a five vector two-level interrupt architecture, a full duplex serial port, a precision analog comparator, on-chip oscillator and clock circuitry. In addition, the AT89C2051 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 Configuration

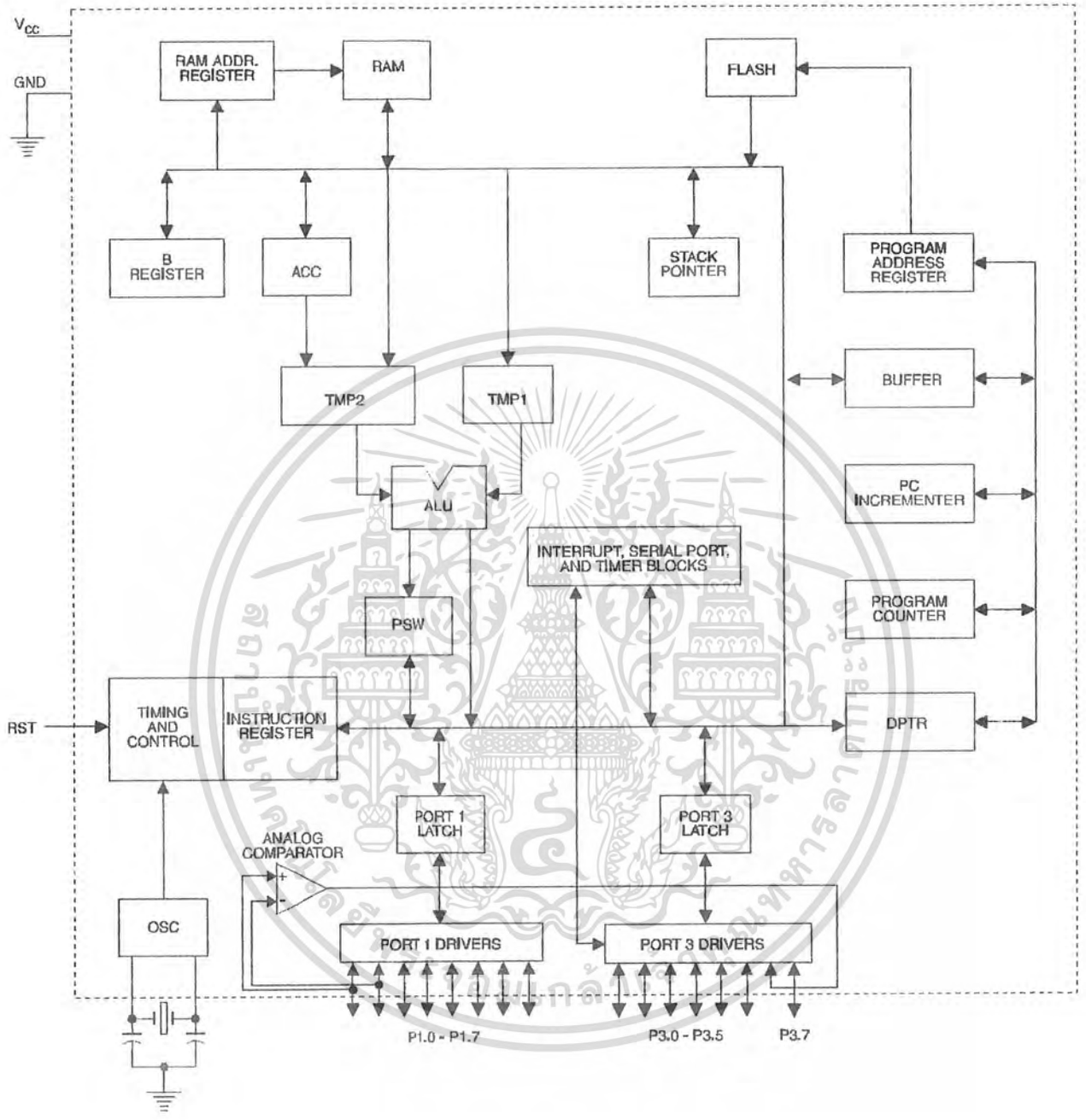
PDIP/SOIC



Rev. 0368E–02/00

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flash Programming Modes

Mode		RST/VPP	P3.2/$\overline{PROG}$	P3.3	P3.4	P3.5	P3.7
Write Code Data[1][3]		12V		L	H	H	H
Read Code Data[1]		H	H	L	L	H	H
Write Lock	Bit - 1	12V		H	H	H	H
	Bit - 2	12V		H	H	L	L
Chip Erase		12V	(2)	H	L	L	L
Read Signature Byte		H	H	L	L	L	L

Notes:
1. The internal PEROM address counter is reset to 000H on the rising edge of RST and is advanced by a positive pulse at XTAL 1 pin.
2. Chip Erase requires a 10 ms $\overline{PROG}$ pulse.
3. P3.1 is pulled Low during programming to indicate RDY/$\overline{BSY}$.

Figure 3. Programming the Flash Memory

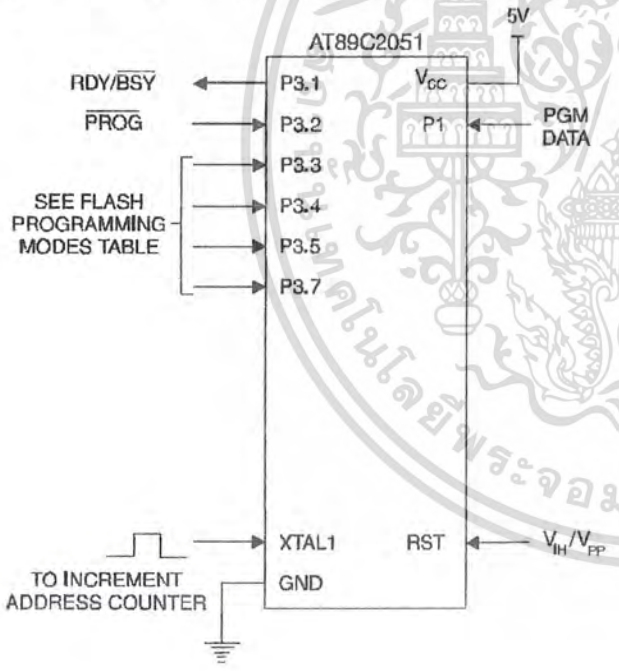
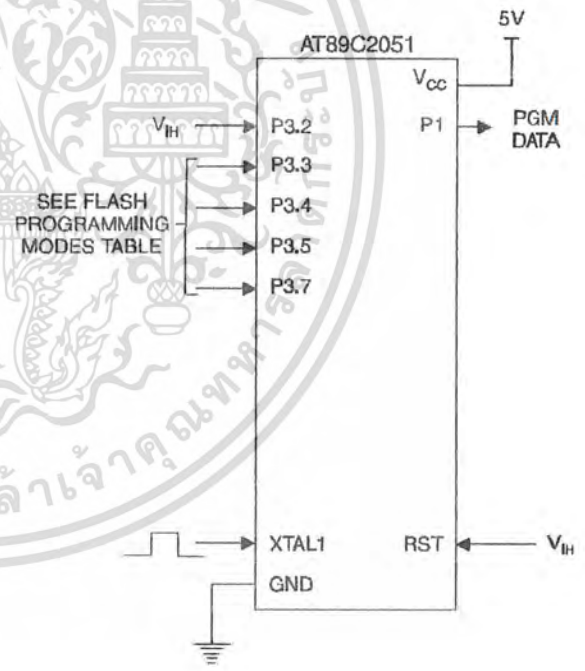


Figure 4. Verifying the Flash Memory



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

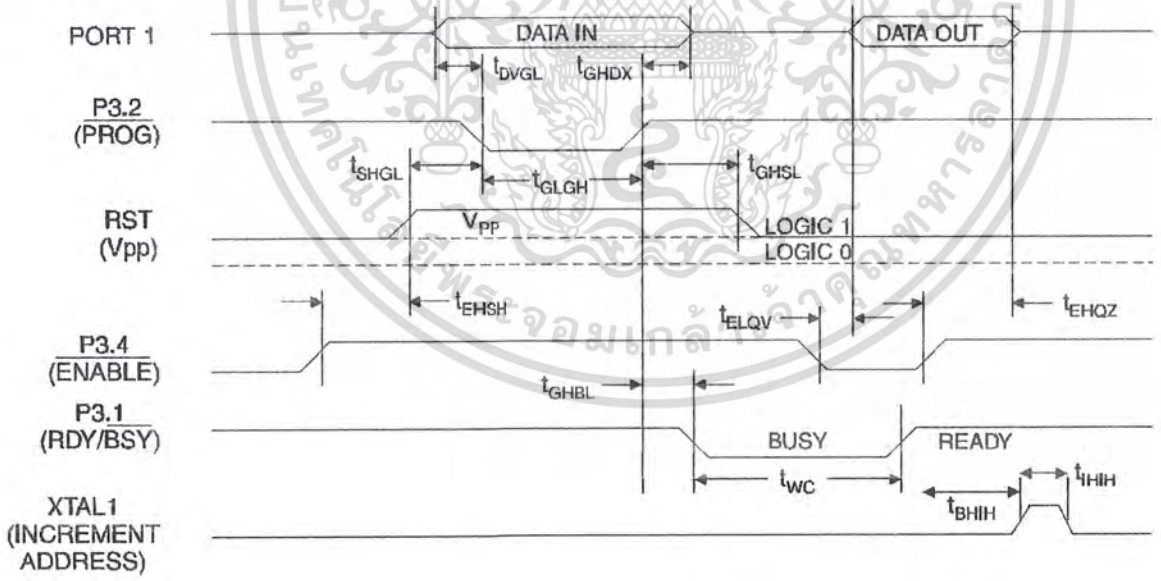
Flash Programming and Verification Characteristics

T_A = 0°C to 70°C, V_{CC} = 5.0 ± 10%

Symbol	Parameter	Min	Max	Units
V_{PP}	Programming Enable Voltage	11.5	12.5	V
I_{PP}	Programming Enable Current		250	µA
t_{DVGL}	Data Setup to $\overline{PROG}$ Low	1.0		µs
t_{GHDX}	Data Hold after $\overline{PROG}$	1.0		µs
t_{EHSH}	P3.4 ($\overline{ENABLE}$) High to V_{PP}	1.0		µs
t_{SHGL}	V_{PP} Setup to $\overline{PROG}$ Low	10		µs
t_{GHSL}	V_{PP} Hold after $\overline{PROG}$	10		µs
t_{GLGH}	$\overline{PROG}$ Width	1	110	µs
t_{ELQV}	$\overline{ENABLE}$ Low to Data Valid		1.0	µs
t_{EHQZ}	Data Float after $\overline{ENABLE}$	0	1.0	µs
t_{GHBL}	$\overline{PROG}$ High to $\overline{BUSY}$ Low		50	ns
t_{WC}	Byte Write Cycle Time		2.0	ms
t_{BHIH}	RDY/$\overline{BSY}$\ to Increment Clock Delay	1.0		µs
t_{IHIL}	Increment Clock High	200		ns

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

Flash Programming and Verification Waveforms



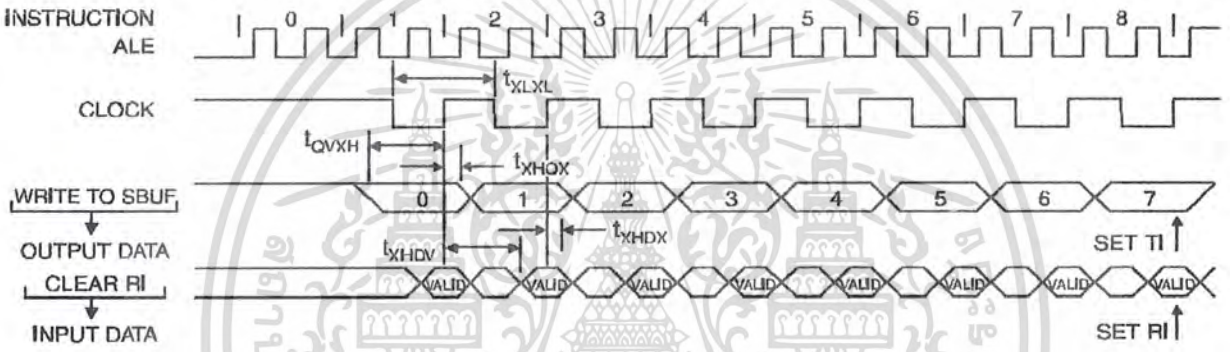
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Serial Port Timing: Shift Register Mode Test Conditions

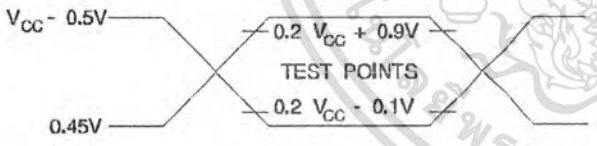
V_{CC} = 5.0V ± 20%; Load Capacitance = 80 pF

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

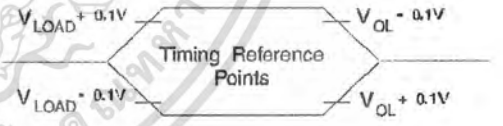


AC Testing Input/Output Waveforms[1]



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 100 mV change frothe loaded V_{OH}/V_{OL} level occurs.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

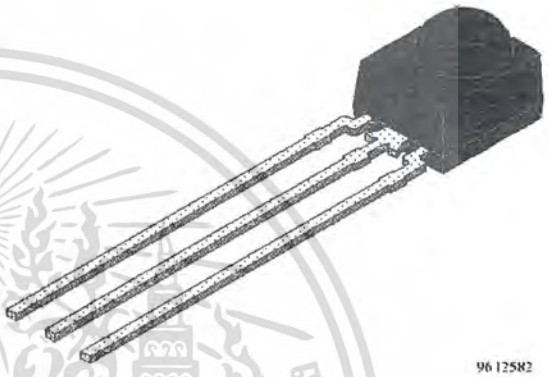
Photo Modules for PCM Remote Control Systems

Available types for different carrier frequencies

Type	fo	Type	fo
TSOP1830	30 kHz	TSOP1833	33 kHz
TSOP1836	36 kHz	TSOP1837	36.7 kHz
TSOP1838	38 kHz	TSOP1840	40 kHz
TSOP1856	56 kHz		

Description

The TSOP18.. – series are miniaturized receivers for infrared remote control systems. PIN diode and preamplifier are assembled on lead frame, the epoxy package is designed as IR filter.
The demodulated output signal can directly be decoded by a microprocessor. The main benefit is the reliable function even in disturbed ambient and the protection against uncontrolled output pulses.



9612582

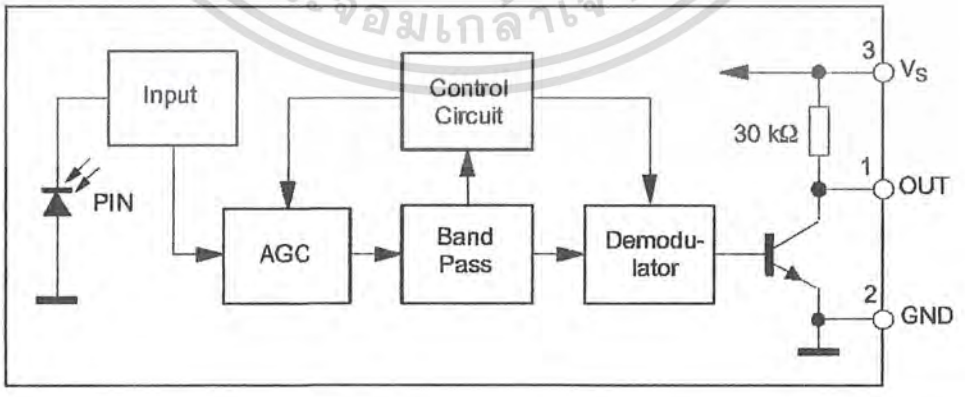
Features

- Photo detector and preamplifier in one package
- Internal filter for PCM frequency
- TTL and CMOS compatibility
- Output active low
- Improved shielding against electrical field disturbance
- Suitable burst length ≥6 cycles/burst

Special Features

- Small size package
- Enhanced immunity against all kinds of disturbance light
- No occurrence of disturbance pulses at the output
- Short settling time after power on (<200µs)

Block Diagram



9612226

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings

T_{amb} = 25°C

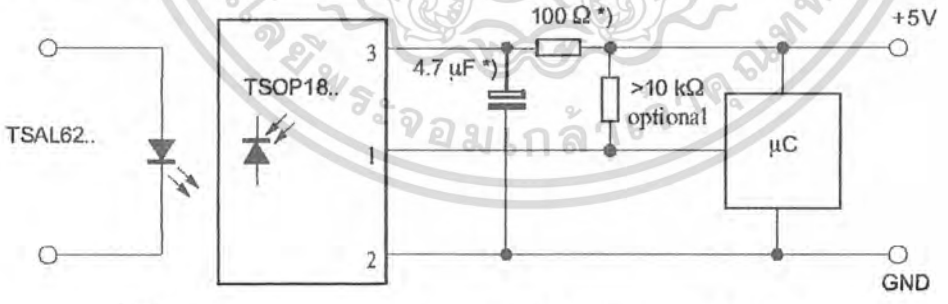
Parameter	Test Conditions	Symbol	Value	Unit
Supply Voltage	(Pin 3)	V_S	–0.3...6.0	V
Supply Current	(Pin 3)	I_S	5	mA
Output Voltage	(Pin 1)	V_O	–0.3...6.0	V
Output Current	(Pin 1)	I_O	5	mA
Junction Temperature		T_j	100	°C
Storage Temperature Range		T_{stg}	–25...+85	°C
Operating Temperature Range		T_{amb}	–25...+85	°C
Power Consumption	(T_{amb} ≦ 85 °C)	P_{tot}	50	mW
Soldering Temperature	t ≦ 10 s, 1 mm from case	T_{sd}	260	°C

Basic Characteristics

T_{amb} = 25°C

Parameter	Test Conditions	Symbol	Min	Typ	Max	Unit
Supply Current (Pin 3)	V_S = 5 V, E_v = 0	I_{SD}	0.9	1.2	1.5	mA
	V_S = 5 V, E_v = 40 klx, sunlight	I_{SH}		1.3		mA
Supply Voltage (Pin 3)		V_S	4.5		5.5	V
Transmission Distance	E_v = 0, test signal see fig.6, IR diode TSAL6200, I_F = 300 mA	d		35		m
Output Voltage Low (Pin 1)	I_{OSL} = 0.5 mA, E_e = 0.7 mW/m², f = f_o	V_{OSL}			250	mV
Irradiance (30 – 40 kHz)	Pulse width tolerance: $t_{pi} - 4/f_o < t_{po} < t_{pi} + 6/f_o$, test signal see fig.6	$E_{e\ min}$		0.3	0.5	mW/m²
Irradiance (56 kHz)		$E_{e\ min}$		0.4	0.7	mW/m²
Irradiance		$E_{e\ max}$	30			W/m²
Directivity	Angle of half transmission distance	$\varphi_{1/2}$		±45		deg

Application Circuit



15905

*) recommended to suppress power supply disturbances

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Suitable Data Format

The circuit of the TSOP18... is designed in that way that unexpected output pulses due to noise or disturbance signals are avoided. A bandpassfilter, an integrator stage and an automatic gain control are used to suppress such disturbances.
The distinguishing mark between data signal (not suppressed) and disturbance signal (supressed) are carrier frequency, burst length and Signal Gap Time (see diagram below).

The data signal should fullfill the following condition:

- Carrier frequency should be close to center frequency of the bandpass (e.g. 38kHz).
- Burst length should be 6 cycles/burst or longer.
- After each burst a gap time of at least 9 cycles is neccessary.
- The data format should not make a continuous signal transmission. There must be a Signal Gap Time (longer than 15ms) at least each 90ms (see Figure A)

Some examples for suitable data format are:
NEC Code (repetitive pulse), NEC Code (repetitive data), Toshiba Micom Format, Sharp Code, RC5 Code, RECS–80 Code, R–2000 Code.

When a disturbance signal is applied to the TSOP18... it can still receive the data signal. However the sensitivity is reduced to that level that no unexpected pulses will occure.

Some examples for such disturbance signals which are suppressed by the TSOP18... are:

- DC light (e.g. from tungsten bulb or sunlight),
- Continuous signal at 38kHz or at any other frequency,
- Signals from fluorescent lamps (see Figure B)
- Continuous IR signal (e.g. 1ms burst, 2ms pause)

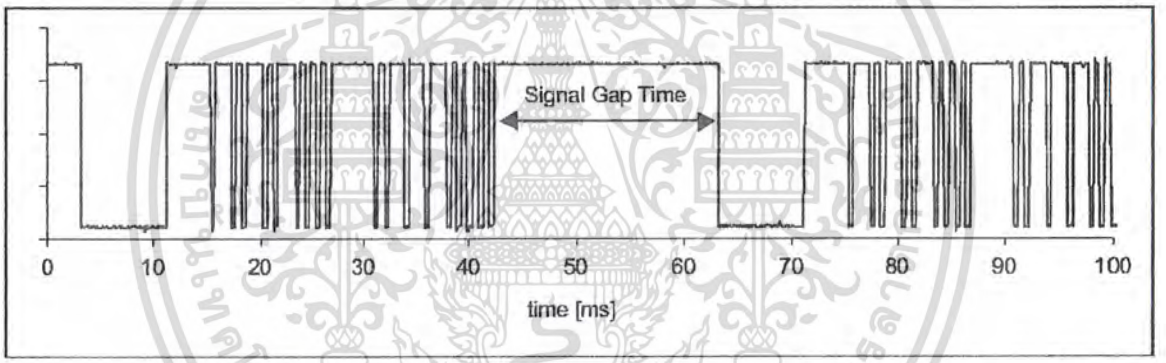


Figure A: Data Signal (Output of IR Receiver) with a Signal Gap Time of 20ms

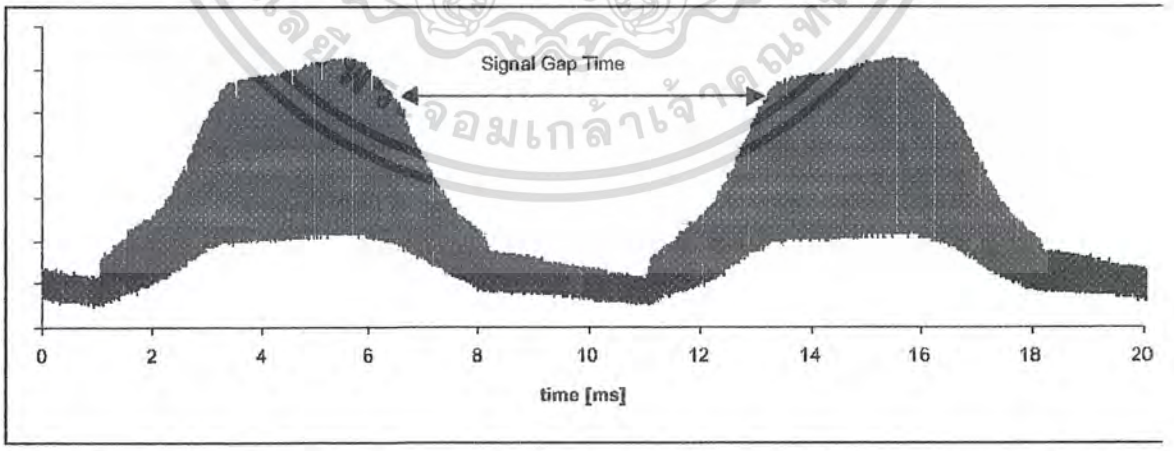


Figure B: Disturbance Signal from Fluorescent Lamp with Signal Gap Time of 7ms (suppressed by TSOP18..)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

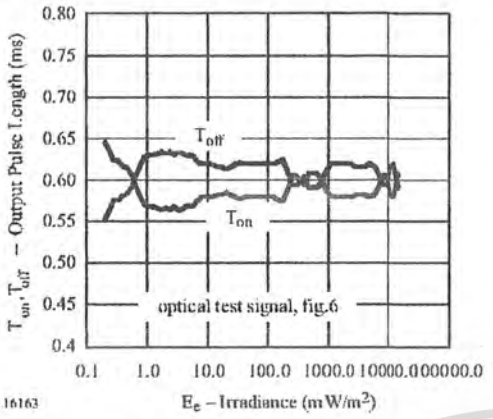


Figure 7. Output Pulse Diagram

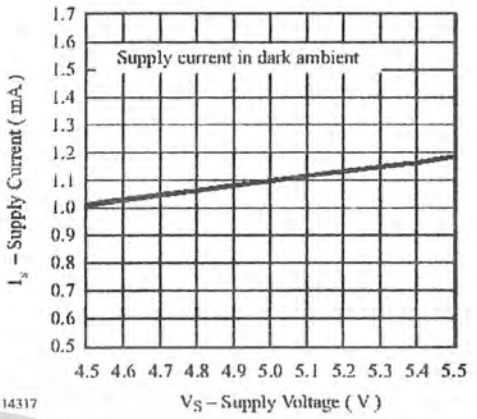


Figure 10. Supply Current vs. Supply Voltage

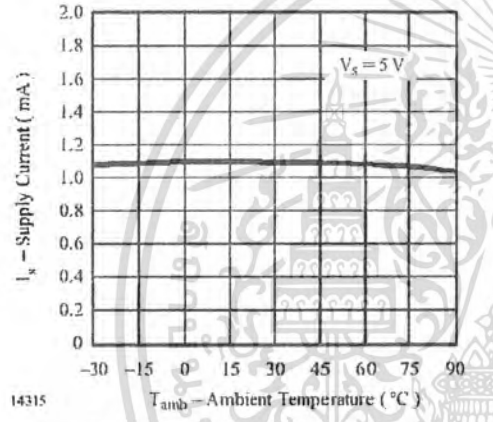


Figure 8. Supply Current vs. Ambient Temperature

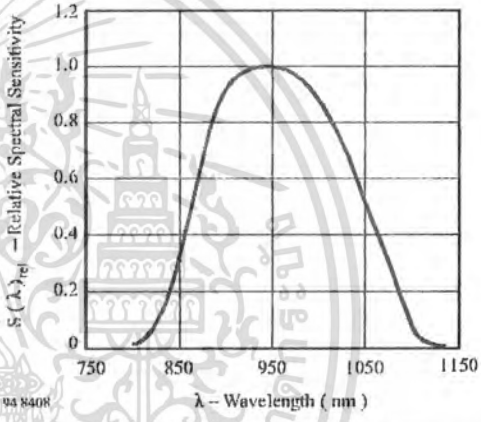


Figure 11. Relative Spectral Sensitivity vs. Wavelength

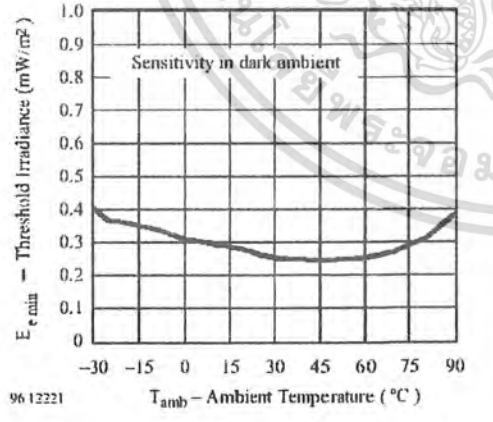


Figure 9. Sensitivity vs. Ambient Temperature

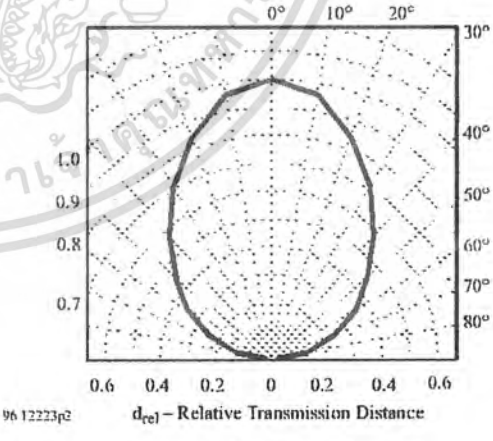


Figure 12. Directivity

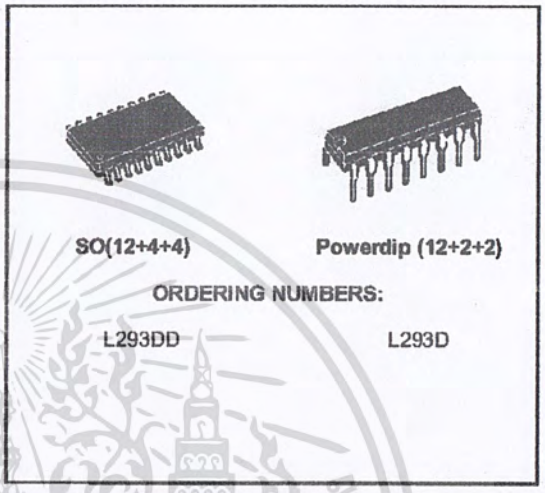
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SGS-THOMSON MICROELECTRONICS

L293D
L293DD

PUSH-PULL FOUR CHANNEL DRIVER WITH DIODES

- 600mA OUTPUT CURRENT CAPABILITY PER CHANNEL
- 1.2A PEAK OUTPUT CURRENT (non repetitive) PER CHANNEL
- ENABLE FACILITY
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)
- INTERNAL CLAMP DIODES



DESCRIPTION

The Device is a monolithic integrated high voltage, high current four channel driver designed to accept standard DTL or TTL logic levels and drive inductive loads (such as relays solenoides, DC and stepping motors) and switching power transistors.

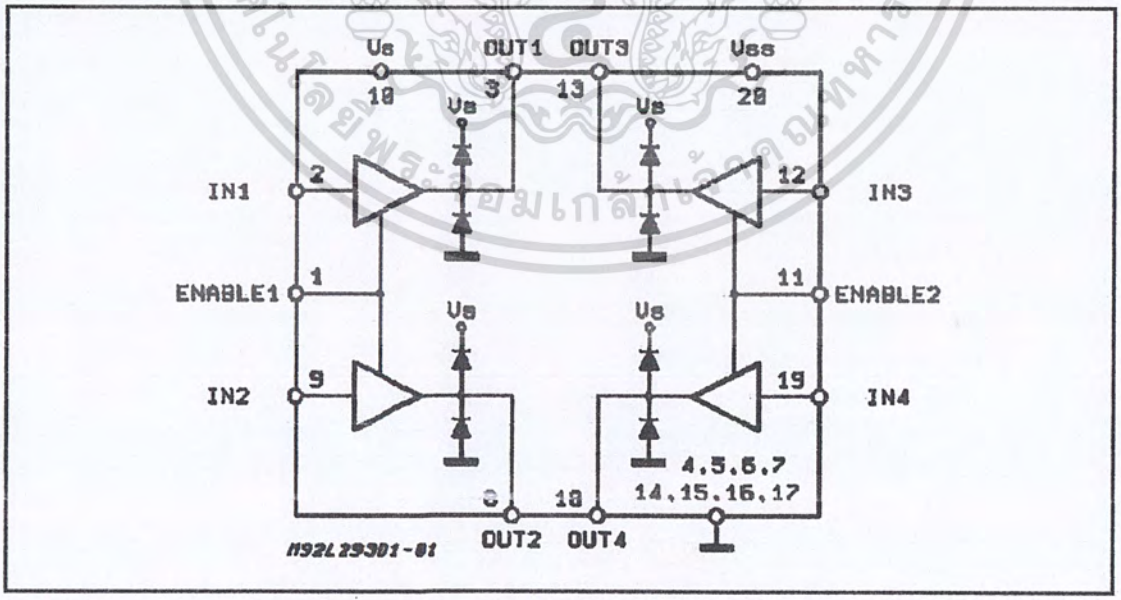
To simplify use as two bridges each pair of channels is equipped with an enable input. A separate supply input is provided for the logic, allowing operation at a lower voltage and internal clamp diodes are included.

This device is suitable for use in switching applications at frequencies up to 5 kHz.

The L293D is assembled in a 16 lead plastic packaage which has 4 center pins connected together and used for heatsinking

The L293DD is assembled in a 20 lead surface mount which has 8 center pins connected together and used for heatsinking.

BLOCK DIAGRAM

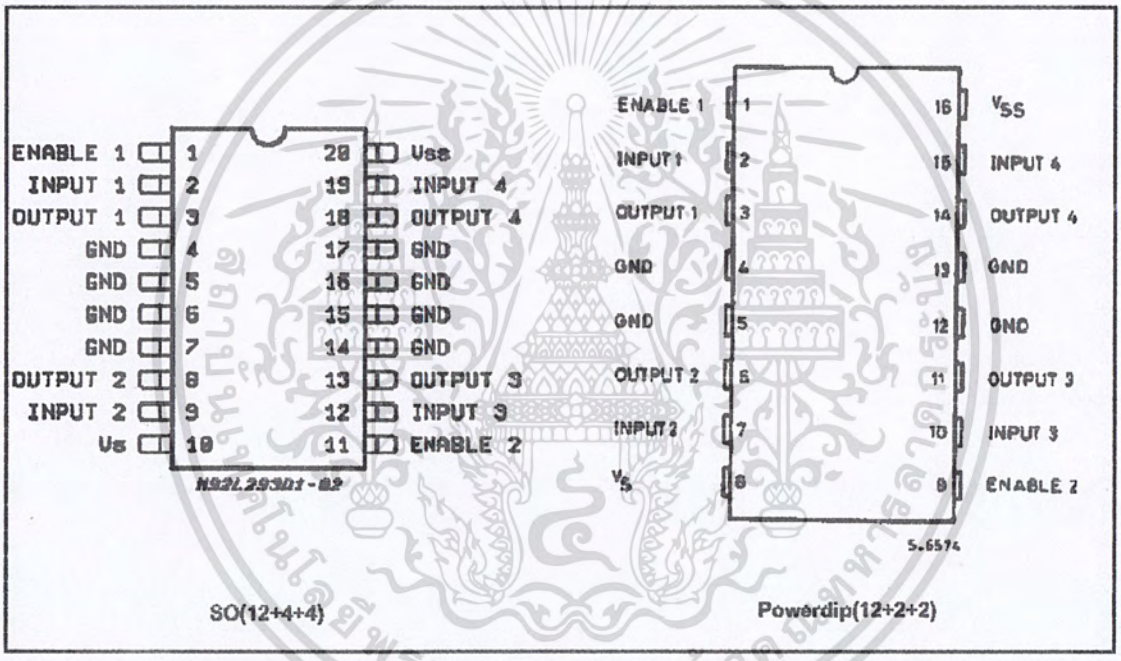


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_S	Supply Voltage	36	V
V_{SS}	Logic Supply Voltage	36	V
V_i	Input Voltage	7	V
V_{en}	Enable Voltage	7	V
I_o	Peak Output Current (100 µs non repetitive)	1.2	A
P_{tot}	Total Power Dissipation at T_{pins} = 90 |C	4	W
T_{stg}, T_j	Storage and Junction Temperature	– 40 to 150	|C

PIN CONNECTIONS (Top view)



THERMAL DATA

Symbol	Decription		DIP	SO	Unit
$R_{th\ j\text{-}pins}$	Thermal Resistance Junction-pins	max.	–	14	|C/W
$R_{th\ j\text{-}amb}$	Thermal Resistance junction-ambient	max.	80	50 (*)	|C/W
$R_{th\ j\text{-}case}$	Thermal Resistance Junction-case	max.	14	–	

(*) With 6sq. cm on board heatsink.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

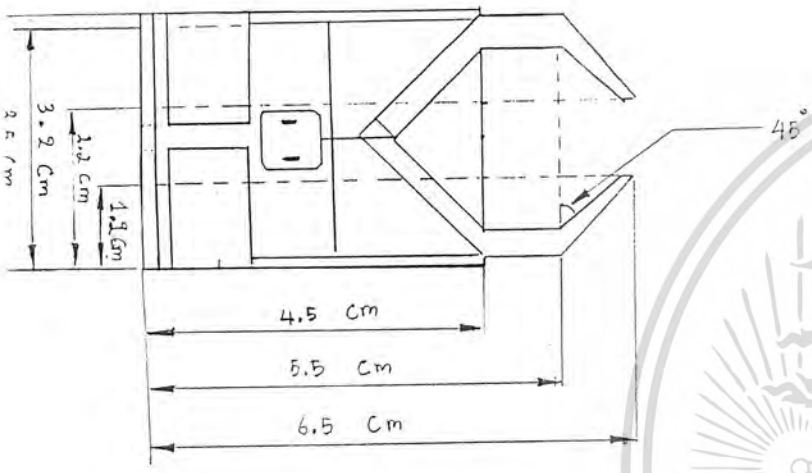
ELECTRICAL CHARACTERISTICS (for each channel, V_S = 24 V, V_{SS} = 5 V, T_{amb} = 25 °C, unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V_S	Supply Voltage (pin 10)		V_{SS}		36	V
V_{SS}	Logic Supply Voltage (pin 20)		4.5		36	V
I_S	Total Quiescent Supply Current (pin 10)	V_i = L ; I_O = 0 ; V_{en} = H		2	6	mA
		V_i = H ; I_O = 0 ; V_{en} = H		16	24	mA
		V_{en} = L			4	mA
I_{SS}	Total Quiescent Logic Supply Current (pin 20)	V_i = L ; I_O = 0 ; V_{en} = H		44	60	mA
		V_i = H ; I_O = 0 ; V_{en} = H		16	22	mA
		V_{en} = L		16	24	mA
V_{IL}	Input Low Voltage (pin 2, 9, 12, 19)		– 0.3		1.5	V
V_{IH}	Input High Voltage (pin 2, 9, 12, 19)	$V_{SS} \leq 7$ V	2.3		V_{SS}	V
		$V_{SS} > 7$ V	2.3		7	V
I_{IL}	Low Voltage Input Current (pin 2, 9, 12, 19)	V_{IL} = 1.5 V			– 10	µA
I_{IH}	High Voltage Input Current (pin 2, 9, 12, 19)	$2.3 \text{ V} \leq V_{IH} \leq V_{SS} - 0.6$ V		30	100	µA
$V_{en\,L}$	Enable Low Voltage (pin 1, 11)		– 0.3		1.5	V
$V_{en\,H}$	Enable High Voltage (pin 1, 11)	$V_{SS} \leq 7$ V	2.3		V_{SS}	V
		$V_{SS} > 7$ V	2.3		7	V
$I_{en\,L}$	Low Voltage Enable Current (pin 1, 11)	$V_{en\,L}$ = 1.5 V		– 30	– 100	µA
$I_{en\,H}$	High Voltage Enable Current (pin 1, 11)	$2.3 \text{ V} \leq V_{en\,H} \leq V_{SS} - 0.6$ V			± 10	µA
$V_{CE(sat)H}$	Source Output Saturation Voltage (pins 3, 8, 13, 18)	I_O = – 0.6 A		1.4	1.8	V
$V_{CE(sat)L}$	Sink Output Saturation Voltage (pins 3, 8, 13, 18)	I_O = + 0.6 A		1.2	1.8	V
V_F	Clamp Diode Forward Voltage	I_O = 600nA		1.3		V
t_r	Rise Time (*)	0.1 to 0.9 V_O		250		ns
t_f	Fall Time (*)	0.9 to 0.1 V_O		250		ns
t_{on}	Turn-on Delay (*)	0.5 V_i to 0.5 V_O		750		ns
t_{off}	Turn-off Delay (*)	0.5 V_i to 0.5 V_O		200		ns

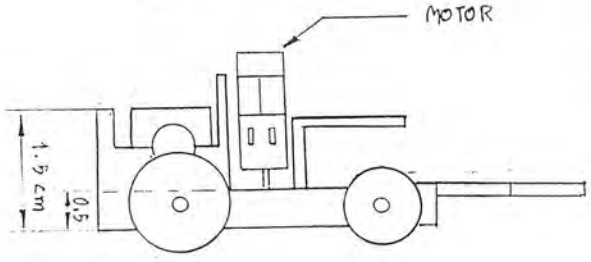
(*) See fig. 1.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

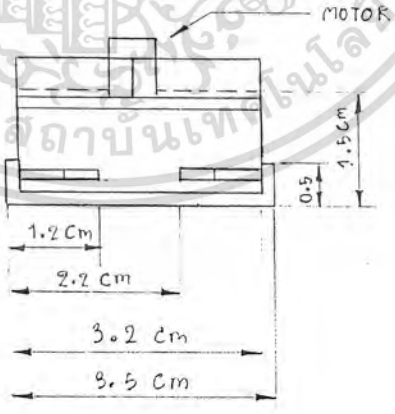
TOP VIEW



SIDE VIEW



FRONT VIEW



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้