

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

หุ่นยนต์สำรวจ

Explorer Robot

โดย

นายนิวัตร คังกัน รหัส 47015211

นายภักดี ศรีประไพ รหัส 47015215

นายปณิธ เบญจานุกรม รหัส 47015251

อาจารย์ที่ปรึกษา

รศ.ดร. ชูชาติ ปิณฑวิรุจน์

๕/๗

๙๘ ๖7477

2549

เลขหมู่.....  
เลขทะเบียน..... 72151  
วัน,เดือน,ปี..... 11 ส.ย. 2550

b. 117 ๖4090  
i.....

รายงานเรื่องนี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์สำหรับปีการศึกษา 2549

ภาควิชา อิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง หุ่นยนต์สำรวจ (Explorer Robot)

ผู้จัดทำ

1. นายนิวัฒน์ กังกัน รหัส 47015211
2. นายภักดี ศรีประไพ รหัส 47015215
3. นายปณิธิ เบลูจานุกรม รหัส 47015251

  
..... อาจารย์ที่ปรึกษา  
(รศ.ดร.ชูชาติ ปิณฑวิรุจน์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## หุ่นยนต์สำรวจ

นายนิวัตร ตั้งกัน รหัส 47015211  
นายภักดี ศรีประไพ รหัส 47015215  
นายปณิธ เเบญจานุกรม รหัส 47015251  
รศ.ดร.ชูชาติ ปิยะทวิรุจน์ อาจารย์ที่ปรึกษา  
ปีการศึกษา 2549

### บทคัดย่อ

ในปัจจุบันหุ่นยนต์ได้เข้ามามีบทบาทและถูกนำมาประยุกต์ใช้งานอย่างกว้างขวางทั้งในวงการแพทย์ อุตสาหกรรม งานวิจัยทางด้านวิทยาศาสตร์ และที่เห็นกันบ่อยๆก็คือ การนำหุ่นยนต์ไปใช้ในการสำรวจหรืองานที่เกี่ยวข้องแทนมนุษย์ ดังนั้นปัญญาประดิษฐ์ฉบับนี้จึงได้นำเสนอแนวความคิดในการศึกษาการออกแบบ และสร้างหุ่นยนต์สำรวจที่ควบคุมผ่านคอมพิวเตอร์แบบไร้สาย โดยจะอาศัยขากกล้อง CCD 2 ตัว มาทำการประมวลผลสัญญาณภาพพิกัด 3 มิติ ระยะห่างจากวัตถุที่คิดขวางเพื่อควบคุมการเคลื่อนที่ของหุ่นยนต์ด้วยไมโครคอนโทรลเลอร์ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Explorer Robot

**Mr.Niwat kangkun ID.47015211**

**Mr.Phakdee Sriprapai ID.47015215**

**Mr.Panit Benjanukro ID.47015251**

**Assoc.Prof.Chuchat Pintawirut Advisor**

**Educational Year 2006**

### Abstract

In the present, Robot is widely extended in to many industry sector, medical surgery and science research. Robot is often used in the survey method and the risky work to replace the humans .The objective of thesis is to study, design and devise a robot which can be controlled via a wireless computer .The 2 CCD cameras will be installed to capture the stereo-scopic image which can be process by PC to device the 3D coordinate. The 3D coordinate will be compiled and used for controlling the movement of Robot Via micro processor.

## กิตติกรรมประกาศ

โครงการนี้จะไม่สามารถลุล่วงไปได้ ถ้าไม่ได้รับความร่วมมือและความช่วยเหลือจากหลาย ๆ ฝ่าย โดยเฉพาะอาจารย์ รศ.ดร.ชูชาติ ปิณฑวิรุจน์ (อาจารย์ที่ปรึกษา), อาจารย์ รศ.ดร. มนัส สังวรศิลป์, อาจารย์พลศาสตร์ เลิศประเสริฐ และอาจารย์ในภาควิชาอิเล็กทรอนิกส์ทุกท่าน ที่ให้การอุปการะในการให้คำปรึกษาและแนะนำเกี่ยวกับโครงการในครั้งนี้ และให้ยืมใช้เครื่องมือและอุปกรณ์อิเล็กทรอนิกส์ในการทดลอง และสั่งสอนให้ความรู้จนสามารถนำมาประยุกต์ใช้งานในการทำโครงการครั้งนี้ อีกทั้งเพื่อนที่ได้ร่วมทำงานและแลกเปลี่ยนความรู้ ความคิดเห็นซึ่งกันและกันจนทำให้ผลงานที่ได้มีประสิทธิภาพและเป็นประโยชน์ต่อส่วนรวมอีกด้วย

นายนิวัตร คังกัน  
 นายภักดี ศรีประไพ  
 นายปณิธ เบญจานุกรม  
 ผู้จัดทำโครงการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ

เรื่อง	หน้า
บทคัดย่อ	I
Abstract	II
สารบัญ	III
สารบัญรูปภาพ	V
สารบัญตาราง	VIII
<b>บทที่ 1 บทนำ</b>	
1.1 วัตถุประสงค์	I
1.2 ขอบเขตของโครงการ	2
1.3 ประโยชน์ที่ได้รับจากโครงการ	2
1.4 รายละเอียดโดยย่อของโครงการ	2
1.5 รายละเอียดของปริญญานิพนธ์	3
<b>บทที่ 2 ทฤษฎีและหลักการ</b>	
2.1 ไมโครคอนโทรลเลอร์ MCS-51	4
2.2 การใช้งานไมโครคอนโทรลเลอร์กับพอร์ทอนุกรม	15
2.3 การสื่อสารข้อมูล	22
2.4 มอเตอร์ (DC Motor)	28
2.5 เพาเวอร์มอสเฟต	29
2.6 การเชื่อมต่อทางแสง	33
2.7 หลักการออกแบบส่วนประกอบและโครงสร้างต่างๆ	33
2.8 คณิตศาสตร์ของภาพดิจิทัล (Digital Image Processing)	36
2.9 การแยกส่วนสี (Color Segmentation)	40
2.10 การติดตามวัตถุ (Object Tracking)	44
2.11 การเกิดภาพสเตอริโอ	46
2.12 หลักการทำงานของมอเตอร์เซอร์โว (Servo motor)	48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เรื่อง	หน้า
<b>บทที่ 3 การออกแบบส่วนต่างๆ ของหุ่นยนต์</b>	
3.1 แนวคิดในการออกแบบ โครงงาน	52
3.2 โปรแกรมภายในเครื่องคอมพิวเตอร์	52
3.3 การออกแบบ ภาคขับเคลื่อน	60
3.4 โครงสร้างการทำงานของหุ่นยนต์	62
<b>บทที่ 4 ผลการทดสอบ</b>	
4.1 การทดลองที่ 1. การติดตามวัตถุ (Object Tracking)	67
4.2 การเคลื่อนที่ของหุ่นยนต์โดยการติดตามวัตถุ	68
4.3 การทดลองที่ 2. ผลการทดลองวัดระยะ โดยทฤษฎี Stereo Scopic	72
4.4 รูปแบบของหน้าต่างโปรแกรมการทดลองการเกิดภาพสเตอริโอ	73
4.5 ผลการทดลองการเคลื่อนที่ของหุ่นยนต์ด้วยทฤษฎีสเตอริโอ สเตอริโอ (Stereoscopic)	74
<b>บทที่ 5 สรุปผลการทดลองและแนวทางการพัฒนาต่อ</b>	
5.1 สรุปผลการทดลองที่ 1 การติดตามวัตถุ (Object Tracking)	77
5.2 การทดลองที่ 2. ผลการทดลองวัดระยะ โดยทฤษฎีสเตอริโอ สเตอริโอ (Stereoscopic)	78
5.3 การแก้ปัญหาของการวัดระยะ โดยทฤษฎีสเตอริโอ สเตอริโอ (Stereoscopic)	78
<b>ภาคผนวก</b>	
<b>กิตติกรรมประกาศ</b>	
<b>เอกสารอ้างอิง</b>	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูปภาพ

	หน้า
รูปที่ 2.1 แสดงโครงสร้างภายในของไมโครคอนโทรลเลอร์ MCS-51	5
รูปที่ 2.2 แสดงรูปร่างและการจัดวางขาต่างๆ ของไมโครคอนโทรลเลอร์ MCS-51	6
รูปที่ 2.3 แสดงการใช้หน่วยความจำสำหรับเก็บโปรแกรม	7
รูปที่ 2.4 แสดงการจัดพื้นที่ของหน่วยความจำโปรแกรมภายในและภายนอก	7
รูปที่ 2.5 แสดงหน่วยความจำสำหรับเก็บข้อมูลภายในไมโครคอนโทรลเลอร์ MCS-51	8
รูปที่ 2.6 แสดงการจัดหน่วยความจำข้อมูล	9
รูปที่ 2.7 แสดงการต่อกับหน่วยความจำข้อมูลภายนอกไอซี	10
รูปที่ 2.8 แสดงการเลือกใช้รีจิสเตอร์ใช้งานทั่วไป R0-R7 ในแต่ละกลุ่ม	11
รูปที่ 2.9 ตารางแสดงถึงหน้าที่ของขาต่างๆ	16
รูปที่ 2.10 แสดงสัญญาณต่างๆที่ส่งในรูปแบบอนุกรม	17
รูปที่ 2.11 แสดงการเชื่อมต่อของสายสัญญาณ	19
รูปที่ 2.12 รูปคลื่นของสัญญาณที่ส่ง	20
รูปที่ 2.13 แสดงโครงสร้างภายในและตำแหน่งขาต่างๆของ Max232	22
รูปที่ 2.14 แสดงการรับส่งข้อมูลแบบทางเดียว (Simplex)	23
รูปที่ 2.15 แสดงการรับส่งข้อมูลสวนทางกันได้แบบผลัดการส่ง(Half Duplex)	24
รูปที่ 2.16 แสดงการรับส่งข้อมูลแบบสวนทางกันได้พร้อมกัน (Full Duplex)	25
รูปที่ 2.17 แสดงการส่งข้อมูลแบบขนาน	25
รูปที่ 2.18 แสดงการส่งข้อมูลแบบอนุกรม	26
รูปที่ 2.19 แสดงการส่งข้อมูลแบบสัมพันธ์	27
รูปที่ 2.20 แสดงการส่งข้อมูลแบบไม่สัมพันธ์	27
รูปที่ 2.21 โครงสร้างของเพาเวอร์มอสเฟต	30
รูปที่ 2.22 แรงดันพียงทลาคที่ตรงกับซอร์สเมื่ออุณหภูมิของรอสต์เปลี่ยนไป	30
รูปที่ 2.23 แสดงตัวอย่างของ Gate Charge Chart	31
รูปที่ 2.24 (ก) แสดงการชาร์จประจุที่ขาเกตตามเวลาที่มีผลต่อการนำกระแส	32
รูปที่ 2.24 (ข) แสดงผลเมื่อเริ่มหตุคนำกระแสของเพาเวอร์มอสเฟต	32
รูปที่ 2.25 แสดงฮิสโตแกรมของระดับความเข้มของภาพขนาด 256 ระดับ	38
รูปที่ 2.26 กราฟฮิสโตแกรมที่พื้นหลังกับวัตถุมีระดับความเข้มภาพไม่แตกต่างกันมาก	40

รูปที่ 2.27 ภาพสีขนาดจากกล้องขนาด 24 บิต	41
รูปที่ 2.28 ภาพสีที่ถูกแปลงเป็นสีเทาขนาด 8 บิต	42
รูปที่ 2.29 แสดงระดับฮิสโตแกรมของรูปที่ 2.28	42
รูปที่ 2.30 แสดงจุดเล็กสุดของภาพที่ประกอบไปด้วยแม่สีแดง สีเขียว สีน้ำเงิน	43
รูปที่ 2.31 แสดงการแทรกกิ่งโคที่ใช้การเปรียบเทียบโมเดล	45
รูปที่ 2.32 แสดงการผสมสีกันของแม่สีทั้งสาม	46
รูปที่ 2.33 แบบจำลองของขบวนการสร้างภาพสเตอริโอ	47
รูปที่ 2.34 มุมของรูปที่ 2.33 โดยที่ระบบพิกัดของกล้องที่หนึ่งซ้อนทับกับระบบพิกัดของโลก	47
รูปที่ 2.35 ชุดเฟืองเกียร์ ส่วนประกอบหลักของ Servo motor มีน้ำหนักเบา และส่งแรงบิดสูง	48
รูปที่ 2.36 บอร์ดควบคุม ของ Servo motor ตามสัญญาณ pulse	49
รูปที่ 2.37 รูปของ สัญญาณที่สามารถป้อนให้กับ Servo motor	49
รูปที่ 2.38 ตำแหน่งการหมุนของ servo motor	50
รูปที่ 3.1 แสดงโปรแกรมที่สร้างด้วย Delphi 7.0	52
รูปที่ 3.2 โฟลวชาร์ทโปรแกรมหลัก	54
รูปที่ 3.3 โฟลวชาร์ทโปรแกรมแยกส่วนสี	56
รูปที่ 3.4 โฟลวชาร์ทโปรแกรมแทรกกิ่งวัตถุ	57
รูปที่ 3.5 โฟลวชาร์ทโปรแกรมการควบคุมหุ่นยนต์แบบ Manual	58
รูปที่ 3.6 โฟลวชาร์ทหลักวัตถุ	59
รูปที่ 3.7 วงจรภาคขับเคลื่อนมอเตอร์แบบ H-Bridge	60
รูปที่ 3.8 สัญลักษณ์ของ MOSFET	61
รูปที่ 3.9 แสดงโครงสร้างการทำงานหลักๆ ของหุ่นยนต์	62
รูปที่ 3.10 แสดงโครงสร้างการทำงาน โดยละเอียดของหุ่นยนต์	63
รูปที่ 3.11 แสดงโครงสร้างของหุ่นยนต์มองจากด้านข้าง	64
รูปที่ 3.12 แสดงโครงสร้างของหุ่นยนต์มองจากด้านหน้า	64
รูปที่ 3.13 แสดงโครงสร้างของหุ่นยนต์มองจากด้านบน	65
รูปที่ 3.14 รูปถ่ายจากด้านบนของหุ่นยนต์	66
รูปที่ 3.15 รูปถ่ายจากด้านหน้าของหุ่นยนต์	66
รูปที่ 4.1 แสดงภาพหน้าต่างของโปรแกรม	67
รูปที่ 4.2 การทดสอบการทำจับภาพวัตถุและทำเทรสโอส	68
รูปที่ 4.3 ทดสอบการเคลื่อนที่ไปทางขวาของหุ่นยนต์	69

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.4 ทดสอบการเคลื่อนที่ไปทางซ้ายของหุ่นยนต์	69
รูปที่ 4.5 ทดสอบการเคลื่อนที่ไปข้างหน้าของหุ่นยนต์	70
รูปที่ 4.6 ทดสอบการเคลื่อนที่ถอยหลังของหุ่นยนต์	71
รูปที่ 4.7 หน้าต่างโปรแกรมการเกิดภาพสเตอริโอ	73
รูปที่ 4.8 การทำเทรสโพลของวัตถุสีในโปรแกรมสเตอริโอสโคปิก	74
รูปที่ 4.9 ตำแหน่งของวัตถุที่ทำให้หุ่นยนต์เคลื่อนที่ไปข้างหน้า	74
รูปที่ 4.10 ตำแหน่งของวัตถุที่ทำให้หุ่นยนต์เคลื่อนที่ไปทางด้านขวา	75
รูปที่ 4.11 ตำแหน่งของวัตถุที่ทำให้หุ่นยนต์เคลื่อนที่ไปทางด้านซ้าย	75
รูปที่ 4.12 ตำแหน่งของวัตถุที่ทำให้หุ่นยนต์หยุดการเคลื่อนที่	76



สารบัญตาราง

	หน้า
ตารางที่ 3.1 แสดงค่าทิศทางเคลื่อนที่ของหุ่นยนต์	62
ตารางที่ 4.1 แสดงผลการทดลองการหาค่า focus	72
ตารางที่ 4.2 ผลการทดลองการหาระยะทาง (Z) ของกล้องสเตอริโอ	72



## บทที่ 1

### บทนำ

ในปัจจุบันพบว่าเทคโนโลยีใหม่ๆ มีบทบาทต่อการพัฒนาสังคม หรือแม้แต่การพัฒนาประเทศ เนื่องจากเทคโนโลยีเป็นพื้นฐานสำคัญอย่างหนึ่งในการพัฒนาทางด้านต่าง ๆ ได้แก่ ด้านการแพทย์ การศึกษา การสื่อสารคมนาคม และการรักษาความปลอดภัย เป็นต้น ซึ่งเทคโนโลยีที่ได้มีการพัฒนาขึ้นมาใหม่ๆ นี้ก็มีจำนวนมากและมีความหลากหลาย แต่ในที่นี้เราจะกล่าวถึงเทคโนโลยีที่กำลังได้รับความสนใจมากในตอนนี้คือ หุ่นยนต์ ซึ่งเราจะสังเกตเห็นได้จากสื่อหลาย ๆ แชนเนล และบุคคลทุกเพศทุกวัยให้ความสนใจเป็นจำนวนมาก ซึ่งหุ่นยนต์ที่มีในปัจจุบันก็มีการใช้งานอยู่หลายประเภท เช่น หุ่นยนต์ที่ใช้ในโรงงานอุตสาหกรรม หุ่นยนต์ที่เป็นของเล่น และหุ่นยนต์สำรวจ เป็นต้น

ประโยชน์ของหุ่นยนต์ก็จะขึ้นอยู่กับประเภทของหุ่นยนต์ โดยในที่นี้เราจะกล่าวถึงหุ่นยนต์สำรวจ ซึ่งจะมีประโยชน์ทางการช่วยเหลือหรือการสำรวจพื้นที่ที่เสี่ยงอันตรายซึ่งมนุษย์ไม่สามารถเข้าไปสำรวจได้ ในส่วนของช่วยเหลือหุ่นยนต์จะช่วยเหลือในด้านการค้นหา ดังนั้นหุ่นยนต์สำรวจจึงจำเป็นอย่างมากในปัจจุบัน ทำให้ต้องมีการวิจัยและพัฒนาหุ่นยนต์ประเภทนี้เป็นอย่างยิ่ง นอกจากประโยชน์ที่ได้รับดังที่กล่าวไว้ข้างต้นแล้ว หุ่นยนต์ประเภทนี้ ยังทำทาสความสามารถและต้องใช้ความรู้หลายๆ แชนเนลในการสร้าง อีกทั้งยังได้สัมผัสกับอุปกรณ์ที่เป็นเทคโนโลยีใหม่ ๆ จึงทำให้คณะผู้จัดทำมีความสนใจที่จะนำหุ่นยนต์สำรวจตัวนี้มาเป็นโครงงาน ซึ่งโครงงานนี้อาจจะยังไม่สามารถนำไปใช้งานจริงๆ ได้ เนื่องจากอุปกรณ์ที่นำมาสร้างยังไม่เหมาะสมกับสถานการณ์จริงๆ แต่ก็สามารถนำไปพัฒนาให้ใช้งานจริงๆ ได้เช่นกัน

#### 1.1 วัตถุประสงค์

1. เพื่อศึกษาหลักการและประยุกต์ใช้งานควบคุมมอเตอร์
2. เพื่อศึกษาหลักการและประยุกต์ใช้งานติดต่อระหว่างคอมพิวเตอร์กับอุปกรณ์อิเล็กทรอนิกส์
3. เพื่อศึกษาหลักการและประยุกต์ใช้งานการสร้างโครงสร้างตัวถังของหุ่นยนต์
4. เพื่อศึกษาหลักการและประยุกต์ใช้งานอุปกรณ์ที่เป็นเทคโนโลยีใหม่ๆ
5. เพื่อศึกษาวิธีการเขียน โปรแกรมควบคุมหุ่นยนต์ด้วยไมโครคอนโทรลเลอร์ MCS-51
6. เพื่อศึกษาวิธีการเขียน โปรแกรมด้วย Delphi

เอกสารนี้เป็นเอกสารที่เผยแพร่ในสื่อหรือการเขียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.2 ขอบเขตของโครงการ

หุ่นยนต์สำรวจจะสามารถเคลื่อนที่ได้ทุกสภาพผิวและยังสามารถเคลื่อนที่เข้าไปในพื้นที่ที่ไม่มีแสงสว่างได้ เนื่องจากการติดตั้งไฟส่องทางไว้ด้วย โดยจะมีการส่งภาพจากกล้องที่ติดตั้งอยู่บนหุ่นยนต์ไปแสดงบนจอมอนิเตอร์ของคอมพิวเตอร์ โดยจะสื่อสารข้อมูลกันแบบไร้สายระหว่างหุ่นยนต์กับคอมพิวเตอร์ ใช้การประมวลผลภาพในการควบคุมหุ่นยนต์แบบอัตโนมัติ และหาพิกัด 3 มิติ โดยใช้เทคนิคสเตอริโอสโคปิก (Stereo Scopic) และสามารถติดตามวัตถุได้

## 1.3 ประโยชน์ที่จะได้รับจากโครงการ

ประโยชน์จากโครงการนี้ คือได้นำทฤษฎีความรู้ที่ได้จากการศึกษามาประยุกต์ใช้งานจริงๆ เช่น การออกแบบวงจรต่างๆ การวิเคราะห์วงจร การประมวลผลภาพ เป็นต้น หรือจะเป็นการเขียนโปรแกรมควบคุม การทำงานของอุปกรณ์ต่างๆ ด้วยไมโครคอนโทรลเลอร์ MCS-51 การเขียนโปรแกรมด้วย Delphi ซึ่งเป็นโปรแกรมที่เราต้องศึกษาด้วยตัวเอง นอกจากนี้ยังได้ความรู้และประสบการณ์ใหม่ๆ ซึ่งจะทำให้เรามีความสามารถและทักษะเพิ่มขึ้น และยังสามารถนำไปใช้ในการทำงานในอนาคตได้อีกด้วย

## 1.4 รายละเอียดโดยย่อของโครงการ

หุ่นยนต์สำรวจ หุ่นยนต์แบบดินตะขาบ เพื่อเคลื่อนที่ในพื้นที่ ที่มีสิ่งขีดขวางได้ โดยมีขนาด กว้างXยาวXสูง = 50\*65\*28 cm. โดยมีน้ำหนักประมาณ 25 กิโลกรัม หุ่นยนต์จะขับเคลื่อนโดยใช้มอเตอร์ 2 ตัว สังกาถึงโดยใช้โซ่และเฟือง ส่วนของกล้องเป็นกล้อง CCD ที่เป็นภาพสี โดยจะต่อเข้ากับการ์ดจับภาพ (Capture Card) ภายในคอมพิวเตอร์และการส่งสัญญาณจะส่งโดยใช้ตัวส่งวิดีโอ (Video Sender) มาแสดงภาพที่คอมพิวเตอร์ ในการควบคุมการเคลื่อนที่ของหุ่นยนต์จะควบคุมโดยส่ง ข้อมูลผ่านพอร์ทอนุกรม และส่งไปยังหุ่นยนต์แบบไร้สายโดยจะผ่านโมดูลเรดิโอแมทริกซ์ (Radio Matrix) ซึ่งใช้ความถี่ในการส่งคือ 2.4 GHz ซึ่งมีการส่งแบบฮาร์ฟดูเพล็กซ์ (Half Duplex) ส่วนของการตรวจ และส่วนของการหาตำแหน่งของวัตถุจะใช้ การประมวลผลภาพ

### 1.5 รายละเอียดของปฏิญานิพนธ์

โครงการ “หุ่นยนต์สำรวจ” เป็นโครงการที่รวมเอาความรู้หลาย ๆ แขนงมาใช้สร้างโครงการนี้ จึงทำให้รายละเอียดของเนื้อหา และทฤษฎีที่เกี่ยวข้องของโครงการมีมากพอสมควร โดยจะมีการแบ่งเนื้อหาออกเป็นบทต่างๆ ดังต่อไปนี้

บทที่ 1 บทนำ จะกล่าวถึงรายละเอียดความเป็นมา, ขอบเขต และรายละเอียดโดยย่อของปฏิญานิพนธ์

บทที่ 2 ทฤษฎีและหลักการ โดยจะกล่าวถึงทฤษฎีและหลักการพื้นฐานต่าง ๆ ที่นำมาใช้ในการออกแบบ และแสดงให้เห็นถึงความสัมพันธ์ในการทำงานของอุปกรณ์ในแต่ละส่วน

บทที่ 3 การออกแบบทางด้านวงจรควบคุมและโครงสร้างหุ่นยนต์ (Hardware) โดยจะกล่าวถึงขั้นตอนในการออกแบบส่วนต่างๆ ของวงจรควบคุมและ โครงสร้างตัวถังของหุ่นยนต์

บทที่ 4 ผลการทดลอง โดยจะกล่าวถึงผลจากการทดสอบวงจรและโปรแกรมต่าง ๆ ว่ามีการทำงานเป็นอย่างไรบ้าง

บทที่ 5 สรุปผลการทดลองและแนวทางการพัฒนาต่อ กล่าวถึงปัญหาที่พบในการทำงานเพื่อนำไปพัฒนาและปรับปรุงต่อไปในอนาคต

## บทที่ 2

### ทฤษฎีและหลักการ

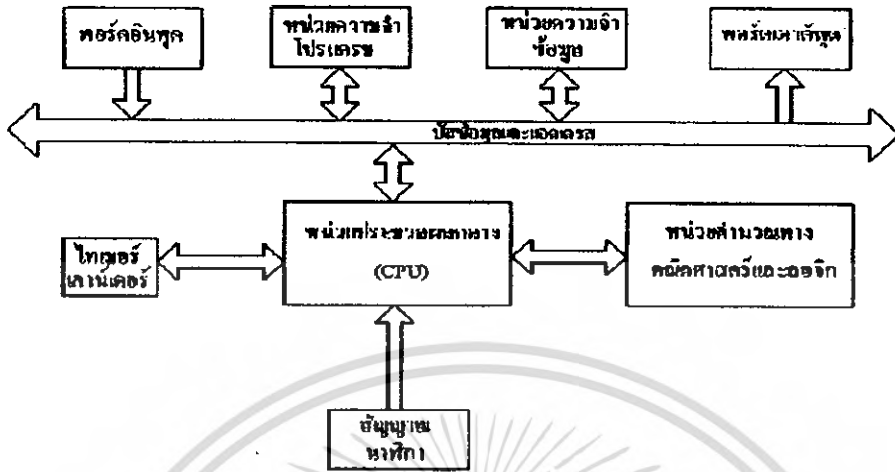
#### 2.1 ไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์ MCS-51 เป็นไมโครคอนโทรลเลอร์ขนาด 8 บิตที่มี อุปกรณ์สนับสนุนประกอบอยู่ภายในหลายอย่าง ได้แก่ หน่วยความจำสำหรับเก็บข้อมูล หน่วยความจำสำหรับเก็บโปรแกรม ตัวตั้งเวลา/ตัวนับ อุปกรณ์รับส่งข้อมูลแบบอนุกรม เนื่องจากโครงสร้างของไมโครคอนโทรลเลอร์มีอุปกรณ์สนับสนุนประกอบอยู่ภายในนี้เอง ทำให้การใช้งานง่ายขึ้น และมีประสิทธิภาพมากขึ้น โดยไม่ต้องมีการเชื่อมต่ออุปกรณ์ภายนอกเพิ่มเติมมากเหมือนกับตัวไมโครโปรเซสเซอร์ทั่วไป นอกจากนี้หากเราต้องการใช้งานไมโครคอนโทรลเลอร์ร่วมกับอุปกรณ์อื่นเพิ่มเติม เช่น ไอซี 8255 หรือหน่วยความจำภายนอก เรายังสามารถนำมาเชื่อมต่อเพิ่มเติมเข้ากับไมโครคอนโทรลเลอร์ได้อีกด้วย

##### 2.1.1 โครงสร้างภายในของไมโครคอนโทรลเลอร์ MCS-51

โครงสร้างภายในพื้นฐานของไมโครคอนโทรลเลอร์ MCS-51 แสดงในรูปที่ 2.1 ประกอบด้วยอุปกรณ์ต่างๆ ดังนี้

- หน่วยประมวลผลกลางขนาด 8 บิต
- หน่วยประมวลผลสำหรับข้อมูลแบบบิต (BOOLEAN PROCESSOR)
- ความสามารถในการอ้างตำแหน่งของหน่วยความจำโปรแกรม 64 กิโลไบต์
- ความสามารถในการอ้างตำแหน่งของหน่วยความจำข้อมูล 64 กิโลไบต์
- หน่วยความจำโปรแกรมภายในขนาด 4 กิโลไบต์ แบบ อิพรอม (เบอร์ 8451)
- หน่วยความจำแบบ แรม ภายในจำนวน 128 ไบต์
- พอร์ตอินพุต/เอาต์พุตแบบขนานจำนวน 32 เส้น ซึ่งสามารถแยกทำงานได้อย่างอิสระ
- วงจรรนับ/จับเวลาขนาด 16 บิต จำนวนสองวงจร
- วงจรสื่อสารแบบอนุกรมแบบคู่เหล็กเต็ม(FULL DUPLEX)
- วงจรรควบคุมการอินเตอร์รัปต์จากแหล่งกำเนิดสัญญาณ 6 ประเภท พร้อมการกำหนดลำดับวงจรมลิตสัญญาณนาฬิกาภายในซึ่ง โครงสร้างการทำงานทั้งหมดของไมโครคอนโทรลเลอร์จะ อาศัยหลักการทำงานที่เกี่ยวข้องกัน โดยอาศัยหลักการทำงานที่เป็นไปตาม โครงสร้างเสมอ



รูปที่ 2.1 แสดงโครงสร้างภายในของไมโครคอนโทรลเลอร์ MCS-51

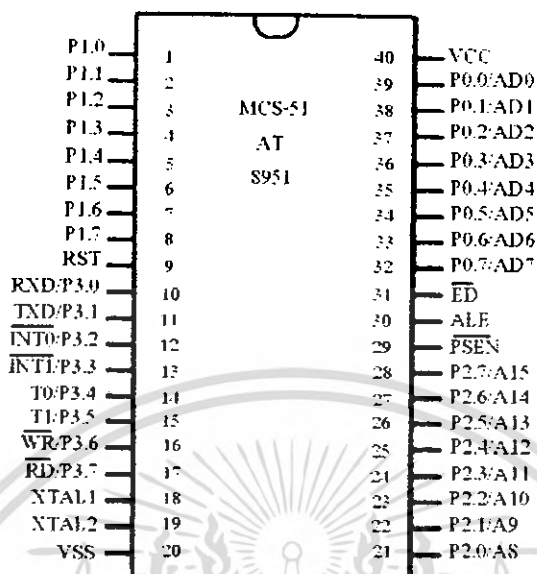
โดยมากแล้วไมโครคอนโทรลเลอร์ตระกูลนี้มักจะมีรูปร่างของไอซีเป็นแบบขนาด 40 ขา ดังแสดงในรูปที่ 2.2 ซึ่งแต่ละขาสัญญาณจะมีหน้าที่ที่ระบุชัดเจนตามสัญลักษณ์ชื่อขาคู่ที่กำกับในแต่ละขา อย่างไรก็ตามจะมีบางขาสัญญาณที่อาจจะมีหน้าที่ได้มากกว่าหนึ่งอย่าง (ซึ่งเขียนกำกับไว้ว่า ALTERNATE FUNCTION ในรูปที่ 2.2) ซึ่งจะไม่สามารถใช้งานในเวลาเดียวกันได้ ตัวอย่างเช่น ขาสัญญาณบิต 0 ของพอร์ต 3 (ใช้ตัวย่อเป็น P3.0) อาจจะใช้เป็นขาสัญญาณเอาต์พุต หรืออินพุตตามปกติ ภายในไมโครคอนโทรลเลอร์ MCS-51 ซึ่งประกอบด้วยหน่วยการทำงานต่างๆ ภายในไอซี MCS-51 จำนวนมาก โดยแต่ละบล็อกซึ่งเป็นวงจรควบคุมรีจิสเตอร์ (REGISTER) หรือหน่วยความจำภายในของไอซี MCS-51 จะถูกเชื่อมต่อเข้าด้วยกันผ่านทางเส้นสัญญาณที่เรียกว่า บัสข้อมูลภายใน รีจิสเตอร์และหน่วยความจำเหล่านี้จะถูกนำไปใช้ระหว่างการประมวลผลคำสั่งหน้าที่ของโปรแกรมที่ผู้ใช้สร้างขึ้นมาก็เป็นการควบคุมการรับหรือส่งข้อมูลระหว่างรีจิสเตอร์เหล่านี้ ซึ่งอาจจะมีการดำเนินการร่วมกับหน่วยการดำเนินงานประมวลผลทางคณิตศาสตร์และลอจิก หรือเรียกว่า ARITHMETIC AND LOGIC UNIT :ALU

### 2.1.2 โครงสร้างหน่วยความจำภายในไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์ MCS-51 แยกการจัดการหน่วยความจำออกเป็นสองส่วนอย่างชัดเจน คือ หน่วยความจำโปรแกรม (PROGRAM MEMORY) และหน่วยความจำข้อมูล (DATA MEMORY) หน่วยความจำทั้งสองนี้ มีหน้าที่แตกต่างกัน และใช้วิธีการอ้างแอดเดรส

สัญญาณการติดต่อแยกออกจากกัน

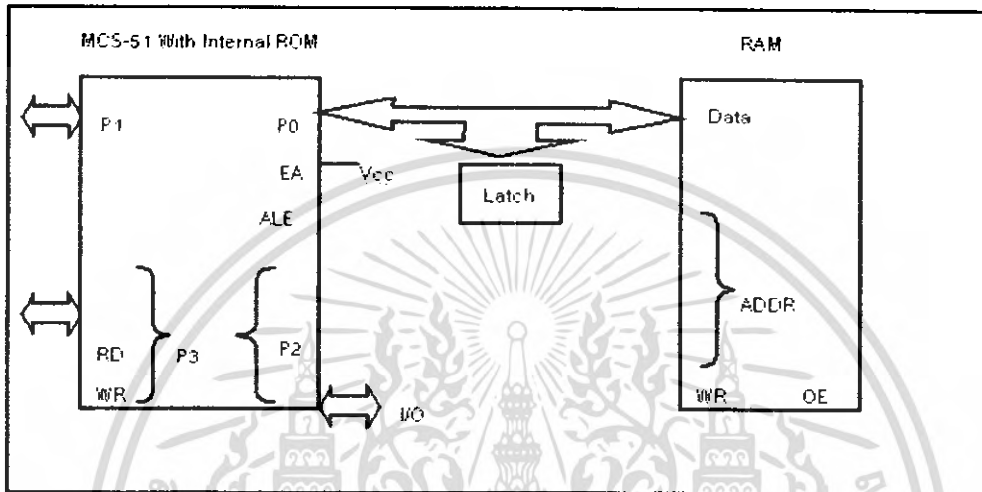
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



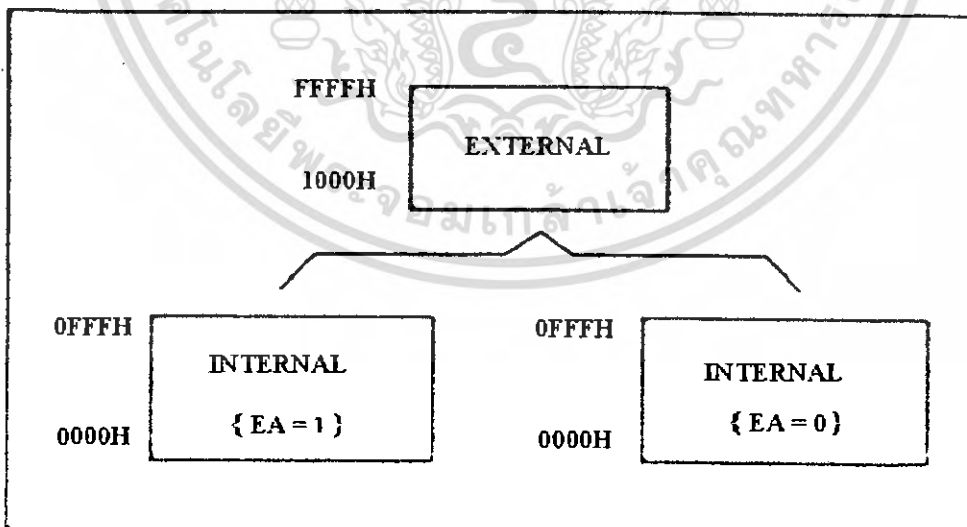
รูปที่ 2.2 แสดงรูปร่างและการจัดวางขาต่างๆ ของไมโครคอนโทรลเลอร์ MCS-51

### 2.1.3 หน่วยความจำโปรแกรม

หน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ MCS-51 เป็นบริเวณหน่วยความจำสำหรับเก็บข้อมูลและคำสั่งใช้งานต่างๆ ซึ่งแม้ว่าจะไม่มีการจ่ายกระแสไฟฟ้าให้กับระบบข้อมูลเหล่านี้ก็ยังคงอยู่ไม่สูญหายโครงสร้างของหน่วยความจำโปรแกรมมีลักษณะเช่นเดียวกับหน่วยความจำที่บรรจุอยู่ในไมโครคอนโทรลเลอร์ MCS-51 ของหน่วยความจำประเภทต่างๆ เช่น หน่วยความจำแบบรอม (READ ONLY MEMORY) หรืออีพรอม (ERASABLE PROGRAMABLE READ ONLY MEMORY) ในไมโครคอนโทรลเลอร์ MCS-51 สามารถอ่านข้อมูลหน่วยความจำโปรแกรมนี้ได้สูงสุดไม่เกิน 64 กิโลไบต์ และแยกประเภทของหน่วยความจำโปรแกรมเป็น 2 ลักษณะ ตามตำแหน่งของหน่วยความจำนั้นคือ หน่วยความจำโปรแกรมภายใน (INTERNAL PROGRAM MEMORY) ซึ่งเป็นหน่วยความจำรอม หรืออีพรอม ที่อยู่ภายในตัวไอซีของไมโครคอนโทรลเลอร์เอง และหน่วยความจำโปรแกรมภายนอก (EXTERNAL PROGRAM MEMORY) ซึ่งเป็นการใช้อีซีหน่วยความจำมาทำหน้าที่เป็นหน่วยความจำโปรแกรมของระบบ



รูปที่ 2.3 แสดงการใช้หน่วยความจำสำหรับเก็บ โปรแกรม

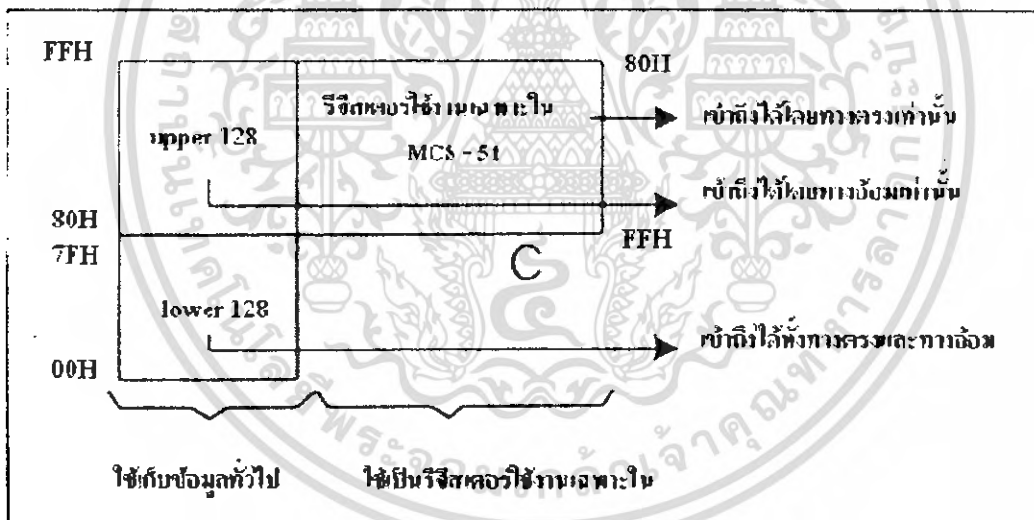


รูปที่ 2.4 แสดงการจัดพื้นที่ของหน่วยความจำโปรแกรมภายในและภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไมโครคอนโทรลเลอร์เบอร์ต่างๆ ของตระกูล 8051 นี้สามารถขยายให้ใช้งานในหน่วยความจำภายนอกได้ทั้งสิ้น โดยกรณีที่มีหน่วยความจำโปรแกรมภายในอยู่แล้วการอ้างตำแหน่งแอดเดรสที่มีทั้งในหน่วยความจำโปรแกรมภายใน และภายนอกนั้นจะต้องทำการควบคุมระดับลอจิกของสัญญาณ ในขณะที่ค้วขนาดหน่วยความจำโปรแกรมภายในของไมโครคอนโทรลเลอร์เบอร์ต่างๆ ภายในตระกูล 8051 จะแตกต่างกันออกไป เพื่อความเหมาะสมกับการนำไปใช้งานลักษณะต่างๆ

- 8051 และ 8052 มีหน่วยความจำแบบรอม 4 และ 8 กิโลไบต์
- 8751 มีหน่วยความจำแบบ อีพรอม ขนาด 4 กิโลไบต์ข้อมูลที่จัดเก็บภายในนี้ ซึ่งสามารถใช้แสงอุลตราไวโอเลตลบ และนำกลับไปบรรจุโปรแกรมใหม่ได้อีกครั้งหนึ่ง
- 8031 และ 8032 ไม่มีหน่วยความจำโปรแกรมอยู่ภายในตัวไอซี ดังนั้นในการนำไปใช้งานจึงจำเป็นต้องอาศัยหน่วยความจำโปรแกรมภายนอกเสมอ



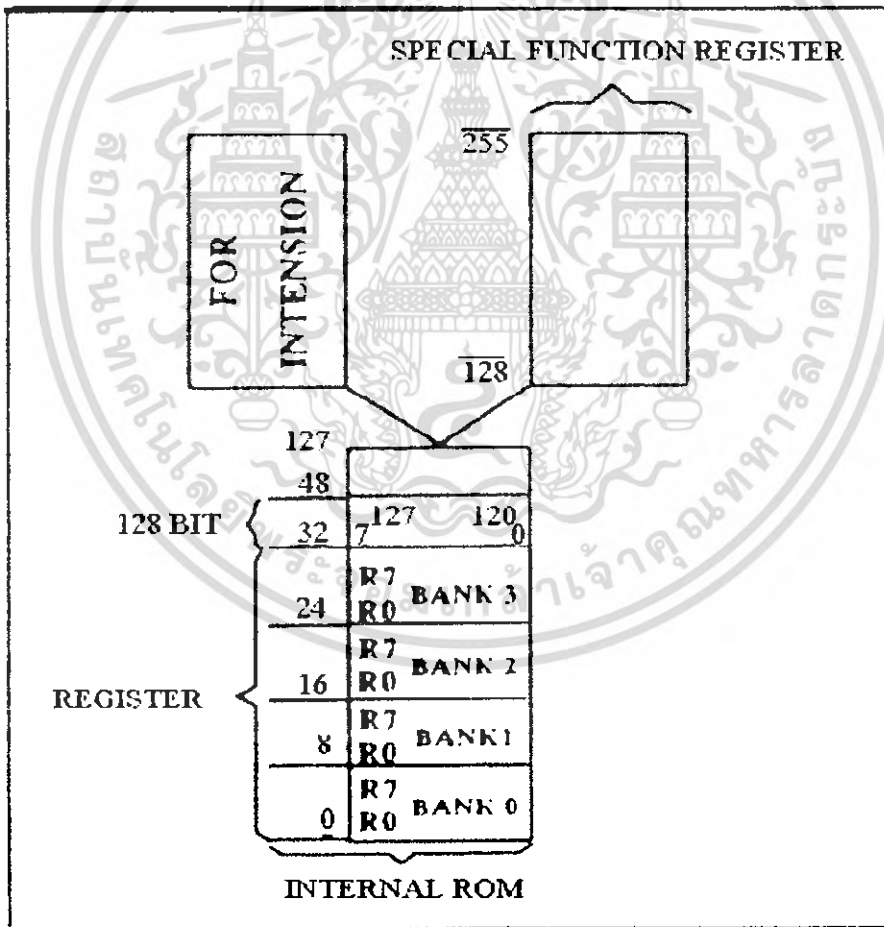
รูปที่ 2.5 แสดงหน่วยความจำสำหรับเก็บข้อมูลภายในไมโครคอนโทรลเลอร์ MCS-51

### 2.1.4 หน่วยความจำข้อมูล

หน่วยความจำข้อมูล (DATA MEMORY) ซึ่งโดยพื้นฐานแล้วเป็นหน่วยความจำแรม สามารถเขียนหรืออ่านข้อมูลได้ (READ OR WRITE MEMORY) ใช้สำหรับเก็บข้อมูลหรือตัวแปรที่เกิดขึ้นในขณะที่กำลังประมวลผลโปรแกรมไว้เป็นการชั่วคราว ซึ่งโดยพื้นฐานแล้ว

หน่วยความจำข้อมูลจัดเป็นหน่วยความจำแรมแบบสแตค ดังนั้นเมื่อไม่มีการจ่ายไฟให้กับไมโครคอนโทรลเลอร์ ข้อมูลที่เก็บไว้จะหายไป และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

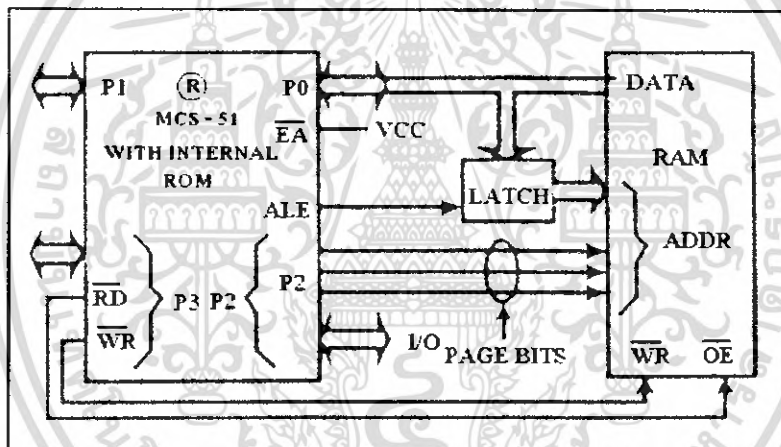
ระบบก็จะมีผลทำให้ข้อมูลที่จัดเก็บไว้ภายในหน่วยความจำนี้สูญไป พื้นที่ของหน่วยความจำข้อมูลของไมโครคอนโทรลเลอร์ MCS-51 มีได้สูงสุดไม่เกิน 64 กิโลไบต์ และแยกประเภทออกเป็นสองลักษณะตามตำแหน่งที่ตั้งของหน่วยความจำนั้น ตามลักษณะของหน่วยความจำโปรแกรมภายใน ซึ่งก็เป็นแรมที่อยู่ภายในตัวไอซีในตระกูลของไมโครคอนโทรลเลอร์ และหน่วยความจำข้อมูลภายนอกซึ่งเป็นการใช้อีซีหน่วยความจำแรมมาเพิ่ม เต็มเข้าไปในวงจรลักษณะเดียวกับ การนำไอซีอีพรอมมาใช้งานเป็นหน่วยความจำโปรแกรมนั่นเอง



รูปที่ 2.6 แสดงการจัดหน่วยความจำข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่หน่วยความจำสำหรับเก็บข้อมูลของไมโครคอนโทรลเลอร์ MCS-51 นี้สามารถแบ่งออกเป็น 2 ส่วนคือ ในส่วนที่เป็นหน่วยความจำสำหรับเก็บข้อมูลภายในไอซี และหน่วยความจำสำหรับเก็บข้อมูลภายนอกไมโครคอนโทรลเลอร์ MCS-51 ทุก ๆ เบอร์จะมีหน่วยความจำเก็บข้อมูลทั่ว ๆ ไป ภายในไอซีอย่างน้อยคือ 128 ไบต์ ไปจนถึง 256 ไบต์ ทั้งนี้ขึ้นกับเบอร์ของไอซี หน่วยความจำสำหรับเก็บข้อมูลภายใน ไอซีในบริเวณ 128 ไบต์เรียกว่า LOWER 128 และในบริเวณ 128 ไบต์หลัง ที่มีเพิ่มในบางเบอร์มีชื่อเรียกว่า UPPER

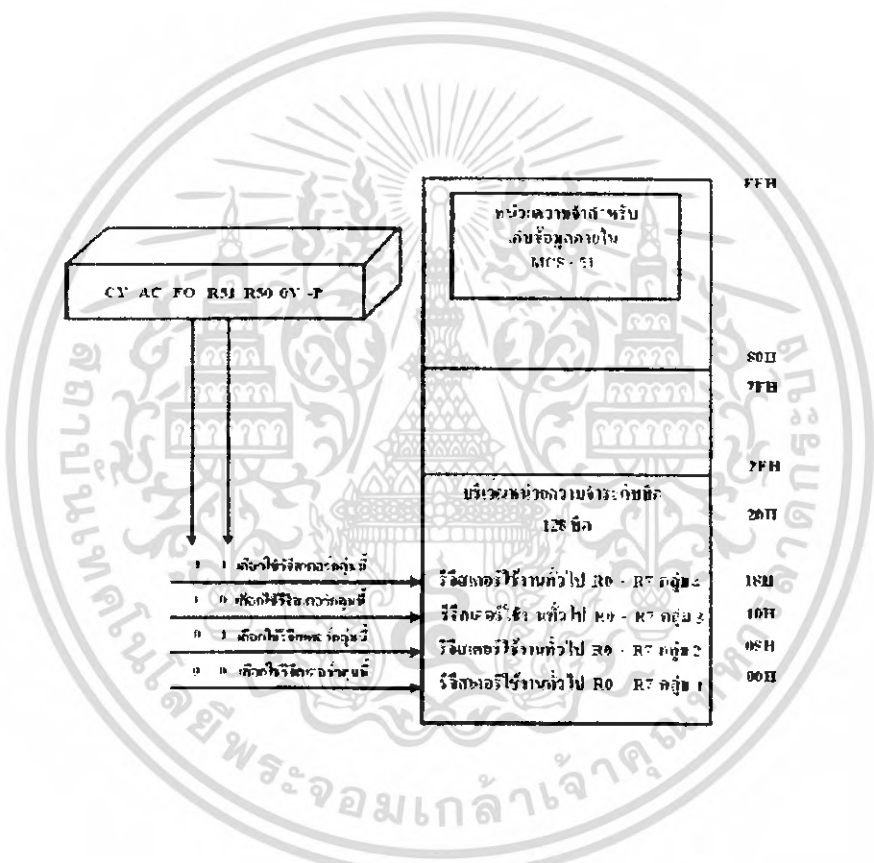


รูปที่ 2.7 แสดงการต่อกับหน่วยความจำข้อมูลภายนอกไอซี

### 2.1.5 รีจิสเตอร์ที่เกี่ยวข้องกับไมโครคอนโทรลเลอร์ MCS-51

รีจิสเตอร์ในกลุ่มนี้ จะเป็นรีจิสเตอร์ขนาด 16 บิตที่ใช้งาน เพื่อเก็บข้อมูลของตัวแอดเดรส เป็นสำคัญ โดยค่าที่อยู่ภายในแอดเดรสนี้จะนำไปเป็นค่าของข้อมูลที่ส่งออกไปทางบัสแอดเดรส ใน ส่วนของไมโครคอนโทรลเลอร์ เพื่อบอกตำแหน่งที่ต้องการติดต่อ รีจิสเตอร์ที่จัดในกลุ่มนี้ ประกอบด้วยรีจิสเตอร์ใช้งานทั่วไป (GENERAL-PURPOSE REGISTERS) รีจิสเตอร์ในกลุ่มนี้ จัดเป็นพื้นที่หน่วยความจำที่ใช้ในการสนับสนุนในการประมวลผล การทำงานจากหน่วยประมวลผลทางคณิตศาสตร์และลอจิก (ALU) เพื่อให้สามารถจัดการข้อมูลให้เร็วที่สุด นอกจากนี้ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ไม่ได้ใช้คำสั่งเหล่านี้ก็ยังสามารถเก็บข้อมูลตัวแปรภายในโปรแกรม จะเห็นได้ว่าชื่อของรีจิสเตอร์ไม่ว่าจะอยู่ในรีจิสเตอร์แบบกาคิ ก็จะมีชื่อว่า R0 ถึง R7 เหมือนกันทั้งสิ้น ดังนั้นในการใช้งานผู้ใช้งานจะต้องให้ความระมัดระวังว่า ต้องการรีจิสเตอร์นั้นๆ จากแบบกาคิๆ ซึ่งการกำหนดเลือกแต่ละกลุ่มของรีจิสเตอร์นี้ก็ทำได้ง่าย เพียงการกำหนดค่าของบิตที่อยู่ภายในแฟลช (PSW) เท่านั้นอย่างไรก็ตาม โดยทั่วไปก็มักจะมีการใช้งานรีจิสเตอร์ R0 ถึง R7 เฉพาะในแบงก์ 0 เท่านั้น ดังนั้นพื้นที่ของแบงก์อื่นๆ ที่เหลือก็สามารถนำมาใช้ในลักษณะของหน่วย ความจำแรม



รูปที่ 2.8 แสดงการเลือกใช้รีจิสเตอร์ใช้งานทั่วไป R0-R7 ในแต่ละกลุ่ม

- รีจิสเตอร์หน้าที่พิเศษ เป็นรีจิสเตอร์หน้าที่พิเศษ (SFR) เป็นรีจิสเตอร์สำหรับการควบคุมหน้าที่ และการทำงานของอุปกรณ์หรือพอร์คของ ไมโครคอนโทรลเลอร์ MCS-51 ทั้งหมด ตำแหน่งของรีจิสเตอร์เหล่านี้จะจัดอยู่ในบริเวณแอดเดรส 80H-FFH การใช้งานรีจิสเตอร์หน้าที่พิเศษเหล่านี้สามารถทำได้ทั้งการระบุชื่อของรีจิสเตอร์ หรือตำแหน่งแอดเดรส ที่เป็นของรีจิสเตอร์นั้นก็ได้การจัดพื้นที่หน่วย ความจำสำหรับรีจิสเตอร์หน้าที่พิเศษเหล่านี้ โดยมีข้อสังเกตว่า

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการสงวนสิทธิ์ในเนื้อหา เมื่อผู้ใช้งานเห็นข้อบกพร่องหรือข้อผิดพลาดในการใช้งาน ไม่ควรแก้ไขใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รีจิสเตอร์ที่อยู่ในตำแหน่งแอดเดรสที่มีจำนวนเป็นทวีคูณของค่า 8 จะสามารถอ้างถึงในระดับบิตได้ด้วย (นั่นคือแอดเดรส 80H 88H 90H A0H A8H B0H B8H D0H E0H และ F0H)

- แอควิวมูเลเตอร์ (ACCUMULATOR) หรือ ACC เป็นรีจิสเตอร์ขนาด 8 บิต ทำหน้าที่ในการเก็บข้อมูลที่จะส่งให้กับหน่วยงานภายในหน่วยประมวลผลกลาง และเก็บผลลัพธ์ที่ได้จากการทำงานเท่านั้น การทำงานของรีจิสเตอร์นั้นมีลักษณะเช่นเดียวกับตัวแอควิวมูเลเตอร์ของโปรเซสเซอร์ทั่วไป การใช้งานในโปรแกรมซึ่งใช้เรียกเป็น รีจิสเตอร์ A

### 2.1.6 ชุดคำสั่งของไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์ MCS-51 ประกอบด้วยคำสั่งทั้งหมดจำนวนมาก ซึ่งสามารถจะจัดกลุ่มคำสั่งเหล่านี้ตามลักษณะและหน้าที่การทำงานที่คล้ายคลึงกัน เพื่อความสะดวกต่อการศึกษาทำความเข้าใจและใช้งาน ดังนี้

2.1.6.1 กลุ่มการถ่ายเทข้อมูล คือ กลุ่มคำสั่งในการโอนย้ายข้อมูล ทำหน้าที่ในโอนย้ายข้อมูลระหว่างรีจิสเตอร์ หรือหน่วยความจำภายในแรม โดยมีรายละเอียดดังนี้ ชุดคำสั่งในการถ่ายเทแรม ภายในนั้น ซึ่งเวลาที่ใช้ในหนึ่งคำสั่งนั้นจะเป็นเวลา เมื่อขณะที่ความถี่ในการทำงานของหน่วยประมวลผลกลางที่ความถี่ 12 เมกะเฮิร์ตซ์ และรายละเอียดของแต่ละคำสั่งมีดังนี้ MOV :จะทำงานในลักษณะเป็นการถ่ายเทข้อมูลที่มีขนาดเป็น ไบต์ หรือ บิตก็ได้ จากแหล่งกำเนิดเข้าสู่ตัวรับข้อมูลในฟิลด์ โอเปอร์แรนด์ PUSH:จะทำงานโดยเพิ่มค่ารีจิสเตอร์ SP ก่อนแล้วจึงทำการถ่ายเทข้อมูล 1 ไบต์จากแหล่งกำเนิดไปบริเวณสแต็คตามตำแหน่งที่รีจิสเตอร์ SP กำหนด POP:การถ่ายเทข้อมูลขนาด 1 ไบต์ จากบริเวณตำแหน่งที่รีจิสเตอร์ SP กำหนดไปยังรีจิสเตอร์ที่โอเปอร์แรนด์ กำหนดและหลังจากนั้น รีจิสเตอร์ SP จะลดค่าลง XCH:คำสั่งแลกเปลี่ยนไบต์ระหว่างแหล่งกำเนิดโอเปอร์แรนด์กับรีจิสเตอร์ AXCHD คำสั่งในการแลกเปลี่ยนขนาดนิบเบิลทางอันดับต่ำของแหล่งกำเนิดโอเปอร์แรนด์กับนิบเบิลอันดับต่ำของแอควิวมูเลเตอร์ ตัวอย่างเช่น ทำการเลื่อนข้อมูลไป 2 ไบต์ทางขวามือ ซึ่งจะมี 2 วิธีคือ ใช้คำสั่ง MOV หรือใช้คำสั่ง XCH รายละเอียดการใช้คำสั่งทั้ง 2 แบบ

2.1.6.2 กลุ่มคำสั่งทางคณิตศาสตร์ เช่น การบวก ลบ คูณ และหารข้อมูลภายในตัวรีจิสเตอร์ต่างๆ ช่วงเวลาการทำงานของแต่ละคำสั่งนั้นจะกำหนดที่ความถี่ของสัญญาณนาฬิกาที่ 12 เมกะเฮิร์ตซ์ คำสั่งทางคณิตศาสตร์ส่วนใหญ่ใช้เวลา 1 ms ยกเว้นคำสั่ง INC DPTR ซึ่งใช้เวลา 2 ms โดยที่คำสั่งการคูณและหารใช้เวลา 4 ms โดยมีรายละเอียดการใช้คำสั่งมีดังนี้ INC:เป็นการบวกหนึ่งกับโอเปอร์แรนด์และใส่ค่าใหม่กลับเข้าที่ตัวโอเปอร์แรนด์นั้น ๆ DEC:เป็นการลบออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตัวเลขที่อยู่ในแหล่งกำเนิด โอเปอร์เรนด์ และนำผลลัพธ์ที่ได้มาเก็บไว้ในตัวโอเปอร์เรนด์นั้น ADD:เป็นการบวกในแอกคิวมูลเตอร์เข้ากับค่าในแหล่งกำเนิด โอเปอร์เรนด์ ADDC:เป็นการบวกค่าต่างๆ ในแอกคิวมูลเตอร์เข้ากับค่าในแหล่งกำเนิด โอเปอร์เรนด์และบวกกับบิตทดด้วย SUBB: เป็นการนำเลขที่แหล่งกำเนิด โอเปอร์เรนด์ ลบออกจากตัวเลขใน A และนำค่าบิตตัวทดมาลบออกอีกและผลลัพธ์ที่ได้นำมาใส่ลงในแอกคิวมูลเตอร์ A MUL:เป็นการคูณแบบ ไม่คิดตัวเครื่องหมายของตัวเลขที่อยู่ใน แอกคิวมูลเตอร์กับ เลขในรีจิสเตอร์ B แล้วได้ผลลัพธ์ 2 ไบต์ นำมาเก็บไว้ใน AB โดย A จะรับอันดับต่ำ ส่วน B จะรับอันดับสูง DIV:เป็นคำสั่งในการหารแบบ ไม่คิดเครื่องหมายที่อยู่ในแอกคิวมูลเตอร์แล้วหารตัวเลขในรีจิสเตอร์ B แล้วนำผลลัพธ์ไปเก็บในแอกคิวมูลเตอร์ และเศษของการหารตัวเลขจะเก็บไว้ในรีจิสเตอร์ B DA:สำหรับการบวกกันทางตัวเลข BCD เป็นการปรับค่ารวมซึ่งเป็นผลมาจากการบวกกันทางไบนารีของระบบตัวเลข BCD ขนาด 2 หลักสองจำนวนการปรับค่าตัวเลขผลรวมด้วยการใช้คำสั่ง DA จะได้ผลลัพธ์กลับมายังแอกคิวมูลเตอร์ รายละเอียดการใช้คำสั่ง

2.1.6.3 กลุ่มคำสั่งทางตรรกศาสตร์หรือแบบลอจิก ทำหน้าที่เกี่ยวกับการประมวลผลแบบลอจิก ต่างๆ เช่นการ AND OR หรือ EX-OR ระหว่างข้อมูลในรีจิสเตอร์ A นั่นเองโดยมีการใช้คำสั่งดังนี้ CPL:เป็นการใช้คำสั่งกลับค่าหรือคอมพลีเมนต์ ข้อมูลในแอกคิวมูลเตอร์จะ ไม่มีผลใดๆ ต่อค่าของแฟล็ก หรือการอ้างถึงตำแหน่งแอดเดรสนั้นตามบิตนั้นๆ RL, RLC, RR, RRC, SWAP: ทั้ง 5 คำสั่งนี้เป็นคำสั่งในการทำงานการวนบิตบนตัวของแอกคิวมูลเตอร์ ซึ่ง RL เป็นการวนบิตทางขวา, RLC เป็นการทำการวนทางซ้ายผ่านบิตทด, RRC เป็นการวนขวาผ่านบิตทด และ SWAP เป็นการวนซ้ายสี่ครั้ง ANL:เป็นการ ADD กันทางตรรกศาสตร์ ระหว่างแหล่งกำเนิดสองโอเปอร์เรนด์ ซึ่งจะสั่งให้ทำงานในรูปแบบของตรรกศาสตร์ทางข้อมูลขนาดเป็น ไบต์หรือบิตกลุ่มคำสั่งแบบบูลีนหรือแบบบิตซึ่งจะเป็นความสามารถ ของไมโครคอนโทรลเลอร์ MCS-51 ที่จะดำเนินการประมวลผลแบบบิตแทนที่จะเป็นข้อมูล ทั้งไบต์เช่นปกติ โดยมีชุดคำสั่งที่จัดการโดยตรง ทุกคำสั่งจะเข้าถึงข้อมูลโดยตรงในระดับบิต โดยมีการบิตแอดเดรสได้ตั้งแต่ 00H - 7FH ในพื้นที่ 128 บิต หน่วยความจำข้อมูลภายในและบิตแอดเดรส 80H-FFH ในบริเวณกลุ่มรีจิสเตอร์ฟังก์ชันพิเศษ (SFR)

2.1.6.4 กลุ่มคำสั่งในการกระโดดไปยังตำแหน่งต่างๆภายในโปรแกรม ซึ่งจะเปลี่ยนลำดับของการประมวลผลภายในโปรแกรมไปยังส่วนต่างๆแทนที่จะดำเนินการไปเป็นลำดับ ต่อเนื่อง โดยที่คำสั่ง JMP จะแบ่งเป็น 3 ลักษณะ คือ SJMP, LJMP, AJMP ซึ่งในแต่ละคำสั่งจะมีข้อแตกต่างของการกระโดด ไปยังแอดเดรสไกลสุดที่ต่างกันคำสั่ง JMP ซึ่งเป็นแบบโมนิติกที่สามารถจะใช้ได้ โดยมีรายละเอียดการใช้งานของคำสั่งดังต่อไปนี้ SMP:จะเป็นการกระโดดแบบการย้ายอันดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตำแหน่งของแอดเดรสตำแหน่งเดิมซึ่งจะสามารถกระโดดได้ -128 ถึง +127 ไบต์ AJMP: ลักษณะแบบนี้จะสามารถกระโดดได้ไกลสุดประมาณ 2 กิโลไบต์ ซึ่งจะใช้น้อยความจำเพียง 2 ไบต์เท่านั้น ในการกำหนด LJMP: ลักษณะแบบนี้จะสามารถกระโดดได้ไกลสุดประมาณ 64 กิโลไบต์ ซึ่งจะใช้น้อยความจำเพียง 3 ไบต์เท่านั้นในการกำหนด JMP @A+DPTR: เป็นการควบคุมการกระโดดไปยังโปรแกรมที่ต้องการเฉพาะภายในส่วนต่างๆ

### 2.1.7 โครงสร้างการอินเทอร์รัปต์ของไมโครคอนโทรลเลอร์ MCS-51

จากแผนภาพโครงสร้างระบบอินเทอร์รัปต์ของไมโครคอนโทรลเลอร์ MCS-51 สัญญาเข้ามามีการอินเทอร์รัปต์ MCS-51 นั้นเกิดขึ้นได้ 5 ลักษณะตามตารางข้อมูล ในรูปที่ 2.6 โดยจะเห็นว่าสามารถที่จะกำหนดเลือกเพื่ออินฮอม (หรืออินาเบิล : ENABLE) และห้าม (หรือดิสเอเบิล: DISABLE) ไม่ให้มีการอินเทอร์รัปต์แต่ละประเภทได้ โดยการกำหนดบิตของข้อมูลที่เกี่ยวข้อง ซึ่งมักจะอยู่ภายในรีจิสเตอร์ TCON และ SCON นอกจากนี้ยังมีตำแหน่งบิตภายในรีจิสเตอร์ IE (INTERRUPT ENABLE REGISTER) ซึ่งทำหน้าที่เสมือนกับเป็นสวิตช์หลักที่เกี่ยวข้องกับสัญญาณอินเทอร์รัปต์ทั้งหมด หากว่ากำหนดไม่ให้เกิดการอินเทอร์รัปต์ แล้วการกำหนดบิตเพื่อห้ามหรืออินฮอมของแต่ละอินเทอร์รัปต์ก็จะมีผลใดๆ เกิดขึ้นยังแสดงให้เห็นว่าสัญญาณอินเทอร์รัปต์ แต่ละประเภทยังสามารถกำหนดระดับความสำคัญ (PRIORITY) ของการอินเทอร์รัปต์ได้สองลักษณะ คือระดับความสำคัญสูงหรือต่ำ (HIGH OR LOW PRIORITY) กล่าวคือ ขณะที่กำลังประมวลผลอยู่ภายในส่วนของโปรแกรมย่อยบริการอินเทอร์รัปต์ ของสัญญาณที่มีระดับความสำคัญต่ำอยู่ ก็อาจจะถูกขัดจังหวะให้ไปประมวลผลของสัญญาณอินเทอร์รัปต์ที่มีระดับความสำคัญสูงกว่า แต่หากว่าเป็นสัญญาณอินเทอร์รัปต์ที่มีระดับความสำคัญต่ำ เช่นเดียวกันแล้วก็ต้องรอให้เสร็จสิ้นการประมวลผลที่ ดำเนินการอยู่ก่อน

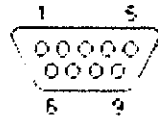
### 2.1.8 การรีเซต

โดยความหมายของการรีเซตเป็นการบังคับให้มีการเริ่มต้นใหม่อีกครั้งหนึ่ง ซึ่ง มักจะกระทำโดยการกำหนดสถานะของสัญญาณที่รีเซตของไอซี MCS-51 ให้เป็นระดับลอจิกที่เหมาะสม เท่านั้น การรีเซตด้วยวิธีนี้ถือว่าการอินเทอร์รัปต์อย่างหนึ่งได้ แต่จะมีลักษณะต่างออกไปจากการอินเทอร์รัปต์ของสัญญาณนี้ได้ ซึ่งมีศัพท์เฉพาะเรียกว่า NON-MASKABLE INTERRUPT นอกจากนี้การดำเนินการของโปรแกรมก็แตกต่างออกไปด้วย โดยจะ ไม่มีการเก็บค่าของคำสั่งที่กำลังจะไปทำในลำดับต่อไปภายในรีจิสเตอร์ PC เมื่อมีการรีเซตเกิดขึ้น โปรแกรมจะถูกสั่งให้กระโดดไปยังแอดเดรส 0000 ทันทันที ซึ่งตำแหน่งนี้จะเป็นตำแหน่งเริ่มต้นของการทำงานของไมโครคอนโทรลเลอร์ MCS-51 เมื่อเริ่มจ่ายไฟให้กับระบบเมื่อใดก็ตามที่มีการรีเซตเกิดขึ้นค่าสถานะต่างๆ ภายในไมโครคอนโทรลเลอร์จะถูกกำหนดกลับไปเป็นค่าเริ่มต้นใหม่อีกครั้ง

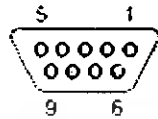
## 2.2 การใช้งานไมโครคอนโทรลเลอร์กับพอร์ตอนุกรม

การติดต่อกับพอร์ตอนุกรมนั้นจะยากกว่าการ ติดต่อกับกับพอร์ตขนาน การสื่อสารของ อุปกรณ์ที่ต่อกับพอร์ตอนุกรมจะถูกเปลี่ยน (Convert) เป็นสัญญาณแบบขนาน แล้วจึงนำไปประมวลผลต่อ ซึ่งจะใช้ Universal Asynchronous Receive / Transmitter (UART) เป็นตัวทำหน้าที่ ส่วนทางด้าน โปรแกรม ก็มีรีจิสเตอร์ที่ต้องจัดการมากกว่า Standard Parallel Port (SPP) อีกหลายตัว ข้อดีของพอร์ตอนุกรม

1. ระยะของสายอนุกรมสามารถมีความยาว ได้มากกว่าสายของขนานมาก ทั้งนี้เพราะสัญญาณของพอร์ตอนุกรม ซึ่งส่วนใหญ่ใช้มาตรฐาน RS-232C จะมีค่า -3 Volt ถึง -15 Volt สำหรับ Logic "1" หรือ "Mark" และมีค่า +3 Volt ถึง +15 Volt สำหรับ Logic "0" หรือ "Space" (สำหรับช่วง +3 Volt ถึงถึง -3 Volt เป็นช่วง Undefined) ส่วนสัญญาณของสายขนานนั้น Logic "1" จะมีค่า +5 Volt และ logic "0" จะมีค่า 0 Volt ทำให้สัญญาณของสายอนุกรมสามารถรับการสูญเสียของสาย (Cable loss) ได้มากกว่าสัญญาณของสายขนานปกติสายขนานจะไปได้เพียง 5 ฟุต ส่วนสาย RS-232 จะไปได้ถึง 50 ฟุตที่ความเร็วสูงสุดของมัน
2. สายอนุกรมจะใช้จำนวนสายไฟน้อยกว่าสายขนานถ้าต่อในลักษณะ Null Modemจะใช้สายเพียง 3 เส้น ขณะที่แบบขนาน จะต้องใช้สาย 19 ถึง 25 เส้น
3. การสื่อสารแบบไร้สายเช่นการใช้ Infra Red การส่งพร้อมกันทีละ 8 บิต แบบขนาน จะทำให้ไม่สามารถแยกแยะได้ว่า Bit ไหนเป็น Bit0 หรือ Bit1 ... เป็นต้น ปัจจุบันอุปกรณ์ IrDA มีความเร็วไม่ต่ำกว่า 115.2K Baud แต่มี Pulse length เพียง 3/16 ของ RS-232 เพื่อประหยัดพลังงาน เพราะส่วนมากใช้ในอุปกรณ์แบบพกพาเช่น Laptop หรือ Palmtop
4. ปัจจุบันไมโครคอนโทรลเลอร์ มักมีการผนวกพอร์ตการสื่อสารแบบอนุกรมไว้ด้วย เพราะใช้จำนวนขาน้อยกว่าแบบขนาน RS-232C กำหนด Baud rate ไว้ไม่เกิน 20K Baud ปัจจุบันได้แก้ไขให้รองรับกับเทคโนโลยีใหม่ได้ จึงมีการปรับปรุงถึง RS-232E ซึ่งมีรายละเอียดอีกหลายอย่าง



(Male at the computer side)



(Female at the cable side)

Pin	Name	RS232	V.24	Dir	Description
1	CD	CF	109	←	Carrier Detect
2	RXD	BB	104	←	Receive Data
3	TXD	BA	103	→	Transmit Data
4	DTR	CD	108.2	→	Data Terminal Ready
5	GND	AB	102	—	Signal Ground
6	DSR	CC	107	←	Data Set Ready
7	RTS	CA	105	→	Request to Send
8	CTS	CB	106	←	Clear to Send
9	RI	CE	125	←	Ring Indicator

รูปที่ 2.9 ตารางแสดงถึงหน้าที่ของขาต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง**

Abbreviation	Full Name	Originator	Function
TD	Transmit Data	DTE	Serial data output (TXD) from DTE.
RD	Receive Data	DCE	Serial data input (RXD) to DTE.
CTS	Clear To Send	DCE	Tell DTE that DCE is ready to exchange data.
(D)CD	(Data) Carrier Detect	DCE	Carrier from remote DCE is detected.
DSR	Data Set Ready	DCE	Tell DTE that DCE is ready to establish a link.
DTR	Data Terminal Ready	DTE	Tell DCE that DTE is ready to establish a link.
RTS	Ready To Send	DTE	Tell DCE that DTE is ready to exchange data.
RI	Ring Indicator	DCE	Ringling signal from the phone line is detected.

**รูปที่ 2.10 แสดงสัญญาณต่างๆที่ส่งในรูปแบบอนกร:**

**72151**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.2.1 หน้าที่ของสัญญาณต่าง ๆ มีดังนี้

**Protective ground** เป็นจุดที่ต่อกับตัวเปลือกของอุปกรณ์ และไม่ต่อกับสัญญาณใด ๆ ในระบบเพื่อใช้ต่อลงดิน เป็นการป้องกันอันตรายจากไฟลัดวงจร และใช้เป็น Shield ป้องกันการรบกวน

**Signal ground หรือ Common return** เป็นจุดสำคัญที่สุดที่ต้องมีในระบบ RS-232 เพราะเป็นจุดอ้างอิงของทุกสัญญาณ ยกเว้น Protective ground

**Transmit data** เป็นสัญญาณข้อมูลที่ส่งจาก DTE ไปยัง DCE ในขณะที่ไม่ได้ส่งข้อมูลมันจะมีสถานะเป็น Logic "1" หรือ "Mark" หรือ "Off" หรือ -5 V ถึง -15 V ที่ด้านส่ง (DTE) หรือ -3 V ถึง -15 V ที่ด้านรับ (DCE) การส่งข้อมูลจะเกิดขึ้นได้ต้องมีสัญญาณควบคุมที่เกี่ยวข้อง "On" ก่อนคือสัญญาณ RTS, CTS, (D)CD, DTR และ DSR

**Receive data** เป็นสัญญาณข้อมูลจาก DCE มายัง DTE ในขณะที่ไม่ได้ส่งข้อมูลมันจะมีสถานะเป็น Logic "1" หรือ "Mark" หรือ "Off" หรือ -5 V ถึง -15 V ที่ด้านส่งหรือ -3 V ถึง -15 V ที่ด้านรับ ในขณะที่ไม่ได้ส่งข้อมูลมันจะมีสถานะเป็น Logic "1" หรือ "Mark" หรือ "Off" หรือ -5 V ถึง -15 V ที่ด้านส่ง (DCE) หรือ -3 V ถึง -15 V ที่ด้านรับ (DTE) กรณีที่เป็นการสื่อสารแบบ Half-duplex สัญญาณ RD จะอยู่สถานะ "Off" ขณะที่สัญญาณ RTS อยู่ในสถานะ "On" และสัญญาณ RD จะยังคงอยู่ในสถานะ "Off" อีกชั่วระยะเวลาหนึ่งหลังจากที่สัญญาณ RTS เปลี่ยนจากสถานะ "On" มาเป็น "Off" แล้วเพื่อให้การรับ-ส่งข้อมูลเสร็จสมบูรณ์

**Request to send** เป็นสัญญาณจาก DTE ส่งไปให้ DCE เพื่อขอส่งข้อมูล ไปปรกติจะอยู่ในสถานะ "Off" เมื่อต้องการส่งข้อมูลจะเปลี่ยนเป็นสถานะ "On" จนกว่าการส่งเสร็จสิ้นจึงเปลี่ยนกลับไปที่สถานะ "Off" ตามเดิม ทั้งนี้ทาง DTE ต้องได้รับสัญญาณ CTS จึงจะสามารถส่งข้อมูล TD ไปยัง DCE ได้ และสัญญาณ RTS ที่กลับสู่สถานะ "Off" จะไม่สามารถเปลี่ยนเป็น "On" ใหม่ ขณะที่สัญญาณ CTS อยู่ในสถานะ "On" ต้องรอจนกว่าสัญญาณ CTS เปลี่ยนมาอยู่ในสถานะ "Off" ก่อน เพื่อป้องกันการเกิด Overrun

**Clear to send** เป็นสัญญาณที่ DCE ส่งให้ DTE เพื่อแจ้งว่าพร้อมรับการส่งข้อมูลจาก DTE

**Data set ready** เป็นสัญญาณที่ DCE ส่งให้ DTE เพื่อแจ้งว่า DCE สามารถเชื่อมต่อ ไปยังปลายทาง และพร้อมที่จะติดต่อกแล้ว

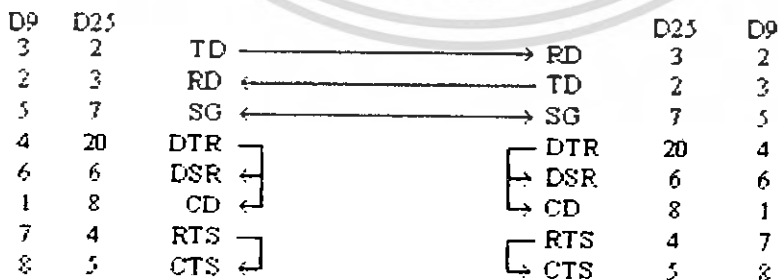
**Data Terminal ready** เป็นสัญญาณที่ DTE ส่งให้ DCE เพื่อแจ้งว่า DTE พร้อมหรือต้องการจะติดต่อสื่อสาร ซึ่งสัญญาณ DTR นี้ต้องเกิดก่อนทาง DCE จึงจะทำการติดต่อ ไปยังปลายทางและเมื่อติดต่อได้แล้วจึงส่งสัญญาณ DSR มายัง DTE เพื่อแจ้งให้รู้ว่าพร้อมรับการสื่อสารแล้ว และถ้า DTR เปลี่ยนเป็น Off แปลว่า DTE ไม่ต้องการติดต่อสื่อสารแล้วทาง DCE ก็จะปิดช่องสื่อสารและเปลี่ยนสัญญาณ DSR เป็น Off ทันที คู่สัญญาณระหว่าง RTS กับ CTS เป็นเรื่องของความพร้อมเกี่ยวกับช่องสื่อสารระหว่าง DTE กับ DCE ส่วนคู่สัญญาณ DTR กับ DSR เป็นเรื่องของความพร้อมเกี่ยวกับตัวอุปกรณ์ DTE กับ DCE

**Data carrier detect** เป็นสัญญาณที่ DCE ส่งให้ DTE เพื่อแจ้งว่าได้รับสัญญาณพาหะจาก DCE ที่อยู่อีกด้านหนึ่งของการสื่อสาร ซึ่งหมายความว่า ช่องการสื่อสารระหว่าง DCE ทั้ง 2 ไม่ขาดตอน พร้อมที่จะทำการสื่อสารได้ ซึ่งอุปกรณ์ DTE หรือ โปรแกรมที่ควบคุมการสื่อสารมักจะตรวจ สัญญาณนี้ ถ้าไม่อยู่ที่ on แสดงว่าช่องการสื่อสารขาด ก็จะไม่ทำการรับหรือส่งข้อมูล

**Ring Indicator** เป็นสัญญาณจาก DCE แจ้งให้ DTE รู้ว่ามีการเรียกจาก DCE ที่อยู่อีกด้านหนึ่งของการสื่อสาร ซึ่งมักจะใช้ในระบบ automatic answering

### 2.2.2 Null Modems

Null Modem ใช้สำหรับเชื่อมต่อระหว่าง DTE 2 ตัวเข้าด้วยกันโดยตรง ซึ่งโดยมากใช้ในการถ่ายข้อมูลระหว่างคอมพิวเตอร์ หรือใช้ในการพัฒนาระบบไมโคร โปรเซสเซอร์



รูปที่ 2.11 แสดงการเชื่อมต่อของสายสัญญาณ

### 2.2.3 Flow Control

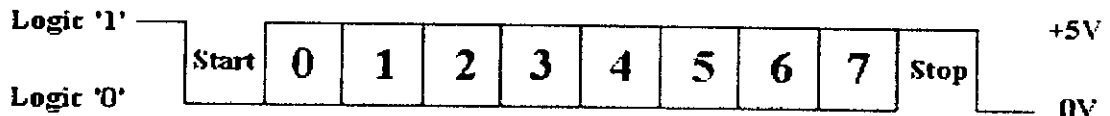
การติดต่อระหว่าง DTE และ DCE จะมีการควบคุมการไหลของข้อมูล เพื่อไม่ให้เกิดกา over flow ขึ้นได้ ซึ่งมีอยู่ 2 แบบคือ Hardware flow control และ Software flow control

สำหรับ Software flow control มักจะเรียกว่า Xon/Xoff flow control ซึ่งใช้รหัส ASCII 17 เป็นสัญญาณ Xon และใช้รหัส ASCII 19 เป็นสัญญาณ Xoff หลักการทำงานก็ง่าย ๆ คือ Modem จะมี Buffer อยู่เมื่อ Modem รับข้อมูลจาก Computer จน Buffer ใกล้เคียงเต็มมันก็จะส่งสัญญาณ Xoff ไปให้ Computer เพื่อให้ Computer หยุดส่งข้อมูลให้มันชั่วคราว และเมื่อ Buffer มีที่ว่างถึงระดับหนึ่ง Modem ก็จะส่งสัญญาณ Xon ไปให้ Computer เพื่อให้ Computer ส่งข้อมูลให้มันต่อ การควบคุมโดยวิธีนี้ประหยัดสายสัญญาณ เพราะรับ-ส่งผ่าน TD และ RD แต่อาจทำให้การสื่อสารช้าลงอย่างเห็นได้ชัดในกรณีที่ใช้กับการสื่อสารที่มี speed ต่ำ เพราะแต่ละตัวอักษร ASCII ที่รับ-ส่ง จะมีขนาด 10 Bit

ส่วน Hardware flow control มักจะเรียกว่า RTS/CTS flow control จะใช้สายสัญญาณของ Serial Port ในการควบคุม ทำให้ไม่บั่นทอนความเร็วของข้อมูล หลักการทำงานคือ เมื่อ Modem มีที่ว่างเพื่อรับข้อมูล มันก็จะส่งสัญญาณ CTS ไปให้ Computer และเมื่อมันใกล้จะเต็ม มันก็จะหยุดส่งสัญญาณ CTS ไปให้ Computer

### 2.2.4 รูปคลื่น สัญญาณ RS-232

การสื่อสาร โดย RS-232 เป็นการสื่อสารแบบ asynchronous หมายความว่าสัญญาณ clock ที่ใช้ควบคุมจังหวะไม่ได้ส่งไปพร้อมกับ Data แต่จะใช้ start bit เป็นตัว sync. ในแต่ละ word ของการสื่อสารและใช้สัญญาณ clock ภายในของแต่ละด้านเป็นตัวให้จังหวะเอง



รูปที่ 2.12 รูปคลื่นของสัญญาณที่ส่ง

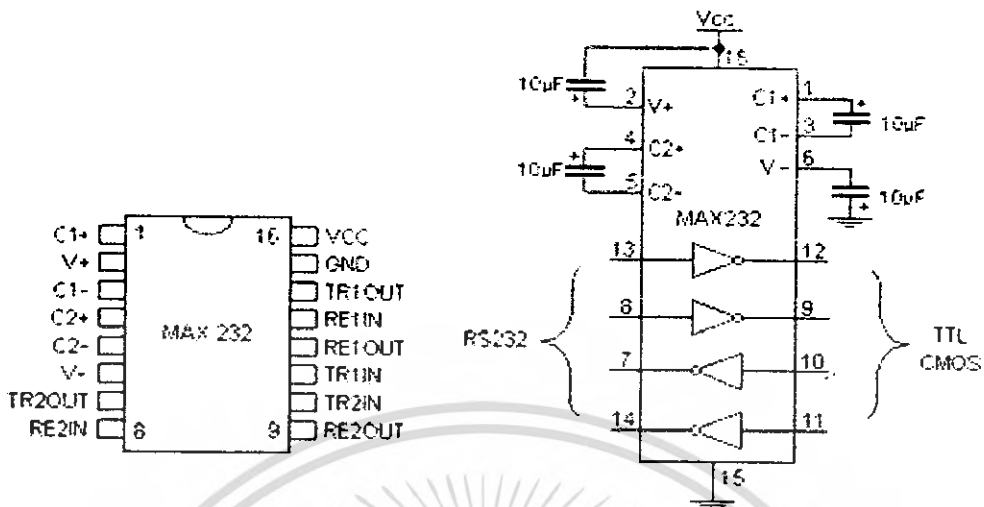
แสดงลักษณะของสัญญาณจาก UART เมื่อใช้ format แบบ 8N1 คือ 8 data bits ไม่มี parity bit และมี 1 stop bit ขณะที่ idle จะอยู่ในสถานะ “Mark” หรือ logic “1” การส่งจะเริ่มจากการส่ง start bit คือ logic “0” และตามด้วย LSB bit จนหมด data bits และถ้ามี parity bit ก็จะส่งที่จุดนี้แล้วลงท้ายด้วย stop bit ซึ่งมีค่าเป็น logic “1” ในรูปได้แสดง bit ที่ต่อถัดจาก stop bit ซึ่งมีค่าเป็น logic “0” หมายความว่า เป็น start bit ของ การส่ง word ถัดไป แต่ถ้ายังไม่มีการส่ง word ถัดไปก็ต้องอยู่ในสถานะของ logic “1” ซึ่งเป็นสถานะของ idle และถ้าสายอยู่ในสถานะของ logic “0” นานกว่าเวลาของการส่ง 1 full word ระบบจะถือว่าเป็นสัญญาณ “Break” เพื่อหยุดการสื่อสาร ดังนั้นต้องไม่ลืมที่จะส่งในสายกลับสู่สถานะ idle เมื่อสิ้นสุดการส่ง การรับ-ส่งข้อมูลในลักษณะนี้เรียกว่าแบบ frame คือมีกรอบปิดล้อมข้อมูลไว้ด้วย start bit และ stop bit

### 2.2.5 ตัวแปลงสัญญาณ RS-232

สัญญาณ RS-232 มีค่าแรงไฟต่างจากที่ใช้ใน UART ดังนั้นจึงต้องมีตัวแปลงสัญญาณเพื่อแปลงระดับสัญญาณให้เหมาะสมก่อนที่ จะเชื่อมต่อกับพอร์ตอนุกรม หรือ RS-232 port ของคอมพิวเตอร์

สัญญาณ RS-232 นั้น logic “0” จะมีค่า +3 V ถึง +25 V และ logic “1” จะมีค่า -3 V ถึง -25 V ส่วนค่าระหว่าง -3 V ถึง +3 V เป็นค่า undefined ระดับสัญญาณนี้ใช้กับทุกสัญญาณไม่ว่าจะเฉพาะสัญญาณรับ-ส่งข้อมูลเท่านั้นแต่ยังรวมถึงสัญญาณควบคุมต่าง ๆ เช่น DTR, RTS, CTS เป็นต้น

IC ที่ใช้มักจะเป็นเบอร์ 1488 (RS-232 Driver) และ 1489 (RS-232 Receiver) โดยภายในแต่ละตัวจะประกอบด้วย inverter 4 ตัวและต้องการไฟเลี้ยง 2 ชุดคือ +7.5 V ถึง +15 V และ -7.5 V ถึง -15 V ซึ่งอาจจะมีปัญหาในเครื่องที่มีไฟเลี้ยง +5 V เพียงชุดเดียว แต่ก็ยังมี IC อีกตัวหนึ่งคือ เบอร์ MAX-232 ซึ่งมีวงจร charge pump สามารถสร้างไฟ +10 V และ -10 V จากไฟ +5 V ได้ พร้อมทั้งมี 2 Tx และ 2 Rx อยู่ใน package เดียวกัน และรองรับ baud rate ได้ถึง 120 Kbps จึงสะดวกมากเพราะใช้ IC เพียงตัวเดียว รูปข้างล่างคือ MAX-232



รูปที่ 2.13 แสดงโครงสร้างภายในและตำแหน่งขาต่างๆของ Max232

ส่วนการที่เราจะนำข้อมูลมาใช้งานก็ต้องแปลงเป็น Parallel ก่อนซึ่งเป็นหน้าที่ของ UART ซึ่งปัจจุบันไมโครคอนโทรลเลอร์มักจะมี serial communication interface (SCI) อยู่ในตัว แต่อาจจะมีงานบางอย่างที่ไม่ได้ใช้ไมโครคอนโทรลเลอร์และต้องการประมวลผลข้อมูลกับทอร์คอนุกรม เช่น คอ ADC เข้ากับ UART หรือคอ LCD display เข้ากับ Serial comm. ก็ต้องใช้ UART ช่วย เช่น เบอร์ 8250 หรือ 16550A หรือเบอร์อื่น ๆ ที่ได้กล่าวมาแล้ว แต่มี UART อีกพวกหนึ่งที่แยก Tx bus กับ Rx bus ออกจากกันทำให้มีความคล่องตัวมากขึ้น

### 2.3 การสื่อสารข้อมูล

การสื่อสารข้อมูลมีจุดมุ่งหมายในการส่ง หรือการถ่ายทอข่าวสารจากจุดหนึ่งไปยังอีกจุดหนึ่งอย่างถูกต้องเหมือนกับคนที่ด้านส่งออกมา ในการสื่อสารรูปแบบใดๆ จะต้องประกอบไปด้วยส่วน ประกอบหลักเบื้องต้น 3 ส่วนดังนี้

1. แหล่งต้นกำเนิดข่าวสาร(Source)
2. ตัวกลางในการสื่อสาร(Media)
3. แหล่งรับข่าวสาร(Receiver)

### 2.3.1 ประเภทของการสื่อสารข้อมูล

ข้อมูลในระบบของการสื่อสารเขียนแสดงได้ด้วยค่าในระบบเลขฐานสอง โดยใช้ค่าตัวเลข 0 หรือ 1 มาประกอบกันเป็นรหัส แต่ในการส่งเราอาศัยการส่งทางไฟฟ้า ข้อมูลจะถูกเปลี่ยนให้อยู่ในรูปแบบทางไฟฟ้าโดยใช้ค่าสัญญาณไฟฟ้า 2 ระดับคือสูงและต่ำ ในการวัดอัตราเร็วในการส่งได้จากจำนวนบิตที่ส่งไปในหน่วยเวลาโดยทั่วไปใช้หน่วย Bit per second (bps) ซึ่งในระบบการสื่อสาร ข้อมูลนั้นอาจจะมีขนาดใดก็ได้แต่ในการส่งนั้นจะต้องมีการกำหนดจุดเริ่มต้น และจุดสิ้นสุดของการแต่ละส่ง ตามปกติจะจัดแบ่งข่าวสารที่จะส่งเป็นบล็อกๆ ซึ่งหนึ่งบล็อกคือกลุ่มของบิตจำนวนหนึ่งที่ถูกส่งออกไปเป็นหน่วยเดียวกัน โดยมีการนำกลุ่มบิตนั้นไปผ่านกระบวนการบางอย่าง เพื่อใช้ในการควบคุมข้อผิดพลาดที่อาจจะเกิดขึ้น

เราสามารถจำแนกวิธีการส่งข้อมูลได้หลายแบบตามคุณสมบัติต่างๆดังนี้

#### 2.3.1.1 การจำแนกตามทิศทางการส่งข้อมูล

สามารถแบ่งการส่งข้อมูลออกได้เป็น 3 ชนิด ดังนี้

##### 2.3.1.1.1 การรับส่งข้อมูลทางเดียว (Simplex)

เป็นการสื่อสารทางเดียว ที่เห็นได้ชัดก็คือ การรับส่งโทรทัศน์ และวิทยุกระจายเสียงนั่นเอง สถานีโทรทัศน์จะเป็นตัวส่งและเครื่องรับทำหน้าที่รับเพียงอย่างเดียว จะส่งข่าวสารมายังสถานีส่งไม่ได้

### รูปที่ 2.14 แสดงการรับส่งข้อมูลแบบทางเดียว (Simplex)

### 2.3.1.1.2 การรับส่งแบบผลัดกันส่ง (Half Duplex)

มีคุณสมบัติสามารถรับและส่งข้อมูลได้ แต่จะต้องสลับการส่งโดยจะส่งพร้อมกันทั้งสองด้านไม่ได้ อุปกรณ์ที่ใช้ในการติดต่อในแบบนี้ ได้แก่ วิทยุสื่อสาร และอินเทอร์เน็ตคอม



รูปที่ 2.15 แสดงการรับส่งข้อมูลสวนทางกันได้แบบผลัดการส่ง(Half Duplex)

### 2.3.1.1.3 การรับส่งสวนทางได้พร้อมกัน (Full Duplex)

การรับส่งแบบนี้ผู้รับและผู้ส่งสามารถรับและส่งพร้อมๆกันได้ ในเวลาเดียวกันโดยไม่จำเป็นต้องรอให้อีกฝ่ายหนึ่งส่งจบเสียก่อน เช่น การพูดโทรศัพท์ของเรา

**รูปที่ 2.16 แสดงการรับส่งข้อมูลแบบสวนทางกัน ได้พร้อมกัน (Full Duplex)**

**2.3.1.2 การจำแนกตามลักษณะการจัดข้อมูล**

สามารถแบ่งการส่งข้อมูลออกได้เป็น 2 ชนิดดังนี้

**2.3.1.2.1 การส่งข้อมูลแบบขนาน (Parallel Transmission)**

เป็นการส่งข้อมูลที่ละ ไบท์ คือส่งทุกบิตของรหัสที่ประกอบขึ้นเป็นอักขระไปพร้อมๆ กันในเวลาเดียวกัน มีข้อดีคือ ความสามารถในการส่งข้อมูลที่สูงขึ้น แต่จำนวนช่องทางการสื่อสารที่ จะต้องมีจำนวนเท่ากับจำนวนที่บิตที่ประกอบขึ้นเป็นอักขระ ซึ่งต้อง ใช้การมัลติเพล็กซ์ข้อมูลต่างๆ

**รูปที่ 2.17 แสดงการส่งข้อมูลแบบขนาน**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.3.1.2.2 การส่งข้อมูลแบบอนุกรม (Series Transmission)

การส่งข้อมูลแบบนี้จะกระทำทีละบิตด้วยแขนแนลเพียงแขนแนลเดียว ทางด้านรับจะมี อุปกรณ์สำหรับจัดการจัดข้อมูลดังกล่าวเป็นชุดอักขระ ตามข้อตกลงระหว่างอุปกรณ์ปลายทางทั้งสอง ที่สื่อสารกัน

#### รูปที่ 2.18 แสดงการส่งข้อมูลแบบอนุกรม

### 2.3.1.3 การจำแนกตามความสัมพันธ์ของข้อมูล

ในการสื่อสารข้อมูลนั้นจะต้องพิจารณาถึงความสัมพันธ์ระหว่างข้อมูล 2 ชนิด คือ

- ความสัมพันธ์ระหว่างบิต (Bit Synchronization)
- ความสัมพันธ์ของอักขระ (Character Synchronization)

ในการสื่อสารข้อมูลเราจำแนกวิธีการส่งข้อมูลตามการกำหนดความสัมพันธ์ของข้อมูลได้ 2 แบบ คือ

#### 2.3.1.3.1 การส่งแบบสัมพันธ์ (Synchronous Transmission)

การส่งแบบนี้ใช้สำหรับการส่งข้อมูลทั้งหมดไปในครั้งเดียวดังรูป 2.19 เทคนิคการส่งแบบนี้ ช่วงเวลาระหว่างบิตต่อบิตจะมีค่าเท่ากัน การส่งมีลักษณะคล้ายกับการส่งข่าวสารในรูปของ เลขฐานสองที่มีจำนวนติดต่อกันไป โดยไม่ได้แยกว่าความยาวใดเป็นช่วงอักขระใดในระบบเช่นนี้ บิตแต่ละบิตจะมีความยาวเท่ากัน ตัวอักษรแต่ละตัวมีช่วงเวลาห่างกันเท่ากับศูนย์ ทางด้านรับนั้น

เพียงหาว่าบิตแรกของตัวอักษรตัวแรกคือบิตใด และทราบขนาดหรือจำนวนบิตในหนึ่งตัวอักษร พร้อมทั้งความเร็วในการส่งก็สามารถแยกข่าวสารของแต่ละอักษรออกมาได้

### รูปที่ 2.19 แสดงการส่งข้อมูลแบบสัมพันธ์

#### 2.3.1.3.2 การส่งแบบไม่สัมพันธ์ (Asynchronous Transmission)

การส่งแบบนี้ตัวอักษรจะถูกส่งออกไปที่เวลาใดๆ ก็ได้โดยไม่ต้องมีความสัมพันธ์ระหว่างตัวอักษรว่าต้องมีเวลาที่แน่นอนอย่างไร ทางด้านรับจะต้องทราบถึงบิตเริ่มต้นของรหัสแต่ละตัวว่าเริ่มต้นเมื่อใดซึ่งเมื่อใดซึ่งสามารถกระทำได้โดยการเพิ่มบิตที่เรียกว่า พัลส์เริ่มต้น (Start Pulse) โดยเติมเข้าไปที่ข้างหน้าชุดของทุกอักษร และเมื่อการส่งส่งครบทุกบิตของตัวอักษรแล้วจะต้อง มีบิตสำหรับบอกถึงการสิ้นสุดที่เรียกว่า พัลส์การสิ้นสุด (Stop Pulse) ส่งมาให้ทางด้านรับมีเวลาสำหรับ การเตรียมข้อมูลของตัวอักษรตัวต่อไปดังรูป 2.20 บางครั้งจึงเรียกระบบการส่งแบบนี้ว่า ระบบการส่งแบบเริ่มหยุด (Start-stop Transmission)

### รูปที่ 2.20 แสดงการส่งข้อมูลแบบไม่สัมพันธ์

## 2.4 ดีซีมอเตอร์ (DC Motor)

### 2.4.1 หลักการทำงานของดีซีมอเตอร์

ดีซีมอเตอร์เป็นทรานสดิวเซอร์แรงบิดซึ่งมีการออกแบบให้มีคุณลักษณะพิเศษคือแรงบิดของเพลลาของมอเตอร์จะเป็นสัดส่วนโดยตรงกับกระแสอาร์เมเจอร์ ความสัมพันธ์ระหว่างแรงบิดของเพลลาและกระแสคือ

$$T = K \Phi I$$

เมื่อ  $T$  คือแรงบิดของเพลลา มีหน่วยเป็น นิวตัน-เมตร

$\Phi$  คือเส้นแรงแม่เหล็ก มีหน่วยเป็น เวเบอร์

$I$  คือกระแส มีหน่วยเป็น แอมแปร์

$K$  คือค่าคงตัว

จะเห็นได้ว่าแรงบิดของเพลลาจะเป็นสัดส่วน โดยตรงกับผลคูณของเส้นแรงแม่เหล็กและกระแส การที่ขดลวดตัวนำเคลื่อนที่ในสนามแม่เหล็กจะทำให้เกิดแรงดันตกคร่อมตัวมันเองโดยแรงดันนี้จะเป็นสัดส่วนกับความเร็วของเพลลาของมอเตอร์และเส้นแรงแม่เหล็ก โดยมีความสัมพันธ์ดังนี้

$$E = K \Phi \omega$$

เมื่อ  $E$  คือแรงดันย้อนกลับ มีหน่วยเป็น โวลท์

$\Phi$  คือเส้นแรงแม่เหล็ก มีหน่วยเป็น เวเบอร์

$\omega$  คือความเร็วของมอเตอร์ มีหน่วยเป็น เรเดียน/วินาที

$K$  คือค่าคงตัว

### 2.4.2 ดีซีมอเตอร์แบบเส้นแรงแม่เหล็กคงที่

ฟิลด์ (Field) ของมอเตอร์ชนิดนี้นิยมใช้แม่เหล็กถาวร มอเตอร์ชนิดนี้จึงอาจจะเรียกว่ามอเตอร์แบบแม่เหล็กถาวร (Permanent-Magnet Motors) มอเตอร์แบบแม่เหล็กถาวรนี้ไม่ต้องการกระตุ้นเพื่อให้เกิดสนามแม่เหล็ก เนื่องจากมีแกนขั้ว (Core) ทำด้วยอัลลอยซึ่งสามารถรักษาคุณสมบัติความเป็นแม่เหล็กหรือขดลวดฟิลด์ มอเตอร์ชนิดนี้จึงมีประสิทธิภาพสูงกว่ามอเตอร์ชนิดขั้วหรือมอเตอร์คอมปาวด์ที่มีพิกัดเท่ากันนอกจากนั้นยังควบคุมความเร็วและทิศทางการหมุนได้ง่ายโดยการกำหนด ขนาดและขั้วแรงดันที่ป้อนให้แก่มอเตอร์ ซึ่งหมายความว่าถ้ากลับขั้ว

ของแรงดันจะทำให้ทิศทางของกระแสที่อาร์เมเจอร์เปลี่ยนแปลงไปเพียงอย่างเดียวเท่านั้น แต่ไม่ทำให้ฟลักซ์เปลี่ยนแปลงจึงได้อัตรา ส่วนระหว่างกระแสอาร์เมเจอร์และแรงบิดมีค่าคงที่ ดังมีความสำคัญดังนี้

$$T=KI\omega$$

$$E=Ke\omega$$

ระบบนี้จะให้ความสัมพันธ์ระหว่างกระแสอาร์เมเจอร์, แรงบิด และความเร็วอยู่ในลักษณะเชิงเส้น สมการทางไฟฟ้าของคิซิมอเตอร์แบบนี้เขียนได้เป็น

$$V=Ke\omega+L(di/dt)+Ri$$

เมื่อ  $V$  คือแรงดันที่ป้อนให้มอเตอร์

$Ke$  คือค่าคงที่ของแรงดันย้อนกลับ

$L$  คือค่าความเหนี่ยวนำของขดอาร์เมเจอร์

$R$  คือความต้านทานที่ขั้วของมอเตอร์

#### 2.4.3 คิซิมอเตอร์แบบอาร์เมเจอร์เป็นแกนหลัก

โครงสร้างของโรเตอร์และสเตเตอร์ของมอเตอร์แบบแกนหลัก โครงสร้างของมอเตอร์แบบนี้โมเมนต์ของแรงเฉื่อยสูงที่สุด และมีค่าอินดักแตนซ์ของโรเตอร์สูงที่สุดด้วย ดังนั้นมอเตอร์นี้จึงมีปริมาณการจุกความร้อนได้สูง และสามารถจะทนโอเวอร์โหลดได้ในระยะเวลาที่ยาวนานโดยไม่ทำให้มอเตอร์เสียหาย

#### 2.5 เพาเวอร์มอสเฟต

เพาเวอร์มอสเฟตเป็นอุปกรณ์ประเภทอุปกรณ์อิเล็กทรอนิกส์กำลังชนิดหนึ่ง ซึ่งมีลักษณะโครงสร้างของเพาเวอร์มอสเฟตจะแบ่งเป็น N-channel และ P-channel จากโครงสร้างจะพบว่ามีเกตกับซอร์สจะถูกแยกออกจากกันด้วยชั้นของซิลิกอน ซึ่งเปรียบเสมือนมีความจุไฟฟ้าอยู่ เมื่อป้อนแรงดันที่เหมาะสม- สมเข้าที่ขั้วเกตจะทำให้ชั้นของซิลิกอนเกิดการเหนี่ยวนำจนเกิดมีกระแสไหล เพาเวอร์มอสเฟตนิยมใช้มากในวงจรสวิตซ์ซึ่งเพาเวอร์ซัพพลาย และการควบคุมมอเตอร์เนื่องจากมีข้อดีกว่าเพาเวอร์ทรานซิสเตอร์คือ

- กำลังสูงพิเศษขณะสวิตซ์ต่ำ
- มีสัญญาณรบกวนต่ำกว่าแบบไบโพลาร์
- มีคุณสมบัติการทำงานที่มั่นคงและเสถียรภาพทางอุณหภูมิ
- ใช้ในวงจรความถี่สูงได้ดี

### รูปที่ 2.21 โครงสร้างของเพาเวอร์มอสเฟต

#### 2.5.1 คุณลักษณะของเพาเวอร์มอสเฟต

แรงดันพังทลายเป็นระหว่างขาเดรนกับขาซอร์สดังรูป 2.23 โดยที่อุณหภูมิต่างๆ ค่าแรง-ดันพังทลายจะลดลงอย่างเห็นได้ชัด ขณะที่เพาเวอร์มอสเฟตนำกระแสอย่างสมบูรณ์แรงดันที่ตกคร่อมระหว่างขาเดรนกับขาซอร์สจะแปรผันตรงกับกระแสที่ไหลผ่านตัวมัน

#### รูปที่ 2.22 แรงดันพังทลายที่เดรนกับซอร์สเมื่ออุณหภูมิของรอกต่อเปลี่ยนไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.5.2 เงื่อนไขของวงจรขับเพาเวอร์มอสเฟต

การขับเพาเวอร์มอสเฟตให้นำกระแสที่แตกต่างจากการขับกระแสไบอัส ในไบโพลาร์ทรานซิสเตอร์ เนื่องจากมีเงื่อนไขการไบอัสที่ต่างกัน สำหรับไบโพลาร์ทรานซิสเตอร์กระแสไหลผ่านคอลเลกเตอร์และอีมิเตอร์ได้ก็ต่อเมื่อมีกระแสไหลไบอัสไหลผ่านที่เบสและอีมิเตอร์ แต่เพาเวอร์มอสเฟตจะมีกระแสไหลผ่านเดรนและซอร์ส ก็ต่อเมื่อแรงดันคกคร่อมขาเกตและซอร์สมีค่าอย่างต่ำเท่ากับค่าแรงดันขีดเริ่ม (Threshold Voltage) ของมัน



รูปที่ 2.23 แสดงตัวอย่างของ Gate Charge Chart

ลักษณะของกราฟแบ่งออกได้เป็น 3 ช่วง ตามผลของประจุที่เพิ่มขึ้น คือ

1. ช่วงเวลาหน่วงก่อนเริ่มนำกระแส (Time-on Delay Time)  $t=0$
2. ช่วงเวลาเริ่มนำกระแส (Rise Time)  $t=0-1$
3. ช่วงเวลาสะสมประจุส่วนเกิน (Excess Charge Time)  $t=2-3$

เมื่อเพาเวอร์มอสเฟตเริ่มชาร์จประจุที่ขาทก จนกระทั่งพ้นช่วงเวลาหน่วงก่อนเริ่มนำกระแส เมื่อแรงดันที่ขาทกมากกว่าแรงดันขีดเริ่มจึงจะเริ่มมีกระแสไหลผ่านเดรนและซอร์ส ที่เวลา  $t=0$  ค่าของแรงดันตกคร่อมเดรนกับซอร์ส  $V_{ds}$  จะไม่ลดลงจนกว่าจะถึงเวลา  $t=1$  แรงดันตกคร่อมเดรนกับซอร์ส  $V_{ds}$  จึงมีค่าลดลงอย่างรวดเร็ว เพาเวอร์มอสเฟตจะนำกระแสได้เต็มที่ในช่วงเวลาเท่ากับ  $t=1$  ถึง  $t=2$  นี้เอง และช่วงนี้แรงดันที่ขาทก  $V_{gs}$  จะคงที่

ในช่วงเวลาสะสมประจุส่วนเกินค่าความต้านทานระหว่างเดรนและซอร์ส จะลดลงเรื่อยๆ หากปล่อยให้มีการสะสมประจุต่อไปในช่วงเวลา  $t=2$  ถึง  $t=3$  ประจุที่สะสมเพิ่มขึ้นจะทำให้เกิดการหน่วงขณะเริ่มหยุดนำกระแส เนื่องจากเพาเวอร์มอสเฟตจะต้องใช้เวลามากในการคายประจุส่วนเกินนี้ทิ้งไป ดังนั้นการขับขาทกด้วยแรงดันที่สูงเกินความจำเป็น จะทำให้ช่วงเวลาเริ่มหยุดนำกระแสเพิ่มขึ้นด้วย

### 2.5.3 การกำหนดเวลาในการเปลี่ยนสถานะ

ปกติแล้วผู้ผลิตจะให้กราฟค่า  $V_{gs}$  ที่เพิ่มขึ้น ในขณะที่ค่าประจุสะสมที่ขาทกเพิ่มขึ้นหรือที่เรียกว่า Gate Charge Chart มาในคาต้าลิทด้วย กราฟนี้มีประโยชน์มากในการคำนวณค่ากระแสไบอัสเกตและเวลาในการเปลี่ยนสถานะของเพาเวอร์มอสเฟต

ก.

ข.

รูปที่ 2.24 (ก) แสดงการชาร์จประจุที่ขาทกตามเวลาที่มีผลต่อการนำกระแส

#### (ข) แสดงผลเมื่อเริ่มหยุดนำกระแสของเพาเวอร์มอสเฟต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การหุขนำกระแสของเพาเวอร์มอสเฟตจึงทำได้ง่าย ๆ ด้วยการคายประจุที่ขาทกทั้งไป เพื่ออัดประจุและคายประจุมีค่าเท่ากันช่วงเวลาเริ่มนำกระแสและช่วงเวลาหุขนำกระแส จะมีค่าเท่ากันด้วย ยกเว้นหากมีประจุสะสมมากในช่วงเวลา  $t=2$  ถึง  $t=3$  การเริ่มหุขนำ กระแสจะมีช่วงเวลาหน่วงเพิ่มขึ้น เนื่องจากต้องใช้เวลาในการคายประจุส่วนเกินทิ้งไป

จะเห็นได้ว่าการขับเพาเวอร์มอสเฟตให้นำกระแสได้นั้น จะต้องมีการอัดและคายประจุ ที่ขาทก วงจรขับเพาเวอร์มอสเฟตจะต้องมีคุณสมบัติของการจ่ายกระแสและนำกระแสได้ที่ประมาณ 200 ถึง 400 มิลลิแอมแปร์ด้วย และต้องให้แรงดันคั่นคร่อมที่ขาทก  $V_{gs}$  มีค่ามากพอ เพื่อที่เพาเวอร์มอสเฟตจะทำงานได้เต็มที่ (ประมาณ 10 โวลท์)

สิ่งสำคัญอีกประการคือ การเลือกใช้งานเพาเวอร์มอสเฟตเราควรเลือกเพาเวอร์มอสเฟต ที่มีพิคกระแสและแรงดันมากกว่ามอดอร์ประมาณ 1.5-2 เท่า

## 2.6 การเชื่อมต่อทางแสง

ตัวเชื่อมต่อทางแสง (Optocoupler) หรือตัวแยกโดยใช้แสง (Opto Isolater) เป็นอุปกรณ์ที่นำมาเชื่อมต่อสัญญาณต่างๆ ของวงจรที่มีกราวด์ต่างกัน สามารถป้องกันการรบกวนซึ่งกันและกันระหว่างภาคอินพุตกับภาคเอาต์พุตได้อย่างเด็ดขาด ออปโตคัปเปิ้ลเป็นอุปกรณ์ที่ประกอบด้วยแหล่งกำเนิดแสงและตัวตรวจจับแสงอยู่ภายในตัวเดียวกัน แหล่งกำเนิดแสงส่วนใหญ่มักจะใช้ไดโอดเปล่งแสงอินฟราเรดที่ทำมาจากแกเลียมอาร์เซไนด์ (GaAs) ส่วนตัวตรวจจับแสง (Photo Detector) สามารถเป็น ไปได้ทั้งโฟโตทรานซิสเตอร์, โฟโตไดโอด, SCR และ TRIAC ในออปโตคัปเปิ้ลสัญญาณอินพุตจะถูกเปลี่ยนเป็นสัญญาณแสงแล้ว จึงส่งสัญญาณแสงนี้ไปยังโฟโตดีเทคเตอร์ ออปโตคัปเปิ้ลถูกออกแบบมาเพื่อป้องกันอุปกรณ์อิเล็กทรอนิกส์ไม่ให้รับแรงกระชากสูงๆ ซึ่งเป็นต้นเหตุที่ทำให้เกิดสัญญาณเอาต์พุตที่ไม่ถูกต้อง

## 2.7 หลักการออกแบบส่วนประกอบและโครงสร้างต่างๆ

### 2.7.1 เพลา (Shaft)

เพลาอาจมีชื่อเรียกแตกต่างกันออกไป ตามลักษณะการใช้งานดังนี้

- เพลา (Shaft) เป็นชิ้นส่วนที่หมุนและใช้ในการส่งกำลัง

- แกน (Axle) เป็นชิ้นส่วนลักษณะเดียวกับเพลาแต่ไม่หมุน ส่วนมากเป็นตัวรองรับชิ้นส่วนที่หมุน เช่น ล้อ ล้อสาขพาน เป็นต้น อย่างไรก็ตามทั้งเพลาและแกนก็นิยมเรียกรวมกันว่าเพลา ไม่ว่าจะชิ้นส่วนจะหมุนหรือไม่หมุนก็ตาม

- สปินเดิล (Spindle) เป็นเพลาขนาดสั้นที่ไม่หมุน เช่น เพลาที่แทนกลึง (Head Shaft)

เป็นเพลาที่คิดเป็นชิ้นส่วนต่อเนื่องกับเครื่องชนิด มอเตอร์ หรือเครื่องต้นกำลังอื่นๆ มีขนาดรูปร่างและส่วนยื่นออกมาต่อกับเพลาอื่นๆ

- เพลาแนว (Line Shaft) หรือเพลาที่ส่งกำลัง (Power Transmission Shaft) หรือเพลาเมน (Main Shaft) เป็นเพลาที่ต่อตรงจากเครื่องต้นกำลังและใช้ในการส่งกำลังไปยังเครื่องจักรอื่นๆ โดยเฉพาะ

- แจ็คชาฟ (Jack Shaft) หรือเคาเตอร์ชาฟ (Counter Shaft) เป็นเพลาขนาดสั้นที่ต่อระหว่างเครื่องต้นกำลังกับเพลาเมนหรือเครื่องจักรกล

- เพลาอ่อน (Flexible Shaft) เป็นเพลาที่สามารถอ่อนตัวหรือโค้งงอได้ เพลาประเภทนี้ทำด้วยสายลวดใหญ่ (Cable) ลวดสปริงหรือลวดเกลียว (Wire Rope) ใช้ในการส่งกำลังในลักษณะที่แกนหมุนทำมุมกันได้แต่กำลังส่งได้น้อย

เพลาอาจจะรับแรงดึง แรงกด แรงบิด หรือแรงคั้น หรือหลายแรงรวมกันก็ได้ ดังนั้นการคำนวณก็ต้องใช้ความเค้นผสมเข้าช่วย แรงเหล่านี้ยังอาจจะมีการเปลี่ยนแปลงขนาดตลอดเวลา ทำให้เพลาเสียหายเพราะความล้าได้ ดังนั้นจะต้องออกแบบเพลาให้มีความแข็งแรงเพียงพอสำหรับการใช้งาน นอกจากนี้เพลายังมีความแข็งแกร่ง (Rigidity) เพียงพอ เพื่อลดมุมบิดภายในเพลาให้อยู่ในขีดจำกัดที่เหมาะสม ระยะโก่ง (Deflection) ของเพลาก็เป็นสิ่งสำคัญในการกำหนดขนาดเพลาเช่นเดีวกัน เพราะถ้าเพลา มีระยะโก่งมากก็จะเกิดการแกว่งขณะหมุนทำให้เกิดความเร็ววิกฤต (Critical Speed) ของเพลาตกลง ซึ่งอาจทำให้เพลาเกิดการสั่นอย่างรุนแรง ในขณะที่ความเร็วของเพลาเข้าใกล้ความเร็ววิกฤตนี้ได้ ระยะโก่งนี้ยังมีผลต่อการเลือกชนิดของเพลาและที่รองรับเพลา เช่น บอลเบริง (Ball Bearing) ก็ต้องมีการเอียงแนว (Misalignment) ในการใช้งานที่เหมาะสมกับเพลาด้วย

## การพิจารณาในการออกแบบเพลลา

การคำนวณหาขนาดของเพลลาที่เหมาะสม ขึ้นอยู่กับลักษณะการใช้งานในบางครั้ง การหาขนาดของเพลลา เพื่อให้เพลลาทนต่อแรงที่มากระทำอย่างเฉียดเท่านั้นไม่เพียงพอ เช่น ในกรณีของเพลลาถูกเบีซว (Cam Shaft) ในเครื่องชนิดสันคาบภายในต้องการให้มรดำแหน่งที่เที่ยงตรง ดังนั้นมุมบิคของเพลลาที่เกิดขึ้นในขณะที่ใช้งานจะต้องมีค่าไม่มากกว่าที่กำหนดไว้ เป็นต้น นั่นคือเพลลาจะต้องมีความแข็งแรงแรงอยู่ในพิภคที่ดองการ ถ้ามุมบิคมากไปนอกจากจะเสิชความเที่ยงตรงทางด้านตำแหน่งแล้ว ยังอาจจะก่อให้เกิดความสันสะเทือนซึ่งมีผลทำให้เฟืองและแบริงที่รองรับเพลลาอยู่เกิดความเสิชหาอได้ง่าอ

ความแข็งแรงแรงที่สำคัญอีกอย่างหนึ่งก็คือ ความแข็งแรงแรงทางด้านระยะโก่ง เพราะจะต้องใช้ระยะโก่งของเพลลาที่อยู่ใต้แรงภายนอกเป็นควสำคัญในการกำหนดระยะเบีซคควอ (Clearanc) ระหว่างล้อสาขพาน เฟือง โครงสร้างของเครื่องจักรตลอดจนการเลือกชนิดของแบริง สำหรับรองรับเพลลาที่หมุนให้เหมาะสม ถ้าเพลลามีระยะโก่งมากเกินไปจะทำให้ความยาวของฟันเฟืองส่วนที่สัมผัสกันหรือขบกันลดลง เป็นผลให้อัตราส่วนการขบ (Contact Ratio) ของเฟืองลดลงด้วย ทำให้ส่งกำลังของเฟืองไม่ราบเรียบเท่าที่ควร การเลือกแบริงชนิดที่อนุญาตให้มีการเอียงแนวได้พอเหมาะสมกับระยะโก่งของเพลลาที่จะเกิดขึ้น

จากเหตุผลดังกล่าวมาแล้วจะเห็นว่าขนาดของเพลลา อาจจะหามาได้โดยใช้ความแข็งแรงแรงที่ดองการ แทนที่จะเป็นความแข็งแรงแรงในด้านการรับแรงภายนอกก็ได้ การหาระยะโก่งของเพลลา อาจทำได้โดยใช้วิธีการกลศาสตร์วัสดุ

### 2.7.2 แบริง (Bearing)

เป็นตัวช่วยลดความเสิชคทาน ความสึกหรอและความร้อน ที่เกิดขึ้นในชิ้นส่วนของเครื่องจักรที่มีการเสิชคสกัน การหล่อลื่นทำได้โดยการใช้สารหล่อลื่น (Lubrication Oil) หรือจารบี (Grease) หรืออากาศและอื่นๆ ซึ่งขึ้นอยู่กับการใช้งานในการเลือกใช้แบริงนั้นจะต้องเลือกใช้แบบให้เหมาะสมกับเพลลาที่เราเลือกใช้ด้วย

### 2.7.3 เฟือง (Gear)

เฟือง เป็นชิ้นส่วนของเครื่องจักรกลอีกอย่างที่มีความสำคัญมากมีการใช้ในงานด้านวิศวกรรมอย่างกว้างขวาง เพราะสามารถส่งกำลังได้แม่นยำเที่ยงตรง ไม่มีการลื่นไหล ให้อัตรา

ทศที่แน่นอนเช่นเคียวกับการส่งกำลังด้วยสายพาน ในการเลือกใช้เฟืองจะต้องเลือกประเภทของเฟืองให้เหมาะสมกับโซ่หรือสายพาน และต้องคำนึงถึงรูปแบบของการส่งกำลังด้วย

#### 2.7.4 โซ่ (Chain)

การขับเคลื่อนโซ่มีการใช้งานกันมากในงานเครื่องจักรกล เนื่องจากมีลักษณะการขับเคลื่อนสายพาน โดยโซ่จะคล้องอยู่กับล้อโซ่หรือเฟืองโซ่ ซึ่งติดอยู่บนเพลาขับและเพลาตาม อัตราทดของการขับเคลื่อนขึ้นอยู่กับขนาดของเฟืองโซ่ทั้งสอง และการขับเคลื่อนนี้จะไม่มีการสลิปขึ้นระหว่างเฟืองกับโซ่

การขับเคลื่อนโซ่มีข้อดีคืออยู่ระหว่างการขับเคลื่อนสายพาน และการขับเคลื่อนเฟืองด้านราคาสมรรถนะในการส่งกำลังและการบำรุงรักษา โซ่สามารถขับได้ในระยะทางไกลกว่าการขับเคลื่อนสายพาน และการขับได้พร้อมกันหลายๆเพลา ซึ่งมีทิศทางการหมุนตามกันหรือสวนกันก็ได้ แต่การใช้โซ่จะมีข้อจำกัดในการส่งกำลังกล่าวคือเมื่อใช้งานที่มีความเร็วปานกลางจนถึงความเร็วสูงสุด จะทำให้เกิดการสึกหรอที่ข้อต่อ และความต้านทานแรงดึงของวัสดุชิ้นส่วนโซ่ นอกจากนี้แล้วถ้าใช้งานด้วยความเร็วสูงสุดจะต้องคำนึงถึงแรงที่เกิดขึ้นในโซ่ เนื่องจากแรงหนีศูนย์กลางในการขับเคลื่อนโดยโซ่หลายเส้นขนานกัน อาจทำให้แรงที่กระทำกับโซ่ไม่กระจายไปเท่ากันตลอดความกว้างของโซ่ และถ้าหล่อลื่นไม่ดีพอก็จะเป็นสาเหตุประการสำคัญที่ทำให้โซ่สึกหรอได้มาก

โซ่ที่คล้องอยู่บนเฟืองโซ่จะมีลักษณะคล้ายรูปหลายเหลี่ยมบนเฟืองโซ่ ทำให้แกนของแรงในแนวสัมผัสกับเฟืองโซ่เปลี่ยนแปลงตลอดเวลาที่ส่งกำลัง นอกจากนี้ในระหว่างเคลื่อนที่เข้าหาและเคลื่อนที่ออกจากเฟืองโซ่ ข้อต่อโซ่แต่ละข้อจะเกิดการหมุนจนทำให้เกิดแรงเสียดทานขึ้นเป็นผลทำให้สูญเสียกำลังและสึกหรอ ผลต่อมาก็คือระยะพีคของโซ่จะเพิ่มขึ้นจนอาจจะหลุดออกจากเฟืองโซ่ได้

#### 2.8 ดิจิตอลอิมเมจโปรเซสซิง (Digital Image Processing)

การประมวลผลภาพด้วยคอมพิวเตอร์นั้น โดยทั่วไปแล้วสามารถทำการแบ่งแยกเป็น 2 ประเภทใหญ่ๆด้วยกันคือ การประมวลผลภาพในระดับต่ำ (Low-level Image Processing) และการประมวลผลในระดับสูง (High-level Processing) ซึ่งสามารถอธิบายความแตกต่างของการประมวลผลทั้งสองระดับได้ดังนี้

การประมวลผลภาพในระดับต่ำนั้นเป็นการประมวลผลเชิงตัวเลขเกือบทั้งหมดเพื่อหาตัวแปรต่างๆ มาอธิบาย ข้อมูลภาพ และมีจุดประสงค์ที่จะนำตัวแปรเหล่านี้ไปใช้ในการประมวลผลในเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระดับสูงต่อไป การประมวลผลภาพในระดับต่ำโดยทั่วไปจะประกอบไปด้วยการกำจัดสัญญาณรบกวน, การทำให้ภาพคมชัดขึ้น, การหาขอบเขตของภาพ, การทำเส้นที่ภาพหรือการแบ่งแยกวัตถุภายในภาพ, การสร้างภาพไบนารี เป็นต้น

ในขณะที่ การประมวลผลในระดับสูงคือการทำให้คอมพิวเตอร์รู้จักและเข้าใจภาพได้ เช่น การจดจำรูปแบบของตัวเลขตัวอักษร, แยกแยะวัตถุที่มีรูปร่างแตกต่างไปจากเดิมที่ถูกกำหนดไว้, การจดจำวัตถุสีแดง เป็นต้น

ซึ่งในโครงการนี้ก็จัดได้ว่า เป็นการประมวลผลสัญญาณภาพแบบระดับต่ำ เพราะลักษณะของการประมวลผลที่จะทำการแยกแยะระหว่างพื้นฉากหลังกับวัตถุเท่านั้น และตรวจสอบเฉพาะวัตถุที่เป็นหุ่นยนต์และวัตถุที่ถูกสมมุติให้เป็นจุดหมายปลายทางเพื่อให้สามารถทำการแบ่งส่วนภาพหรือ(Image Segmentation) มาเป็นส่วนแรกดังนี้

### 2.8.1 การแบ่งส่วนของภาพ

การแบ่งส่วนของภาพ(Segmentation) นั้นเป็นการกำหนดขอบเขตของวัตถุภายในภาพ บริเวณรอบจุดใดๆที่แตกต่างกันให้สามารถเห็นความแตกต่างชัดขึ้นเพื่อนำไปประมวลผลขั้นสูงต่อไปได้ง่ายขึ้น ซึ่งสิ่งนี้เป็นส่วนที่จำเป็นต่อการวิเคราะห์ภาพว่าวัตถุในภาพนั้น มีขนาดใหญ่เท่าใด ส่วนไหนเป็นพื้นหลัง, แล้ววัตถุนั้นที่อยู่เท่าไร ดังนั้นการแบ่งส่วน (Segmentation) นั้นเป็นพื้นฐานที่จำเป็นสำหรับการชี้และการอธิบายวัตถุที่อยู่ในภาพ ให้สามารถถูกประมวลผลโดยโปรแกรมได้ดี และสะดวกยิ่งขึ้น ซึ่งแบ่งส่วน (Segmentation) นี้สามารถทำได้ง่าย ๆ โดยการนำข้อมูลภาพที่รับเข้ามานำไปเปรียบเทียบกับค่าคงที่ค่าหนึ่งซึ่งเรียกว่า ค่าเทรชโฮลด์(Threshold) และภาพที่ได้หลังจากการเปรียบเทียบจะถูกเรียกว่า ภาพไบนารี(Binary Image)

### 2.8.2 การทำเทรชโฮลด์ (Threshold Technique)

การทำเทรชโฮลด์ถือว่าเป็นเทคนิคที่สำคัญในการประมวลผลภาพในส่วนของการทำ เส้นที่ภาพ คือการแยกองค์ประกอบของภาพ ไปเป็นส่วนประกอบย่อยๆ ที่มีความสัมพันธ์ทางกายภาพของภาพนั้น และส่วนประกอบที่ถูกแยกออกมานั้นอาจถูกนำไปประมวลผลภาพในส่วนอื่นได้ต่อไป ซึ่งการทำเส้นที่ภาพจะมีหลักการทำในแนวเดียวกันกับสายตาของคน คือสามารถแยกลักษณะเด่นออกมาจากภาพที่มองเป็นได้และเทคนิคการทำเทรชโฮลด์ ซึ่งถือว่าเป็นเทคนิคในการแยกองค์ประกอบของภาพที่ง่ายเทคนิคหนึ่งมีหลักการว่า จุดภาพที่มีคุณสมบัติอยู่ในช่วงใดๆจะถูกจัดเป็นกลุ่มได้โดยที่ระดับความเข้ม นั้นสามารถที่จะแบ่งแยกกลุ่มภาพออกเป็น 2 กลุ่ม ได้อย่างชัดเจน คือกลุ่มของวัตถุ (Object) ซึ่งมีระดับความเข้มของภาพ  $g(x,y)$  ค่อนข้างต่ำ (มืด) กับกลุ่มของส่วนที่เป็นพื้นหลัง (Background) ที่จะมึระดับความเข้มของภาพ  $g(x,y)$  ค่อนข้างสูง(สว่าง) ดังเช่นภาพที่ 2.1 ซึ่งแสดงฮิสโตแกรมของระดับของระดับความเข้มของภาพที่ถูกแบ่งออกเป็น 256

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระดับ(ในกรณีของภาพขนาดละเอียด 8 บิต) จะเป็นที่ว่าการที่แยกกลุ่มข้อมูลออกเป็น 2 กลุ่มอย่างชัดเจนย่อมสามารถทำได้โดยการเลือกค่าเทรชโฮลด์ที่มีความเข้มอยู่ระหว่างกลุ่มทั้งสอง บนฮิสโตแกรมระดับความเข้มของภาพแล้วทำการตรวจสอบแต่ละจุดของภาพว่า ถ้ามีค่า  $g(x,y)$  น้อยกว่าค่าเทรชโฮลด์ถือว่าเป็นจุดภาพของวัตถุที่แสดงได้ ด้วยจุดค่าแตกหากว่าจุด  $g(x,y)$  นั้นที่ค่ามากกว่าหรือเท่ากับค่าเทรชโฮลด์ก็ถือว่าเป็นจุดภาพใน ส่วนพื้นหลังที่แสดงได้ด้วยจุดขาว ดังนั้นข้อมูลภาพ  $gthr(x,y)$  ที่ผ่านการทำเทรชโฮลด์สามารถนิยามได้ดังนี้

$$\begin{aligned} \text{IF } & g(x,y) \geq T & gthr(x,y) \text{ Object} = 255 \\ \text{Else} & & gthr(x,y) \text{ Background} = 0 \dots\dots (2.1) \end{aligned}$$

ซึ่งเงื่อนไขข้างต้นนี้เรานูमानว่า สนใจเฉพาะวัตถุที่สว่างกว่าพื้นหลัง หากว่าเป็นการหาวัตถุที่วางอยู่ในพื้นที่ที่สว่างกว่าวัตถุก็แสดงได้ว่า

$$\begin{aligned} \text{IF } & g(x,y) \leq T & gthr(x,y) \text{ Object} = 255 \\ \text{Else} & & gthr(x,y) \text{ Background} = 0 \dots\dots(2.2) \end{aligned}$$

โดยที่

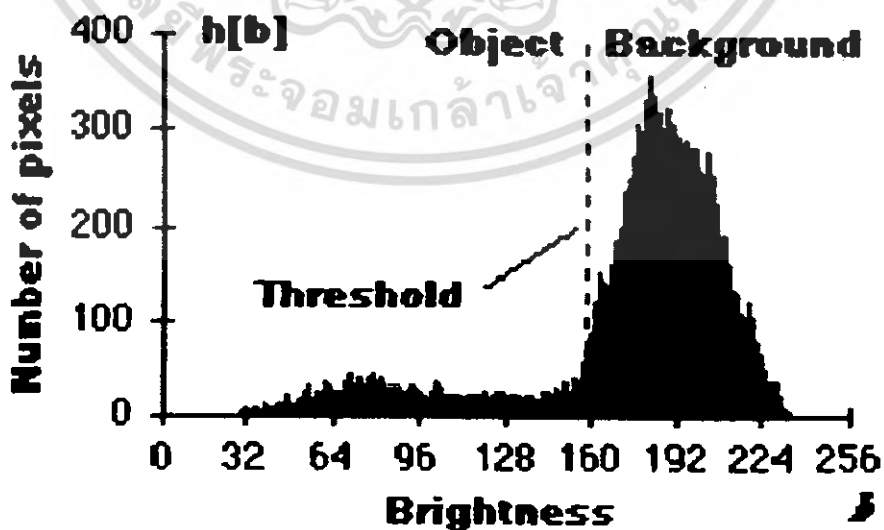
$gthr(x,y)$  คือ ข้อมูลภาพผลลัพธ์เป็นไบนารี

$g(x,y)$  คือ ข้อมูลภาพอินพุตที่มีระดับความเข้ม 0 ถึง L ระดับ

T คือ ค่าเทรชโฮลด์เป็นค่าคงที่ที่มีค่าระหว่าง 0 ถึง L

0 คือ จุดดำ (เป็นส่วนที่เป็นวัตถุ)

255 คือ จุดขาว (เป็นส่วนที่พื้นขาว)



รูปที่ 2.25 แสดงฮิสโตแกรมของระดับความเข้มของภาพขนาด 256 ระดับ

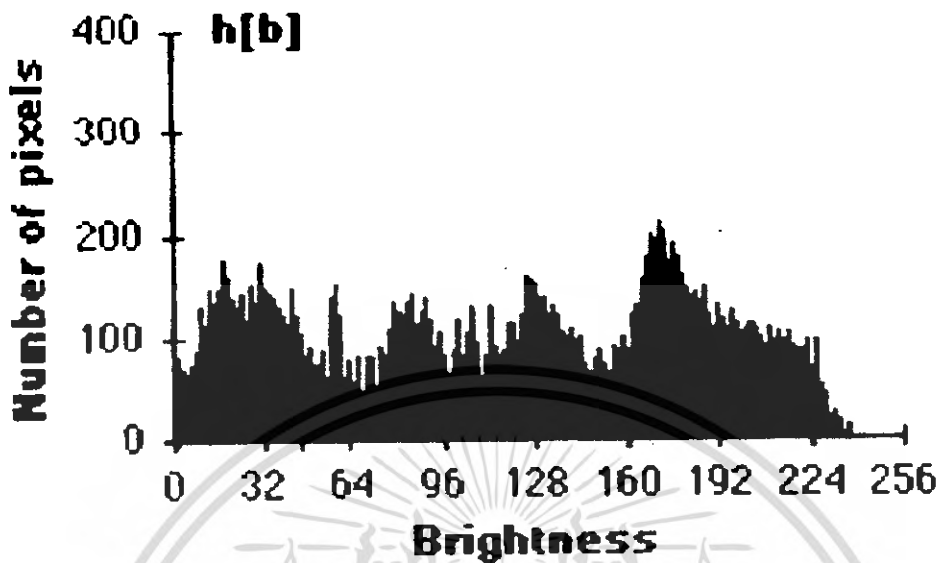
เมื่อได้ทำการศึกษาเกี่ยวกับการทำเทรชโฮลด์ไปแล้วภาพที่ผ่านการทำเทรชโฮลด์เราสามารถเรียกอีกอย่างหนึ่งว่าภาพ ไบนารี (Binary Image) ก็ย่อมเกิดคำถามที่จะตามมาว่า แล้วจะใช้ค่าเทรชโฮลด์เท่าไรดีละ ซึ่งการกำหนดค่าของการทำเทรชโฮลด์นี้สามารถทำได้ 2 รูปแบบ คือ การกำหนดค่าเทรชโฮลด์แบบคงที่ตลอดทั้งภาพ (Fixed Threshold) และอีกวิธีคือเลือกค่าเทรชโฮลด์จากระดับฮิสโตแกรม (Histogram-derived threshold)

### 2.8.2.1 เทรชโฮลด์แบบคงที่ตลอดทั้งภาพ (Fixed Threshold)

ซึ่งเป็นทางเลือกหนึ่งที่ถูกใช้ทำเทรชโฮลด์โดยไม่ได้ขึ้นกับข้อมูลภาพที่ได้นำมาประมวล โดยการทำเช่นนี้ได้ก็ต่อเมื่อเราทราบแล้วว่าวัตถุที่มีสีเข้มมากๆ ได้ถูกวางไว้บนพื้นหลังที่มีสีขาวหรือ อ่อนกว่าวัตถุ โดยที่พื้นหลังจะต้องมีความสว่างที่เท่ากันตลอด ดังนั้นแล้วเราสามารถเลือกค่าเทรชโฮลด์เป็น 128 จากระดับความเข้มของภาพระหว่าง 0 ถึง 255 ได้ซึ่งจะทำให้ผลลัพธ์ที่ได้ถูกต้องมากที่สุด แต่วิธีการนี้จะมีข้อจำกัดตรงที่หากว่าความสว่างของพื้นหลังมีค่าไม่คงที่แล้วจะทำให้เกิดการผิดพลาดในการตรวจจับวัตถุได้

### 2.8.2.2 เทรชโฮลด์จากระดับฮิสโตแกรม (Histogram-derived thresholds)

ในกรณีทั้งข้อมูลภาพมีความไม่สม่ำเสมอเกิดขึ้นในส่วนของวัตถุ หรือส่วนของพื้นหลังหรือ ในทั้งสองส่วนซึ่งภาพในลักษณะเช่นนี้ ฮิสโตแกรมระดับความเข้มของภาพที่เกิดขึ้นอาจมีลักษณะดังรูปที่ 2.2 ดังนั้นในการกำหนดค่าเทรชโฮลด์เพื่อจะนำมาใช้งานจึงสามารถใช้หลักการของการหาค่าเฉลี่ย (Mean) ระหว่างระดับความสว่างที่ความเข้มภาพสูงที่สุดกับระดับความมืดที่ความเข้มภาพสูงที่สุดเช่นกัน โดยสามารถแสดงสมการ ได้ดังต่อไปนี้



รูปที่ 2.26 กราฟฮิสโตแกรมที่พื้นหลังกับวัตถุมีระดับความเข้มภาพไม่แตกต่างกันมาก

$$T = \frac{h[g(x,y)](\text{Brightness}) + h[g(x,y)](\text{Darkness})}{2}$$

โดยที่

$h[g(x,y)]$  (Brightness) คือ ความสว่างจุดที่มีจำนวนของพิกเซลมากที่สุด

$h[g(x,y)]$  (Darkness) คือ ความมืดจุดที่มีจำนวนของพิกเซลมากที่สุด

ซึ่งเมื่อทำการคำนวณเทรซโฮลด์ได้แล้ว ก็สามารถทำการเซกเมนต์ภาพได้โดยนำค่าเทรซโฮลด์ที่ได้มาแทนค่าในสมการที่ 2.1 หรือ 2.2 ได้ทันที

### 2.9 การแยกส่วนสี (Color Segmentation)

ในโครงงานชุดนี้ได้ทำการเลือกใช้สีที่จะนำมาประมวลผลในโปรแกรมอยู่กันทั้งหมดอยู่ 3 สีด้วยกันซึ่งทั้งหมดจะเป็นแม่สีได้แก่ สีแดง (แทนสีของวัตถุปลายทาง) สีเขียว และสีน้ำเงิน ซึ่งเหตุผลที่ใช้แม่สีดังกล่าวก็เพราะว่า เราจะสามารถออกแบบโปรแกรมที่จะใช้ในการกรองสีได้ง่ายกว่าสีอื่นๆที่ไม่ใช่แม่สี ดังรูปที่ 2.5 แสดงฮิสโตแกรมของภาพแม่สีทั้งสามที่รับเข้ามาจากกล้องซึ่งได้ถูกแปลงให้เป็นภาพเกรย์ (Gray Image) ขนาดความละเอียดที่ 8 บิต ก่อนการคำนวณหา

ระดับฮิสโตแกรม เหตุที่ต้องแปลงภาพให้เป็นภาพเกรย์ก่อนเพราะว่าเพียงต้องการแสดงให้เห็นถึงเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่บนสื่อออนไลน์ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

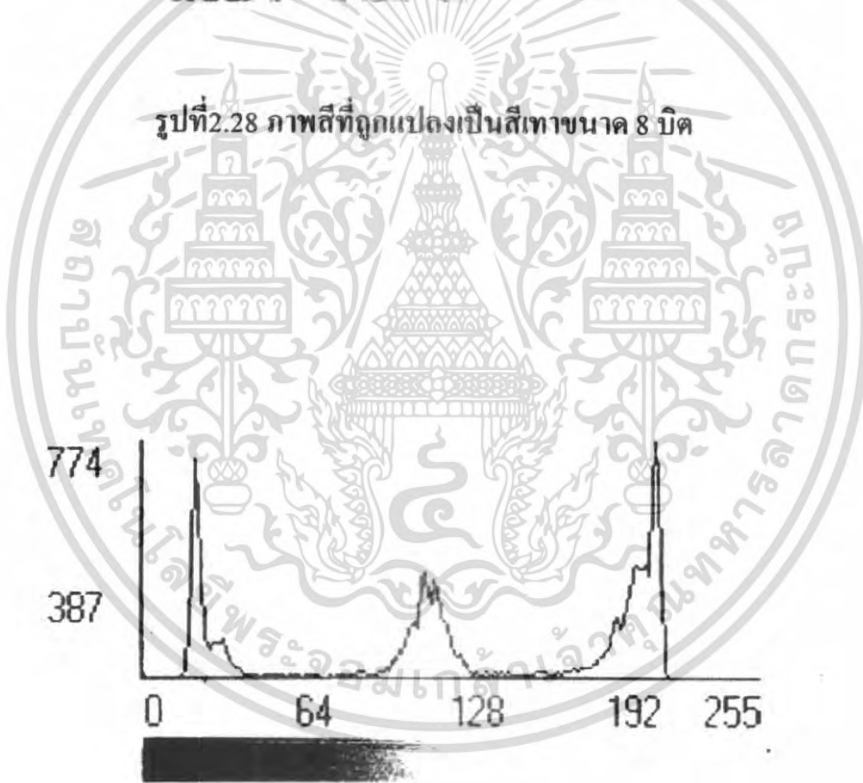
ความแตกต่างของกลุ่มขงจุดสี (Color Pixels) ของแม่สีทั้งสามสีซึ่งจะเห็นว่าเมื่อนำเอาภาพสีที่มีวัตถุแม่สีทั้งสามมาแปลงเป็นภาพเกรย์แล้วจะได้ระดับของฮิสโตแกรมที่มีความแตกต่างดังนี้คือ สีน้ำเงินที่ถูกแปลให้กลายเป็นระดับสีเทา (Gray Color) แล้วมีระดับความสว่างที่บริเวณ 0 <ระดับความสว่าง> 64 เพราะมีความสว่างน้อยกว่าแม่สีอื่น สีแดงจะอยู่บริเวณ 64 <ระดับความสว่าง> และสีเขียว 128 <ระดับความสว่าง> 255 จะเห็นว่าแม่สีต่างๆ นั้นจะกระจายออกเป็นกลุ่มอย่างเห็นได้ชัดเจน แสดงดังรูปต่อไปนี้



รูปที่ 2.27 ภาพสีขนาดจากกล้องขนาด 24 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Glaz Image



รูปที่ 2.29 แสดงระดับฮิสโตแกรมของรูปที่ 2.28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.9.1 การเข้าถึงระดับพิกเซลในภาพ (Pixel Accessing)

ภาพหนึ่งภาพหรือหนึ่งรูปนั้นจะประกอบไปด้วยจุดสีต่างๆ มากมายที่มีความแตกต่างกันของระดับสีเต็มไปหมด ซึ่งสีต่างๆ ที่เรามองกันล้วนเกิดการผสมกันระหว่างแม่สีทั้งสามสีคือ สีแดง สีเขียว และสีน้ำเงิน เมื่อพิจารณาเข้าไปถึงจุดเล็กสุดของภาพจะประกอบด้วยจุดสี 3 จุดซึ่งถูกเรียกว่าพิกเซล (Pixel) จะประกอบไปด้วยแม่สีทั้งสามได้กล่าวไปแล้วซึ่งได้แสดงดังรูปที่ 2.6



รูปที่ 2.30 แสดงจุดเล็กสุดของภาพที่ประกอบไปด้วยแม่สีแดง สีเขียว สีน้ำเงิน

ซึ่งสีต่างกันกับความสว่างของแต่ละพิกเซลก็จะไม่เหมือนกัน เช่น จุดของภาพจุดหนึ่ง ได้ถูกกำหนดให้แสดงสีแดงซึ่งเป็นแม่สี (ไม่มีสีอื่นนอกจากสีแดง) ดังนั้นจึงขอสมมุติให้พิกเซลทั้งสามเสมือนว่าเป็นหลอดไฟสามสีคือ หลอดไฟสีแดง หลอดไฟสีเขียว และหลอดไฟสีน้ำเงิน ตามลำดับ เพื่อให้ง่ายต่อความเข้าใจ ในการทำงานของหลอดไฟทั้งสามก็จะมีเพียงหลอดสีแดงเท่านั้นที่ติดสว่าง (ความสว่างถูกกำหนดโดยระดับสีสโตรแกรม 0-255 ระดับ หรือ 8 บิต) ส่วนสีที่เหลือก็จะดับอยู่แบบนั้นและถ้าจุดภาพหรือ Dot ดังกล่าวถูกกำหนดให้แสดงสีขาว หลอดไฟทั้งสามก็จะติดสว่างกันหมด และความสว่างของสีของทั้งสามหลอดก็ต้องเท่ากัน (ความสว่างเท่ากับ 255 เพราะถ้าต่ำกว่านี้จะเป็นสีเทา) เป็นต้น

ดังนั้นในการเข้าถึงข้อมูลในระดับพิกเซลก็เพื่อทำการอ่านค่าความสว่างของสีต่างๆ ว่ามีขนาดเท่าไรบ้างเพื่อนำมาเปรียบเทียบกับค่าคงที่ที่จะทำการแยกสี (Color Segmentation) ซึ่งการ

เปรียบเทียบค่านี้ก็คือการทำทรานส์โพลด์สีนั่นเอง ซึ่งจะมีหลักการอยู่ว่าจุดดังกล่าวมีระดับความสว่างเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้จะเขียนค่าการคำนวณว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของสีต่างๆอยู่ในช่วงของแม่สีใดบ้าง ถ้าสมมุติว่าภาพจากกล้องที่รับเข้ามามีความสว่างอยู่ในยาของสีน้ำเงินพอดี (เช่นความสว่างของพิเซล สีน้ำเงิน-สีแดง-สีเขียว = 150-90-65) ซึ่งภาพรวมของจุดสีดังกล่าวคือสีน้ำเงินเพียงแต่อาจไม่ใช่สีน้ำเงินที่แท้จริงซึ่งมีค่าเท่ากับ (255-0-0) ดังนั้นจึงต้องทำการปรับความสว่างของพิเซลในจุดดังกล่าว (Pixels in Dot) โดยวิธีการทำเทรซโฮลด์คิงที่ได้กล่าวเอาไว้แล้ว

หลังจากที่เราได้ทำการแยกสีหรือ (Color segmentation) ไปแล้วลำดับต่อไปก็จะเป็นการตรวจจับหรือตรวจหาชนิดของวัตถุ (ตัวหุ่นยนต์และจุดหมายปลายทาง) ว่าอยู่ตรงส่วนใดของภาพ ซึ่งเรียกวิธีการนี้ว่า (Object Tracking)

## 2.10 การติดตามวัตถุ (Object Tracking)

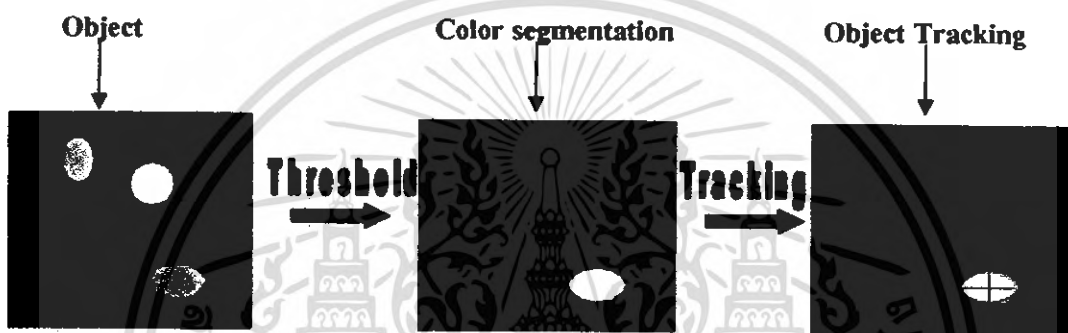
ระบบที่ใช้การประมวลผลรูปภาพ (Image Processing) เพื่อการคัดแยกหรือตรวจหาวัตถุที่ต้องการจากรูปภาพนั้น โดยส่วนมากแล้วจะต้องมีวิธีการใดวิธีการหนึ่งที่จะใช้ติดตามหรือแยกแยะวัตถุที่ต้องการออกจากวัตถุอื่นๆ ที่ปะปนมากับในรูปออกให้ได้ ซึ่งก็ได้มีผู้คิดค้นวิธีการอยู่หลายแบบด้วยกัน และในรายงานเล่มนี้ก็ได้ค้นคว้าหาวิธีจากแหล่งข้อมูลขนาดใหญ่ซึ่งก็คือ อินเทอร์เน็ต (Internet) ซึ่งได้พบวิธีที่น่าจะเป็นได้สำหรับการเขียนโปรแกรมมี 2 แบบดังนี้ วิธีแรก การแมชชิงกับแม่แบบ (Template Matching with Distance Transforms) หรือ (Model Comparing Method) ซึ่งวิธีการนี้เป็นวิธีที่มีความยืดหยุ่นค่อนข้างสูงใช้ได้กับรูปภาพที่มีสภาพแสงไม่คงที่ แต่ก็มีความซับซ้อนในการออกแบบและเขียนโปรแกรมอย่างมากและใช้เวลาในการทำงานสูง ซึ่งจะกล่าวให้เห็นถึงหลักการการทำงานของวิธีแรกอย่างคร่าวๆดังนี้

### 2.10.1 การแมชชิงโดยใช้การเทียบกับต้นแบบต้นแบบ (Model Comparing Method)

ในวิธีการนี้เราจะต้องมีสิ่งที่สำคัญอันดับแรกก็คือ ต้นแบบ (Model) ที่จะใช้เปรียบเทียบซึ่งอาจได้จากรูปถ่ายหน้าคนที่ต้องการทดสอบแล้วนำภาพที่ได้มาทำการตัดทอนส่วนที่ไม่จำเป็นในรูปทิ้งไป (เก็บเอาแต่ลักษณะเด่นของหน้าคนในรูป) หลังจากที่ได้โมเดลมาแล้วก็นำมาเก็บไว้เพื่อรอการเรียกใช้งาน ขั้นตอนต่อไปพิจารณารูปภาพที่จะนำมาทำการแมชชิงกับหน้าคน โดยจะเตรียมจากการแยกส่วนภาพ ซึ่งในที่นี้คือการหาขอบภาพ (เพื่อให้ง่ายต่อการเปรียบเทียบ) หรือที่รู้จักกันในชื่อ (Edge Detection) เมื่อได้ทำการเตรียมรูปภาพที่จะนำไปเปรียบเทียบกับตัวโมเดลที่เก็บไว้แล้วนั้น ขั้นตอนต่อไปก็เป็นส่วนของการเปรียบเทียบระหว่างโมเดลกับรูปภาพ ซึ่งจะมีวิธีการเพื่อใช้หาวิธีการเปรียบเทียบอีกขั้นหนึ่งก็คือ คิสแทนซ์ ทรานส์ฟอร์ม (Distance Transform) และหนึ่งในวิธีการคือวิธีของ (Hausdorff Transform) c) แต่จะไม่กล่าวถึงรายละเอียดในรายงานเล่มนี้

เนื่องจากเกินขอบเขต แต่ก็ได้รวบรวมเอกสารที่เกี่ยวข้องไว้ในซีดี-รอมเรียบร้อยแล้วและหลังจากที่ได้ทำการเปรียบเทียบระหว่างรูปภาพทั้งสองแล้ว ผลที่ได้ดังแสดงในรูปที่ 2.7

ในหลักการหัวข้อที่ 2.4.1 ที่ได้กล่าวมานั้นนอกจากจะสามารถทำการแทรกกิ่งใบหน้าคนได้แล้วถ้าหากนำไปทำการจัดการกับ โมเดลให้เป็นฐานข้อมูลที่เหมาะสมแล้วเราจะสามารถจดจำ ใบหน้าหรือรู้จักว่าคนที่กำลังอยู่ในภาพนั้นเป็นใครๆ เดียวกันกับที่เก็บไว้ในฐานข้อมูลหรือไม่ได้ก็อาจจะเรียกว่ารู้จักวัตถุ (Object Recognition) นั่นเอง



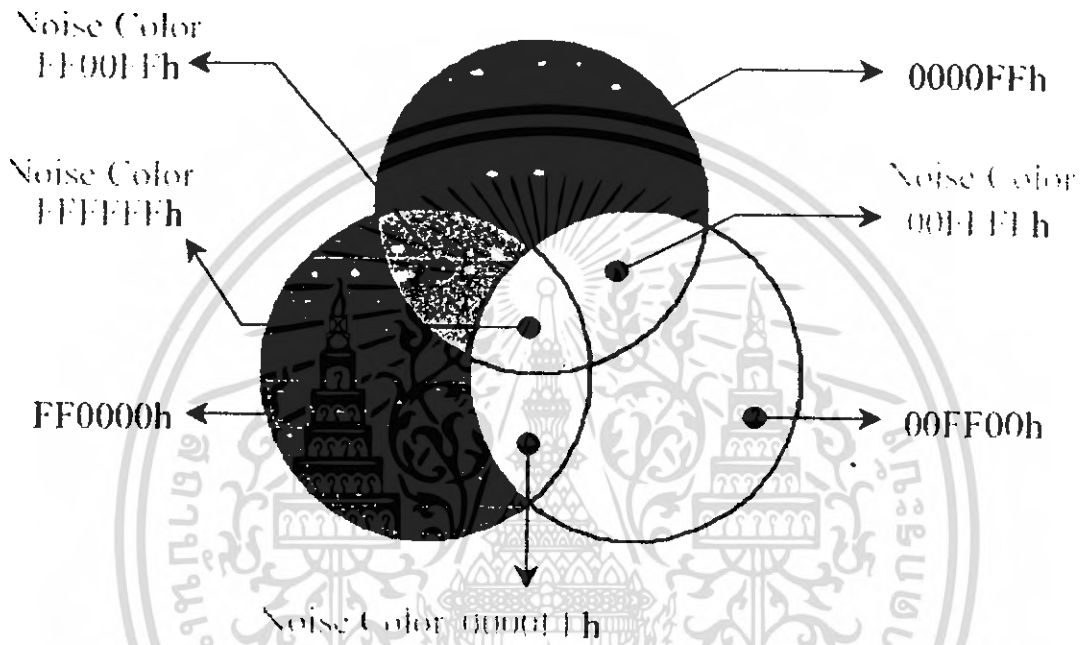
รูปที่ 2.31 แสดงการแทรกกิ่งโดยใช้การเปรียบเทียบกับโมเดล

### 2.10.2 การแทรกกิ่งโดยการคัดแยกสีหรือกรองสี (Color Filtration)

เป็นวิธีการที่ถูกนำมาใช้ในโครงการชุดนี้เหตุผลก็เพราะสีที่ใช้ในโครงการนี้มีเพียงแม่สี 3 สีคือ สีแดง สีเขียว และสีน้ำเงิน เท่านั้นทำให้มีความรวดเร็วในการประมวลผลเนื่องจากการแทรกกิ่งแบบนี้จะใช้หลักการกรองเอาสีอื่นที่ไม่ต้องการทิ้ง ให้เหลือไว้แค่สีของวัตถุที่ต้องการเท่านั้นแต่การกรองสีนั้นหากมีสัญญาณรบกวนที่เป็นสีกับวัตถุที่ต้องการหาในภาพ ก็จะเกิดการผิดพลาดในการแทรกกิ่งได้ ดังนั้นจะต้องคำนึงถึงขนาดของสิ่งที่จะแทรกกิ่งด้วยเพื่อตัดปัญหาของสัญญาณรบกวนออกไปจากนั้นจึงหาว่ากลุ่มของสีที่อยู่ในวัตถุที่มีขนาดเท่ากับที่ต้องการหรือไม่ ถ้าใช้ก็จะทำการหาขอบเขตหรือรัศมีของวัตถุดังกล่าว เพื่อหาตำแหน่งศูนย์กลางของวัตถุออกมาเราก็จะได้พิกัดที่นำไปใช้งานได้แล้วรูปที่ 2.8 แสดงการผสมสีกันของแม่สีทั้งสามซึ่งกำหนดให้แม่สีดังกล่าวมีระดับความสว่างของแต่ละแม่สีเท่ากับ  $255-255-255=R-G-B$  (คือระดับที่เป็นไปได้ของเลขฐานสองขนาด 8 บิต\*3 สี) หรือเขียนแทนด้วยสัญลักษณ์เลขฐานสิบหกจะได้เป็น FFFFFFFh จะเห็นว่าบริเวณที่มี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การซ้อนทับกันระหว่างแม่สีจะเกิดสีใหม่ขึ้นมาซึ่งในการแทรกก็งโดยการกรองสีจะถือว่าสีนั้นคือ สัญญาณรบกวนต้องกำจัดออกไปและจะกล่าวถึงวิธีการอีกครั้งในบทที่ 3



รูปที่ 2.32 แสดงการผสมสีกันของแม่สีทั้งสาม

### 2.11 การเกิดภาพสเตอริโอ (Stereo Imaging)

การเกิดภาพสเตอริโอเป็นการสร้างภาพของวัตถุด้วยกล้อง 2 ตัววางอยู่ข้างกันภาพที่ได้จาก กล้องทั้งสองสามารถนำมาใช้ในการหาพิกัด 3 มิติของวัตถุได้ พิจารณาแบบจำลองการเกิด ภาพสเตอริโอในรูปที่ 2.33 ระยะระหว่างจุดกึ่งกลางของเลนส์ของกล้องทั้งสองเราเรียกว่า Baseline (B) เราต้องการหาพิกัด  $(X, Y, Z)$  ของจุด  $w$  ซึ่งมีจุดภาพ  $(x_1, y_1)$  และ  $(x_2, y_2)$  เราสมมติว่ากล้องทั้งสองนั้นมีลักษณะที่เหมือนกัน และระบบพิกัดของกล้องทั้งสองนั้นวางอยู่ในแนวที่ตรงกัน (Perfectly Aligned) ถ้าระบบพิกัดของกล้องและของ โลกซ้อนทับกันพอดีระนาบ  $x y$  ของภาพวาง อยู่ในแนวเดียวกันกับระนาบ  $xy$  ของโลก ดังนั้นค่าพิกัด  $Z$  ของ  $w$  สำหรับระบบพิกัดกล้องทั้งสองมีค่าเดียว

ถ้าระบบพิกัดของกล้องที่หนึ่งและของ โลกซ้อนทับกันพอดี ดังแสดงในรูปที่ 2.33 เราได้ว่า เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น มิใช่เพื่อเผยแพร่โดยไม่ได้รับอนุญาต การค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$X_1 = \frac{X_1}{f} Z$$

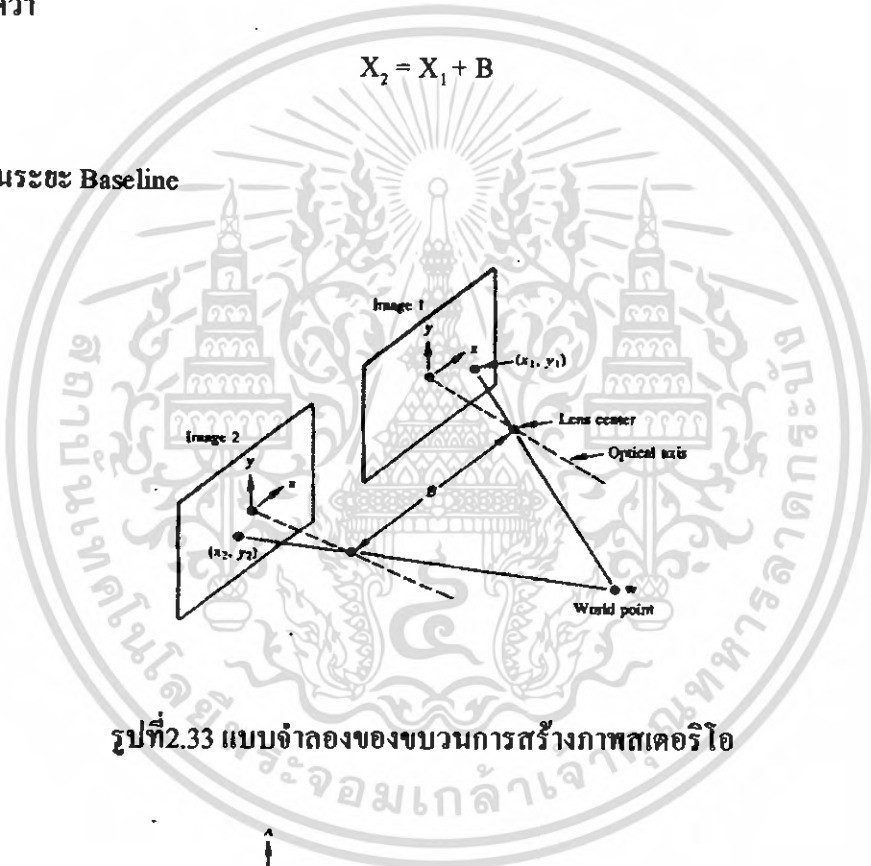
โดยที่ตัวห้อยบน  $X$  และ  $Z$  เป็นการบอกว่าพิกัดกล้องที่หนึ่งซ้อนทับกับพิกัดโลก ทำนองเดียวกัน ระบบพิกัดของกล้องที่สองและของโลกซ้อนทับกันพอดี เราได้ว่า

$$X_2 = \frac{X_2}{f} Z$$

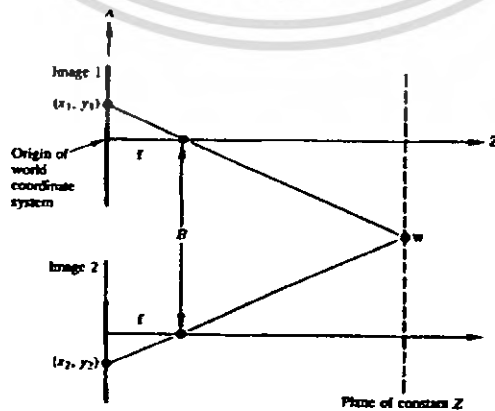
เนื่องจากการแยกกันของกล้องทั้งสองและเนื่องจาก  $w$  เป็นค่าเดียวสำหรับระบบพิกัดของทั้งสอง กล้อง เราได้ว่า

$$X_2 = X_1 + B$$

โดยที่  $B$  เป็นระยะ Baseline



รูปที่ 2.33 แบบจำลองของขบวนการสร้างภาพสเตอริโอ



รูปที่ 2.34 มุมของรูปที่ 2.33 โดยที่ระบบพิกัดของกล้องที่หนึ่งซ้อนทับกับระบบพิกัดของโลก

เอาแทนค่า (3) และ (4) ลงใน (1) และ (2) เราได้ว่าการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$X_1 = \frac{x_1}{f} Z \quad (5)$$

และ

$$X_{1+B} = \frac{x_2}{f} Z \quad (6)$$

ลบ (5) จาก (6) และแก้สมการหา Z เราได้ว่า

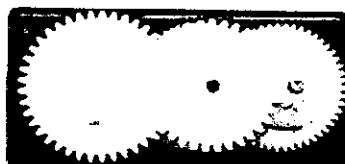
$$Z = \frac{fB}{x_2 - x_1}$$

นั่นหมายความว่าถ้าเราทราบผลต่างของพิกัดภาพ  $x_2$  และ  $x_1$  ระยะ Baseline และความยาวโฟกัสของกล้องเราสามารถคำนวณหาพิกัด Z ของ w ได้ สำหรับพิกัด X และ Y สามารถคำนวณโดยใช้ (5) โดยใช้พิกัด  $(x_1, y_1)$  หรือ  $(x_2, y_2)$

## 2.12 หลักการทำงานของ DC SERVO MOTOR

ตามปกติทั่วไปแล้วนั้น DC SERVO MOTOR นั้น จะสามารถหมุนไปในทิศทางตามเข็มนาฬิกาและทวนเข็มนาฬิกาได้เพียงแค่ 180 องศา หรือ ครึ่งวงกลมเท่านั้น ซึ่งวิธีการสั่งงานให้ SERVO หมุนไปในตำแหน่งใดๆนั้น จะอาศัยลักษณะของสัญญาณ Pulse เป็นตัวบ่งบอก แต่สำหรับ DC SERVO MOTOR ที่เรานำมาใช้ในการขับเคลื่อนล้อเพื่อนำพาตัวรถให้เคลื่อนที่ไปในทิศทางต่างๆได้นั้น จะต้องทำการดัดแปลงหรือ Modify ให้สามารถหมุนได้รอบตัวหรือ 360 องศา เสียก่อน โดย DC SERVO MOTOR ที่ได้จัดไว้ในชุดของ ET-ROBOT RD2 นั้น ได้ผ่านการปรับแต่งการทำงานของมอเตอร์ให้สามารถหมุนเป็นวงรอบ (360 องศา) ได้เป็นที่เรียบร้อยแล้วโดยวิธีการควบคุมให้มอเตอร์ซึ่งทำการดัดแปลงแล้วให้หมุนไปในทิศทางต่างๆนั้น จะมีลักษณะดังนี้

Servo Motor คือ มอเตอร์ ไฟฟ้ากระแสตรง DC MOTOR ภายใน Servo motor ก็จะมีส่วนของเพ็ืองเกียร์และบอร์ดควบคุม มีสายสัญญาณในการควบคุมการหมุน ของ Servo motor 1 เส้น และอีก 2 เส้นเป็น VCC กับ Ground โดยตัวของ Servo motor จะสามารถหมุนได้ 180 องศา แต่จะมีบางรุ่นที่สามารถหมุนได้ถึง 210 องศา โดยสามารถที่จะหมุนไปทางซ้าย 90 องศา ทางขวาอีก 90 องศา



รูปที่ 2.35 ชุดเพ็ืองเกียร์ ส่วนประกอบหลักของ Servo motor มีน้ำหนักเบา และส่งแรงบิดสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.36 บอร์ดควบคุม ของ Servo motor ตามสัญญาณ pulse

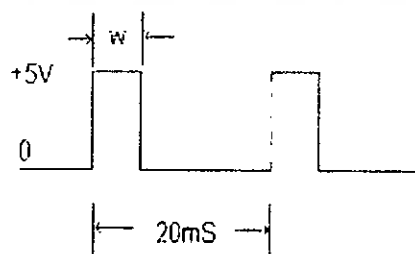
### ส่วนประกอบของ Servo motor

ตามท้องตลาดทั่วไปก็จะมี Servo motor อยู่หลายยี่ห้อและราคาก็แตกต่างกันออก อย่างไรก็ตาม Servo motor ก็จะมีคุณสมบัติคล้าย ๆ กันดังนี้

1. Servo motor จะหมุนเมื่อมีสัญญาณ pulse เข้าไป โดยสัญญาณ ความกว้าง pulse นี้จะทำหน้าที่ควบคุมการหมุนและทิศทางของ Servo motor
2. กิน ไฟประมาณ 4-6 volt กินกระแส 9.7 mA (Idle) และ 130 mA (Moving)
3. สามารถหมุนได้ 180 องศา (บางรุ่นหมุนได้ 210 องศา) ที่เป็นเช่นนี้ก็เพราะป้องกันความเสียหายที่อาจเกิดขึ้นกับวงจรภายใน ดังจะได้ศึกษาต่อไปในหลักการทำงาน
4. ขนาดเล็ก น้ำหนักเบา

### หลักการการทำงานของ Servo Motor

เราสามารถควบคุมการทำงานของ Servo motor ได้โดยป้อนสัญญาณความกว้าง pulse ให้กับ Servo motor ซึ่งความกว้างของ สัญญาณ pulse นี้จะไปกำหนดตำแหน่งของ Servo motor ว่า จะหมุนไปอยู่ตำแหน่งใด ซึ่งเราสามารถป้อนความกว้าง pulse ได้ตั้งแต่ 1.0 ms -2.0 ms โดยที่ คาบเวลา (Period) ในการส่งสัญญาณความกว้าง pulse เท่ากับ 20 ms ดังรูป



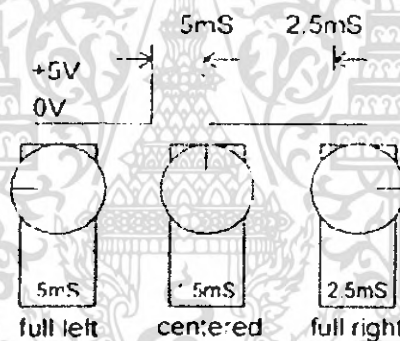
รูปที่ 2.37 รูปของ สัญญาณที่สามารถป้อนให้กับ Servo motor

โดยจะมีจุดที่อ้างอิง 3 จุดด้วยกันคือ

1. ตำแหน่งซ้ายสุด เราสามารถกำหนดให้ Servo motor เคลื่อนที่ไปยังตำแหน่งซ้ายสุด (-90 องศา) ได้ โดยการป้อนความกว้าง Pulse เท่ากับ 0.5 ms

2. ตำแหน่งตรงกลาง เราสามารถกำหนดให้ Servo motor เคลื่อนที่ไปยังตำแหน่งตรงกลาง (0 องศา) ได้ โดยการป้อนความกว้าง Pulse เท่ากับ 1.5 ms

3. ตำแหน่งขวาสุด เราสามารถกำหนดให้ Servo motor เคลื่อนที่ไปยังตำแหน่งขวาสุด (90 องศา) ได้ โดยการป้อนความกว้าง pulse เท่ากับ 2.5 ms จากจุดอ้างอิงทั้ง 3 เราสามารถที่จะคำนวณตำแหน่งที่ต้องการให้ Servo motor หมุนไปได้ เช่นถ้าเราต้องการให้ Servo motor หมุนไปยังตำแหน่ง -45 องศาเราก็ทำการป้อน  $(0.5+1.5)/2=1.0$  ms ให้กับ Servo motor ดังรูป



รูปที่ 2.38 ตำแหน่งการหมุนของ servo motor

Max-Min pulse ความกว้าง pulse ที่สามารถป้อนให้กับ Servo motor ได้

1. ป้อน 0.5 ms Servo จะหมุนไปตำแหน่ง ซ้ายสุด
2. ป้อน 1.5 ms Servo จะหมุนไปตำแหน่งตรงกลาง
3. ป้อน 2.5 ms Servo จะหมุนไปยังตำแหน่ง ขวาสุด

สัญญาณ Pulse ที่สามารถป้อนได้ เราสามารถป้อนสัญญาณความกว้าง pulse ให้กับ Servo motor ได้ จากรูป

1. ป้อน 1.0 ms Servo หมุนไปยังตำแหน่ง -45 องศา
2. ป้อน 2.0 ms Servo หมุนไปยังตำแหน่ง +45 องศา

### ข้อควรระวัง

ความกว้างของสัญญาณ pulse ที่กล่าวไว้นั้นเป็ยเพียงค่าประมาณเท่านั้น ซึ่ง Servo motor ที่มีอยู่ตามท้องตลาดอาจจะใช้ สัญญาณ pulse ที่แตกต่างจากนี้ก็ได้ และแต่ละยี่ห้ออาจจะใช้ความกว้างของสัญญาณแตกต่างกันก็ได้ ดังนั้น เพื่อให้แน่ใจควรศึกษาคู่มือ ของ Servo นั้น ๆ ให้เข้าใจก่อนอีกประเด็นหนึ่งที่ต้องกล่าวถึงก็คือ ทิศทางการหมุนของ Servo motor กล่าวคือ Servo motor แต่ละยี่ห้ออาจจะไม่หมุนไป ในทิศทางเดียวกันถ้าหากเราป้อนสัญญาณ pulse ให้ Servo motor ขนาดเท่ากัน เช่นสมมุติว่าถ้าเราป้อน สัญญาณ pulse 2.0 ms บางยี่ห้ออาจจะหมุนไปตามเข็มนาฬิกาไปที่ +45 องศา เช่นยี่ห้อ Hitec และมีบางยี่ห้ออาจจะหมุนทวน เข็มนาฬิกาไปที่ -45 องศา เช่นยี่ห้อ GWS เป็นต้นส่วนการกำหนดความเร็วของ Servo motor เราก็สามารถทำได้โดย การป้อนค่าความกว้างของสัญญาณ pulse ไปเป็น step ทุก 20 ms กล่าวคือ ถ้าเราอยากจะให้ Servo เคลื่อนอย่างรวดเร็วก็สามารถทำได้โดย ป้อน สัญญาณ pulse โดยใช้ Step น้อย ๆ ออกตัวอย่างเช่น ต้องการให้ Servo motor เคลื่อนที่ จากตำแหน่งตรงกลางไปซ้ายสุดอย่างรวดเร็ว

ป้อน 1.5 -----> 0.5 ms

ถ้าต้องการให้ช้าลง ก็สามารถทำได้ 1.5 ms----->1.0 ms ----->0.5 ms

อีกนัยหนึ่งคือ ถ้าต้องการให้ช้ามากเท่าใดก็ต้องเพิ่ม Step ให้มากขึ้นตามไปด้วย โดยเราต้อง ป้อนให้กับ Servo ทุก ๆ 20 ms นั้นเอง

### บทที่ 3

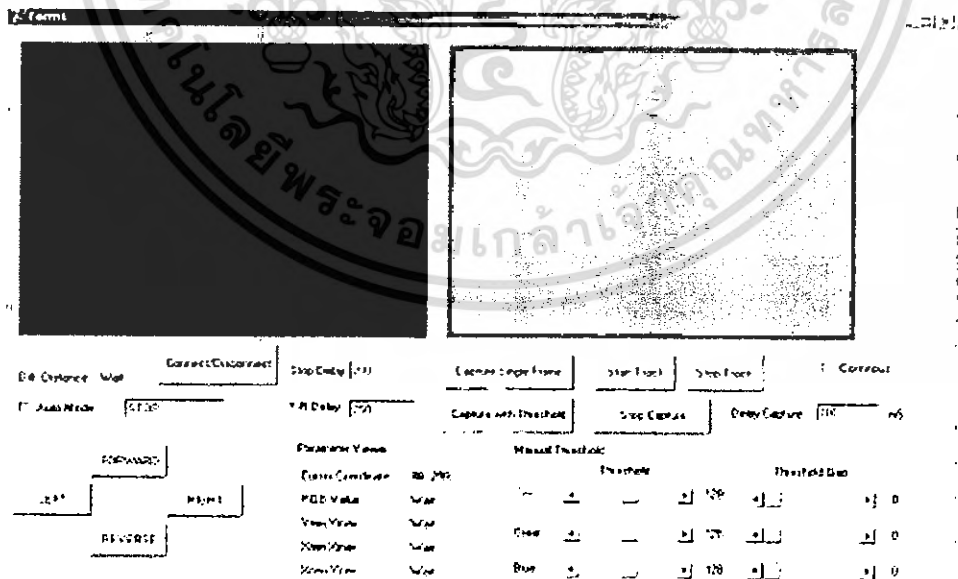
#### การออกแบบและการสร้าง

##### 3.1 แนวคิดในการออกแบบโครงการ

แนวความคิดการออกแบบโครงการนี้จุดนี้จะแยกออกเป็น 3 ส่วนใหญ่ๆด้วยกันคือ ส่วนแรกคือส่วนของโปรแกรมที่จะถูกใช้ภายในเครื่องคอมพิวเตอร์ ส่วนที่สองคือ ส่วนของตัวหุ่นยนต์ และส่วนที่สาม คือสนามที่จะถูกใช้ในการทดสอบโครงการ โดยสามารถแยกอธิบายทั้งสามส่วนได้ดังนี้

##### 3.2 โปรแกรมภายในเครื่องคอมพิวเตอร์

โปรแกรมที่ใช้งานเพื่อทำการติดต่อและควบคุมตัวหุ่นยนต์นั้นเป็นโปรแกรมในลักษณะของกราฟฟิคยูสเซอร์อินเตอร์เฟส (Graphics User Interface) คือ โปรแกรมที่ใช้ในการแสดงผลแบบรูปภาพเพื่อให้ผู้ใช้งานสามารถทำงานได้โดยไม่ต้องยุ่งกับคำสั่งจำนวนมาก และโปรแกรมที่ใช้ในการสร้างของโครงการนี้คือ บอแดน เคลไฟล์ 7.0 (Delphi 7.0) ซึ่งโปรแกรมที่ใช้ภาษาปาสคาล เป็นภาษาพื้นฐานในการเขียนโปรแกรม ซึ่งหน้าตาของโปรแกรมที่ได้สร้างมาแล้วนั้นมีรูปร่างหน้าตาตามรูปที่ 3.1



รูปที่ 3.1 แสดงโปรแกรมที่สร้างด้วย Delphi 7.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนที่เป็นหัวใจหลักของ โครงการนี้คือการประมวลผลภาพหรือ (Image Processing) ซึ่งเป็นโปรแกรมสำหรับใช้ในการประมวลสัญญาณที่รับเข้ามาจากกล้องวิดีโอ แล้วจะผลที่ได้จากกระบวนการไปควบคุมหุ่นยนต์โดยผ่านทางพอร์ตอนุกรม (Serial Port) ของเครื่องคอมพิวเตอร์อีกต่อหนึ่งโดยจะมีวิธีการออกแบบโปรแกรมดังนี้

### 3.2.1 การรับภาพจากกล้องวิดีโอ (Image Acquisition)

ก่อนอื่นขออธิบายก่อนว่าโปรแกรมเคลไฟล์ คือซอร์ฟแวร์หรือเครื่องมือที่ใช้สำหรับเพื่อเขียนหรือสร้างพัฒนาซอร์ฟแวร์ของเคลไฟล์ นั้นจัดเป็นเครื่องมือเขียน โปรแกรมชนิดวิซวล เช่นเดียวกับโปรแกรม ซี ++ บิวเดอร์ (Boland C++ Builder) วิซวลเบสิก (Visual Basic) วิซวลซี ++ (Visual C++) เป็นต้น

ซึ่งในการเขียนโปรแกรมจะมีเครื่องมือพื้นฐานเตรียมให้ใช้งาน ได้อย่างสะดวกเรียนว่า วิซวลคอมโพเนนท์ไลบรารี VCL (Visual Component Library) และภายในโครงการก็ได้มีการ VCL ที่มีชื่อว่า tscap32 ซึ่งนอกจากจะใช้เป็นเครื่องมือในการรับสัญญาณภาพจากกล้องวิดีโอแล้ว ยังมีคำสั่งที่รองรับการประมวลผลภาพให้อย่างมาก

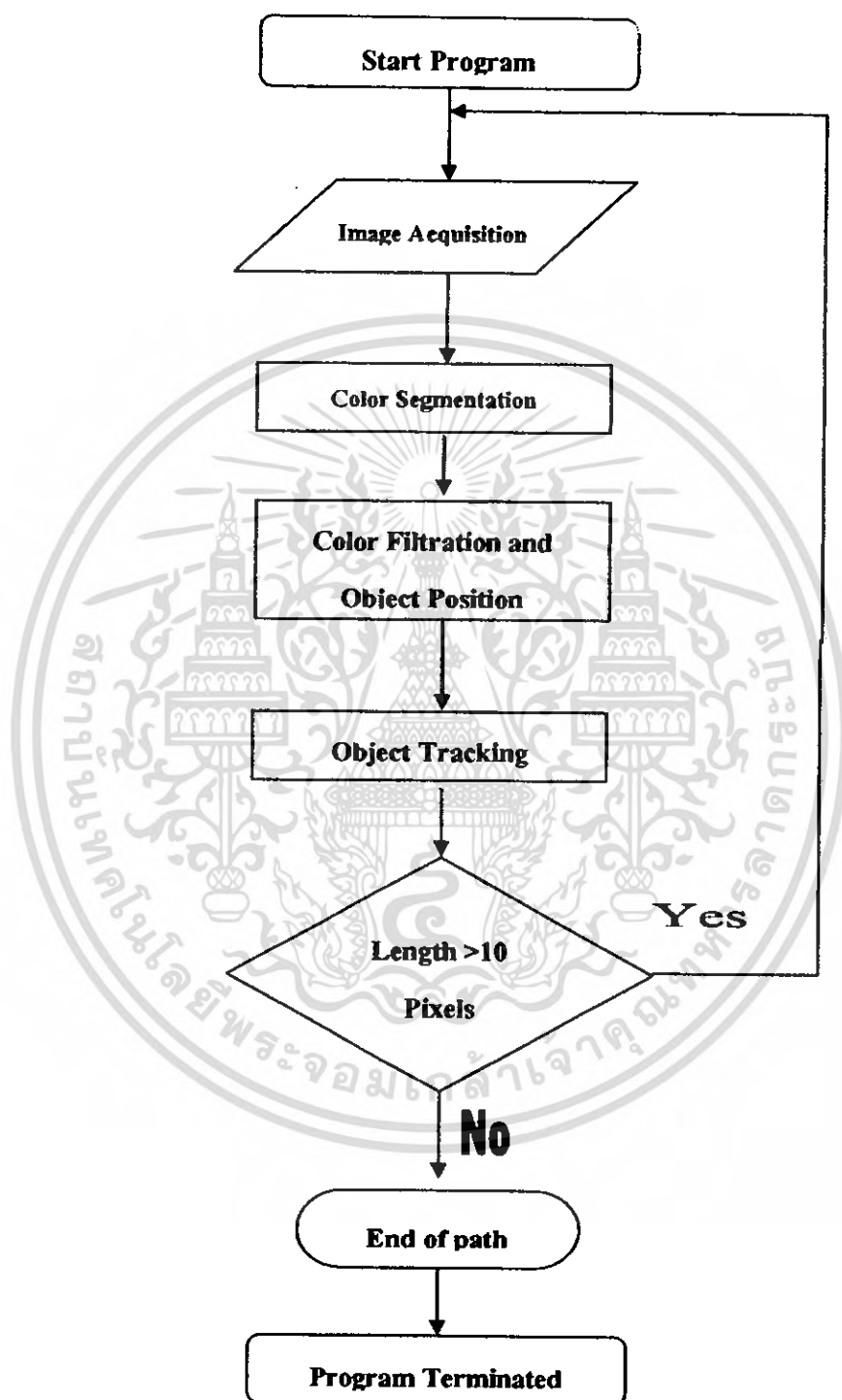
### 3.2.2 การควบคุมพอร์ตอนุกรม (Serial Port Controlling)

ข้อจำกัดของ เคลไฟล์ (Delphi 7.0) ที่ทำงานบนระบบปฏิบัติการวินโดวส์เอ็กซ์พี และเอ็นที ก็คือไม่สามารถติดต่อกับพอร์ตของเครื่องคอมพิวเตอร์โดยใช้ภาษาเคลไฟล์ได้โดยตรงแต่ไม่มีปัญหาเพราะจะต้องใช้การติดต่อกับพอร์ตที่มีอยู่ทางอ้อมแทน โดยการติดต่อผ่านทางซอร์ฟแวร์ ไดรเวอร์แทน

ดังนั้นวิธีการที่จะสามารถติดต่อกับพอร์ตต่างๆของเครื่องคอมพิวเตอร์บนระบบปฏิบัติการดังกล่าว ดังนั้นโครงการนี้ได้ใช้คอมโพเนนท์ที่ติดต่อกับพอร์ตอนุกรมที่มีชื่อว่าCommDrv32 ซึ่งเป็นคอนโทรลเลอร์ที่ทำหน้าที่ติดต่อกับพอร์ตของเครื่องคอมพิวเตอร์ที่ต้องใช้งาน

### 3.2.3 การทำงานของโปรแกรม

ในการเขียน โปรแกรมจะทำการแยกส่วนของโปรแกรมที่ถูกเรียกใช้งานบ่อยๆ ให้อยู่ในรูปแบบของโปรแกรมย่อยหรือ ซับรูทีน (Sub Routine) เพื่อให้โปรแกรมสามารถแก้ไขได้สะดวกเมื่อเกิดปัญหาขึ้น โดยจะแสดงส่วนของโปรแกรมหลักได้ดังโฟลวชาร์ตต่อไปนี้



รูปที่ 3.2 โฟลวชาร์ตโปรแกรมหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### Image Acquisition

การทำงานเริ่มโปรแกรมจะทำการรับภาพจากกล้องวิดีโอ ผ่านคอมพิวเตอร์ที่ชื่อ iscap32 เพื่อนำมาแสดงผลที่หน้าจอซึ่งจะตั้งความลึกของสีไว้ที่ 24 บิต และความละเอียดของภาพตั้งไว้ที่ 320x240 Pixels และต่อจากนั้นภาพที่ได้จะถูกนำไปทำการแยกสี (Color Segmentation)

### Color Segmentation

เพื่อปรับภาพที่ได้ให้เหมาะสมก่อนนำไปตัดแยกสี เหตุที่ต้องทำก็เพราะในภาพที่รับเข้ามาจากกล้องวิดีโอ นั้นยังมีปัญหาจากแสงที่ไม่สม่ำเสมอ ที่มาดกกระทบกับวัตถุทำให้สีของวัตถุที่กล้องจับได้นั้นเกิดการผิดเพี้ยน เช่นความมืดของสีที่บริเวณขอบของภาพ สีของวัตถุที่ไม่เท่ากันเป็นต้น ดังนั้นการแยกสีดังกล่าวก็เพื่อให้เกิดความชัดเจนสีของทั้งหมดออกมาเท่ากันตลอดทั้งภาพ

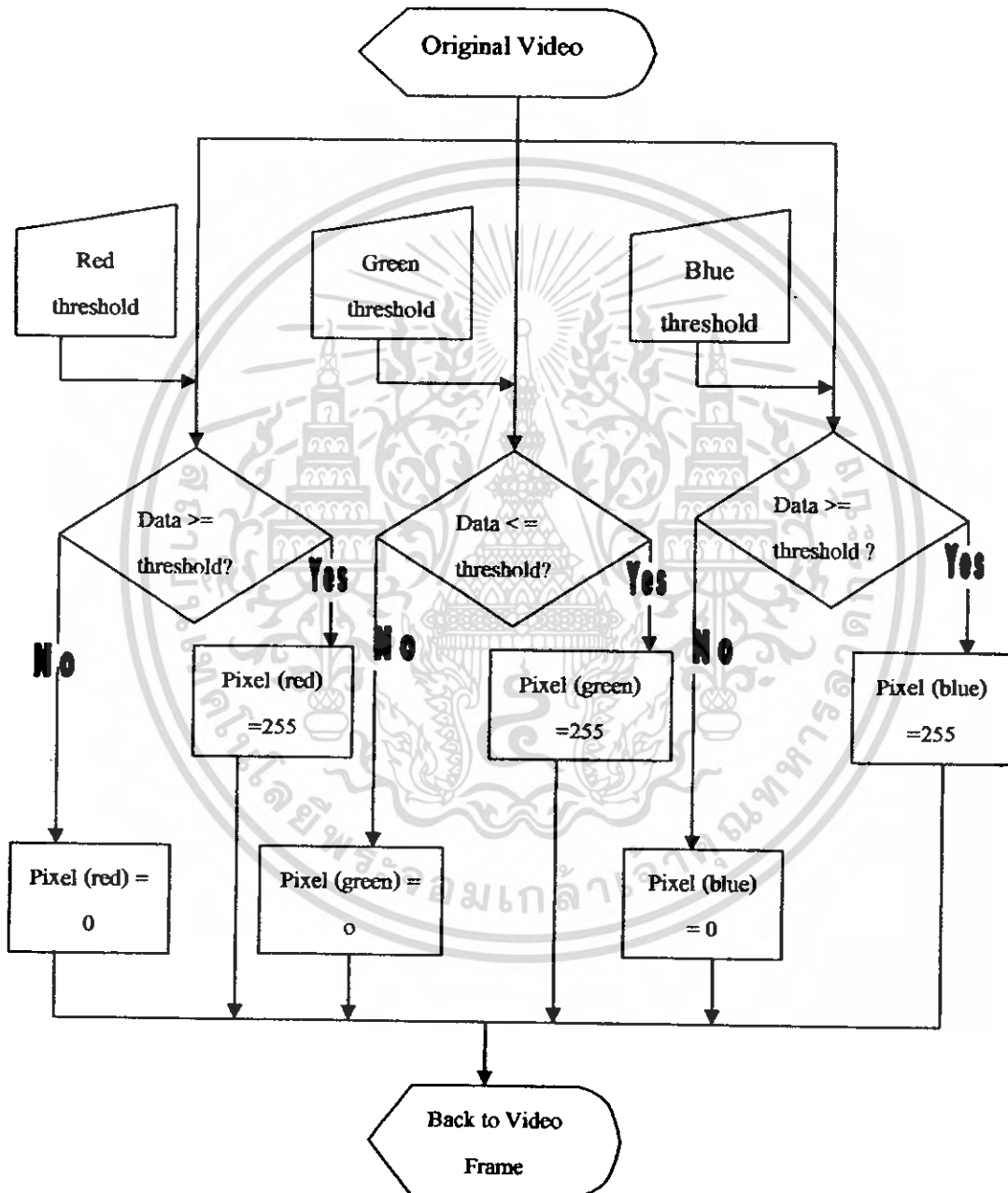
### Color Filtration and object Positioning

ต่อไปหลังจากที่เราได้ภาพที่เหมาะสมที่จะใช้ในการคัดแยกสีของวัตถุแล้ว เราก็จะทำการกรองสีในส่วนขอโปรแกรมย่อย ซึ่งในส่วนนี้จะมีโปรแกรมย่อยอยู่ 3 ส่วนด้วยกันคือ โปรแกรมวัตถุกรองสีแดง โปรแกรมกรองวัตถุสีเขียว โปรแกรมกรองวัตถุสีน้ำเงิน ซึ่งหลังจากได้ทำการกรองเอาสีทั้งสามออกมาจากกันแล้ว จากนั้นจึงทำการหาตำแหน่งของวัตถุในภาพว่าอยู่ ณ ตำแหน่งของภาพโดยการอ้างอิงกับจำนวนพิกเซลในภาพนั้น โดยจะหาตำแหน่งเพียงสีเขียวเท่านั้นคือ สีแดง

### Object Tracking

ต่อไปเป็นโปรแกรมสำหรับการติดตามวัตถุ ซึ่งวัตถุที่จะถูกติดตามคือสีแดง โดยทำการรับค่าตำแหน่งของวัตถุสีแดงจากโปรแกรมย่อยก่อนหน้านี้มาทำการหาพิกัดของวัตถุ เพื่อเป็นตัวบอกทิศทางละจุดหมายที่หุ่นยนต์จะต้องเดินไป ทิศทางหาได้จากพิกัดของจุดกึ่งกลางวัตถุ(Xcen) นำมาลบกับจุดกึ่งกลางของภาพ (ความกว้างของภาพ/2) ถ้าค่าเป็นบวกแสดงว่าวัตถุอยู่ทางด้านขวามือ หรือถ้าเป็นลบแสดงว่าวัตถุอยู่ทางซ้าย หลังจากรู้ทิศทางแล้วก็จะทำการสั่งให้หุ่นยนต์เคลื่อนที่ตามวัตถุจะอยู่ตรงกลาง ต่อไปจะทำการตรวจสอบความห่างของวัตถุ โดยเทียบจากขนาดของภาพวัตถุ ถ้าภาพวัตถุเล็กแสดงว่าวัตถุอยู่ไกลขึ้นหุ่นยนต์ก็จะเดินหน้า แต่ถ้าภาพวัตถุใหญ่ขึ้นแสดงว่าวัตถุเข้ามาใกล้หุ่นยนต์ก็จะถอยหลังเพื่อรักษาระยะความห่าง

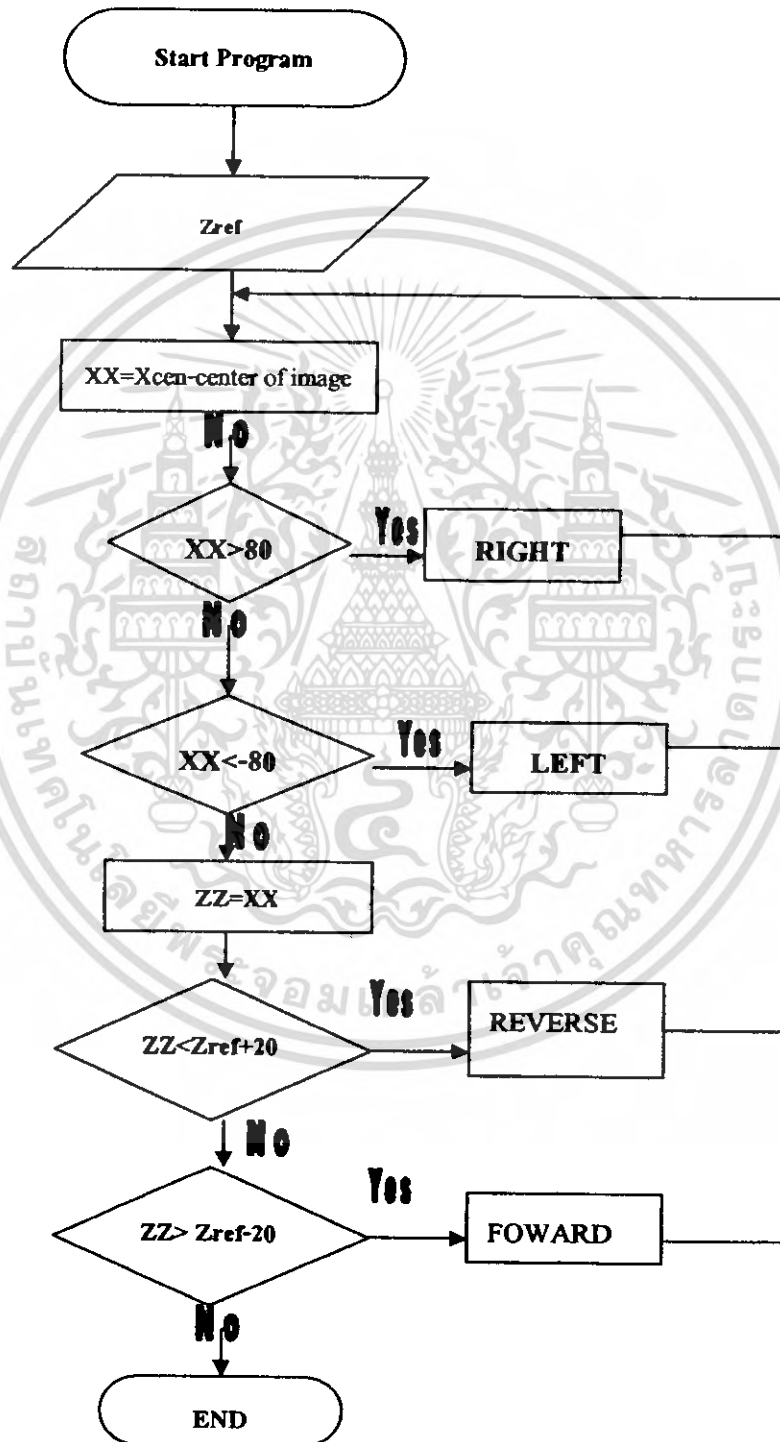
## 3.2.3.1 โฟลว์ชาร์ตแยกส่วนสี



รูปที่ 3.3 โฟลว์ชาร์ตโปรแกรมแยกส่วนสี

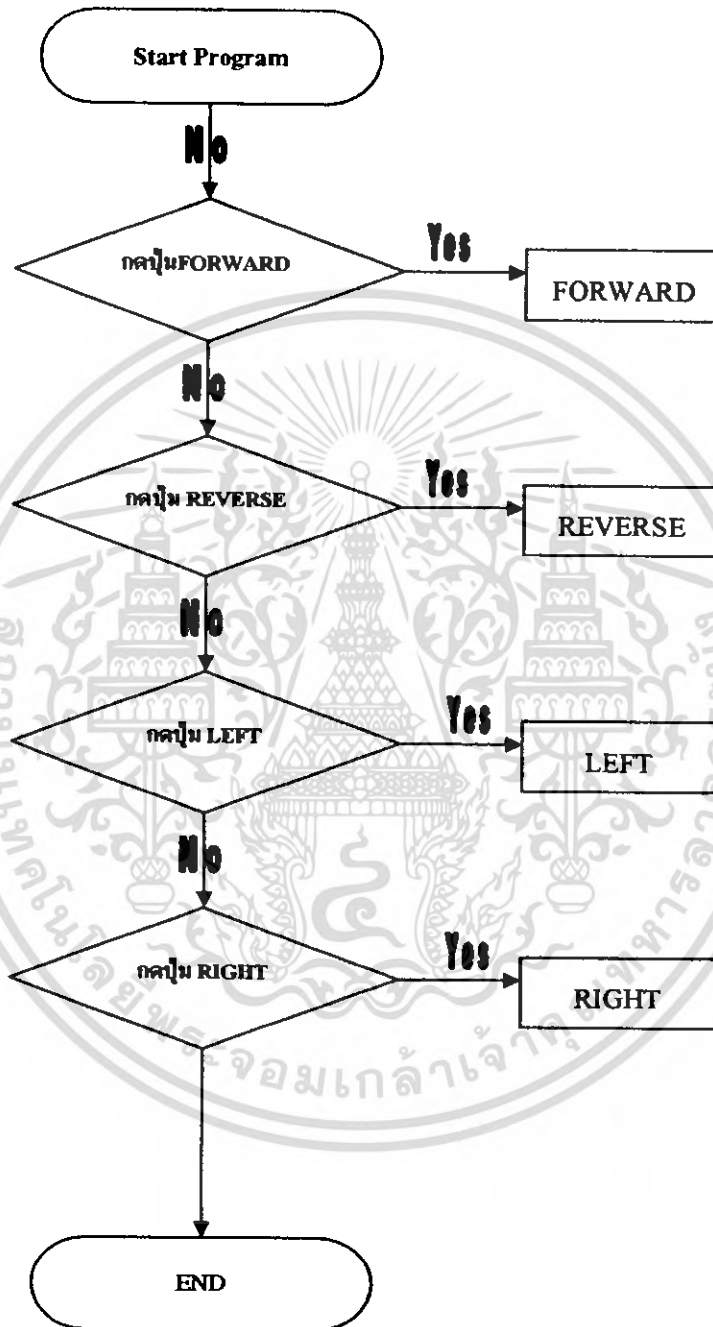
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 3.2.3.2 โฟลว์ชาร์ตติดตามวัตถุ



รูปที่ 3.4 โฟลวชาร์ทโปรแกรมแทร็กกิ้งวัตถุ

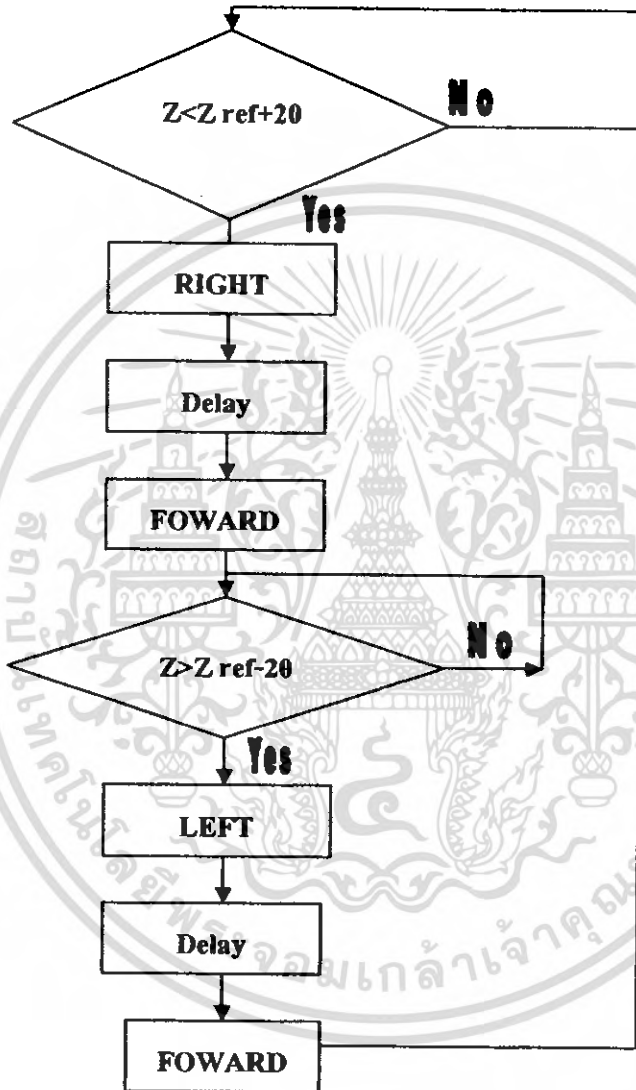
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 โฟลวชาร์ทโปรแกรมการควบคุมหุ่นยนต์แบบ Manual

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 3.2.3.3 โพลัวซาร์ตหลบหลีกวัตถุ

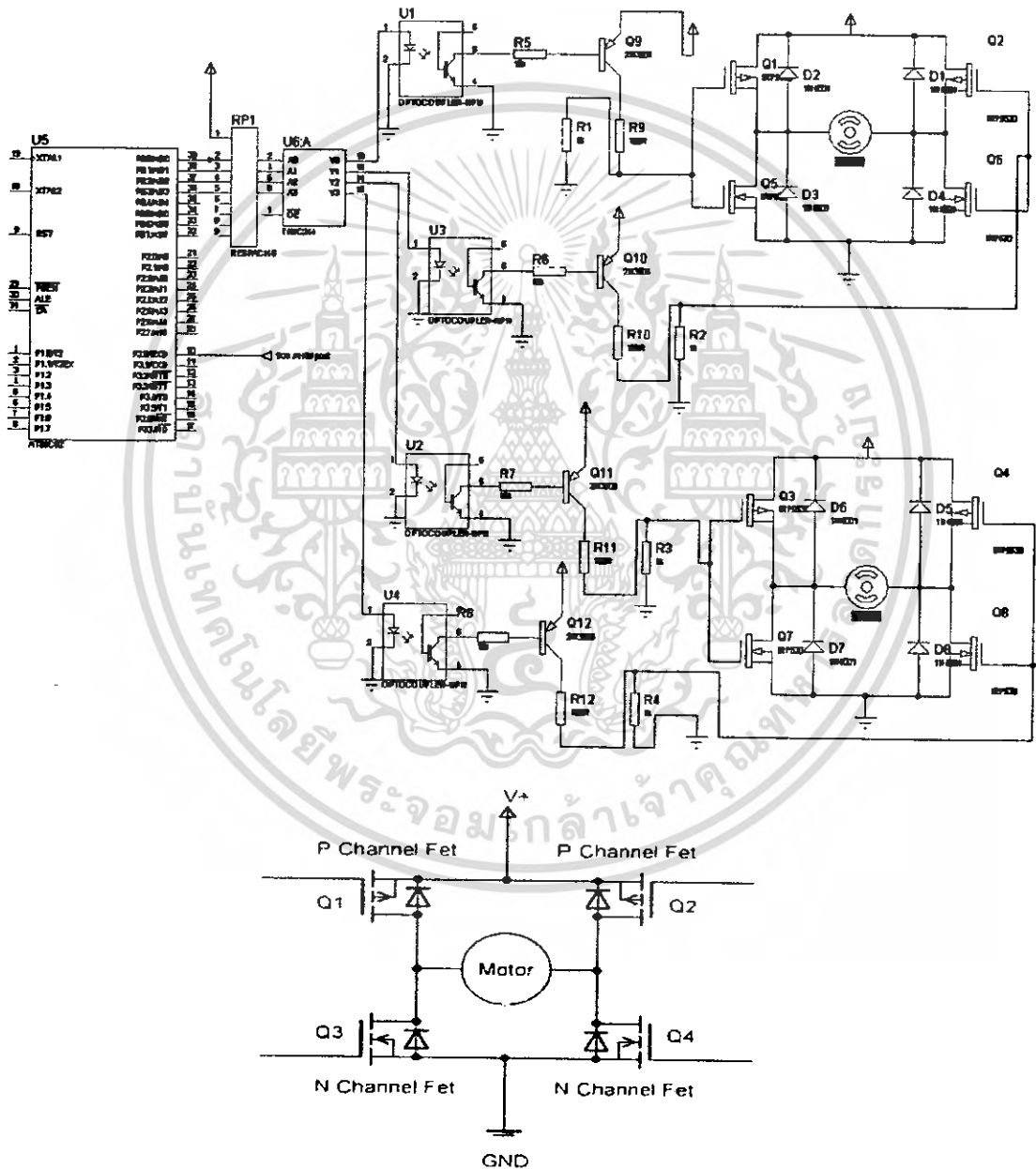


รูปที่ 3.6 โพลัวซาร์ตหลบหลีกวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

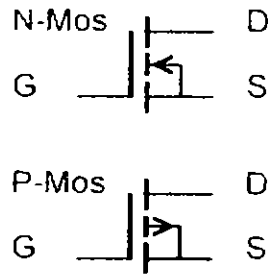
### 3.3 การออกแบบ ภาคขับเคลื่อนมอเตอร์

ในโครงการนี้ ใช้มอเตอร์กระแสตรง คำนึงภาคขับเคลื่อนมอเตอร์จึงใช้วงจร H-Bridge โดยใช้มอเตอร์เฟสชนิด ที-แซลแนล เบอร์ IRF9530 กับ เอ็น-แซลแนล เบอร์ IRF530 ในการควบคุมการไหลของกระแส



รูปที่ 3.7 วงจรภาคขับเคลื่อนมอเตอร์แบบ H-Bridge

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 สัญลักษณ์ของ MOSFET

โดยเมื่อ Q1, Q4 ทำงาน และ Q3, Q2 ไม่ทำงาน จะทำให้มีกระแสไหลจาก Q1 ผ่าน มอเตอร์ ไป Q4 ลงกราวด์ มอเตอร์จะหมุนในทิศตามเข็มนาฬิกา

$$\text{แรงดันตกคร่อม มอสเฟต Q4 (ขณะไม่มีโหลด)} V_{DSn} = I \cdot R_{ds(on)} = 2A \cdot 0.16\Omega = 0.32V$$

แรงดันตกคร่อม มอสเฟต Q1 (ขณะไม่มีโหลด)

$$V_{DSp} = I \cdot R_{ds(on)} = 2A \cdot 0.3\Omega = 0.6V$$

$$\text{เมื่อ } R_{DS(on)} \text{ p-mos (IRF9530)} = 0.3\Omega$$

$$R_{DS(on)} \text{ n-mos (IRF530)} = 0.16\Omega$$

ดังนั้นจะมีแรงดันตกคร่อมที่มอเตอร์เท่ากับ

$$V_{\text{motor}} = V_s - (V_{DSp} + V_{DSn}) = 24V - (0.32V + 0.6V) = 23.08V$$

โดยเมื่อ Q2, Q3 ทำงาน และ Q1, Q4 ไม่ทำงาน จะทำให้มีกระแสไหลจาก Q2 ผ่าน มอเตอร์ ไป Q3 ลงกราวด์ มอเตอร์จะหมุนในทิศทวนเข็มนาฬิกา

จะมีแรงดันตกคร่อม มอสเฟต Q3 (ขณะไม่มีโหลด) เท่ากับ  $V = I \cdot R_{ds(on)} = 2A \cdot 0.16\Omega = 0.32V$

จะมีแรงดันตกคร่อม มอสเฟต Q2 (ขณะไม่มีโหลด) เท่ากับ  $V = I \cdot R_{ds(on)} = 2A \cdot 0.3\Omega = 0.6V$

ดังนั้นจะมีแรงดันตกคร่อมที่มอเตอร์เท่ากับ

$$V_{\text{motor}} = V_s - (V_{DSp} + V_{DSn}) = 24V - (0.32V + 0.6V) = 23.08V$$

ไดโอดจะทำหน้าที่ป้องกันการเสียหายของมอสเฟตจากสนามไฟฟ้าย้อนกลับ (back EMF) จากมอเตอร์ ชั่วระว่างคือ Q1 และ Q3 (Q2 และ Q4) จะต้องไม่ทำงานพร้อมกัน เพื่อป้องกันการมอสเฟตเสียหาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมทิศทางการหมุนของมอเตอร์จะควบคุมโดยใช้ ไมโครคอนโทรลเลอร์ โดยมีการต่อ Opto-isolate เพื่อแยกแหว่งภาคขับมอเตอร์กับภาคของ ไมโครคอนโทรลเลอร์เพื่อป้องกันการรบกวนจากมอเตอร์กระแสตรง ไมโครคอนโทรลเลอร์จะรับข้อมูลจากคอมพิวเตอร์ซึ่งเขียนโปรแกรมควบคุมการหมุนของมอเตอร์ผ่านพอร์ตอนุกรม

Key control	ทิศทางการเคลื่อนที่
8	เดินหน้า
4	เลี้ยวซ้าย
6	เลี้ยวขวา
2	ถอยหลัง

ตารางที่ 3.1 แสดงค่าทิศทางการเคลื่อนที่ของหุ่นยนต์

### 3.4 โครงสร้างการทำงานของหุ่นยนต์

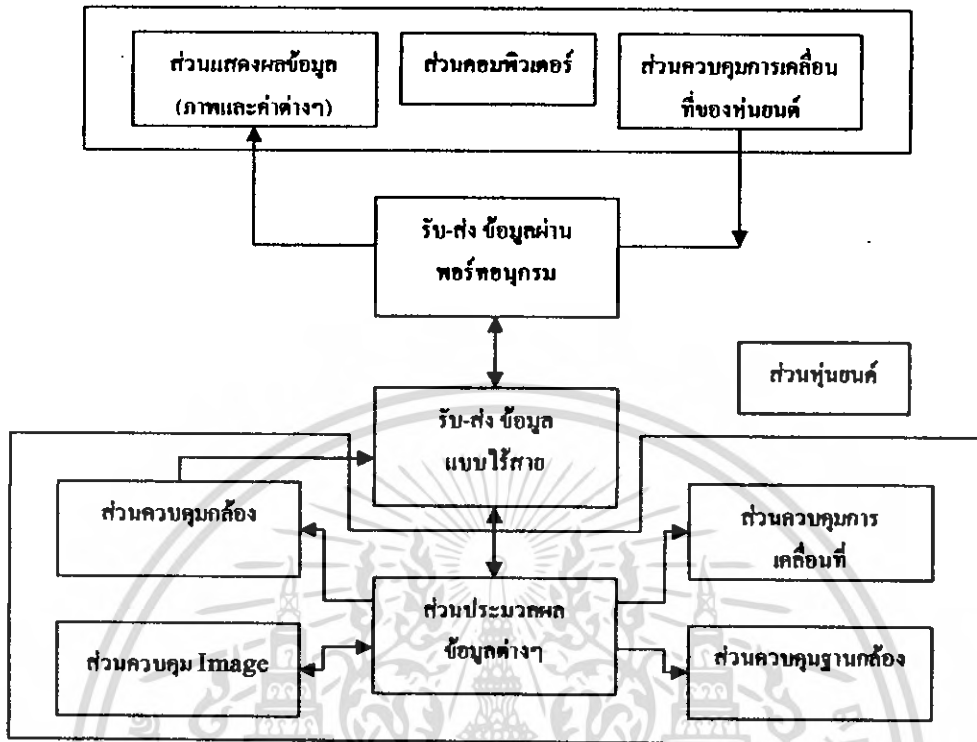
โครงสร้างการทำงานหลักๆ ของหุ่นยนต์ สามารถแบ่งได้ 3 ส่วนใหญ่ๆ ดังรูปที่ 3.1



รูปที่ 3.9 แสดงโครงสร้างการทำงานหลักๆ ของหุ่นยนต์

จากรูปที่ 3.9 ในส่วนของคอมพิวเตอร์จะทำหน้าที่เป็นตัวควบคุมการเคลื่อนที่ของหุ่นยนต์ และ ยังรับ-ส่งข้อมูลระหว่างหุ่นยนต์กับคอมพิวเตอร์ โดยจะใช้พอร์ตอนุกรมเป็นส่วนรับ-ส่งข้อมูล ส่วนของการสื่อสารแบบไร้สายก็จะทำหน้าที่เป็นตัวส่งผ่านข้อมูลระหว่างหุ่นยนต์กับคอมพิวเตอร์ โดยจะผลัดกันส่งข้อมูลกันหรือเป็นการส่งแบบฮาร์ฟ ดิวเพล็กซ์ (Half Duplex) ส่วนสุดท้ายคือ หุ่นยนต์ ซึ่งเป็นส่วนที่มีหน้าที่การทำงานหลากหลายมาก ดังนั้นเราจึงอธิบายการทำงานโดยละเอียด ดังรูปที่ 3.10

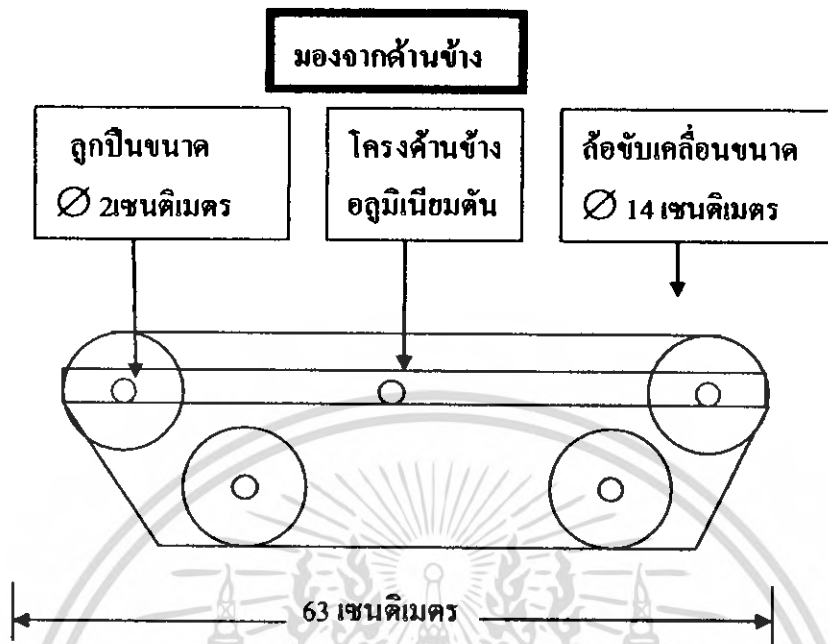
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



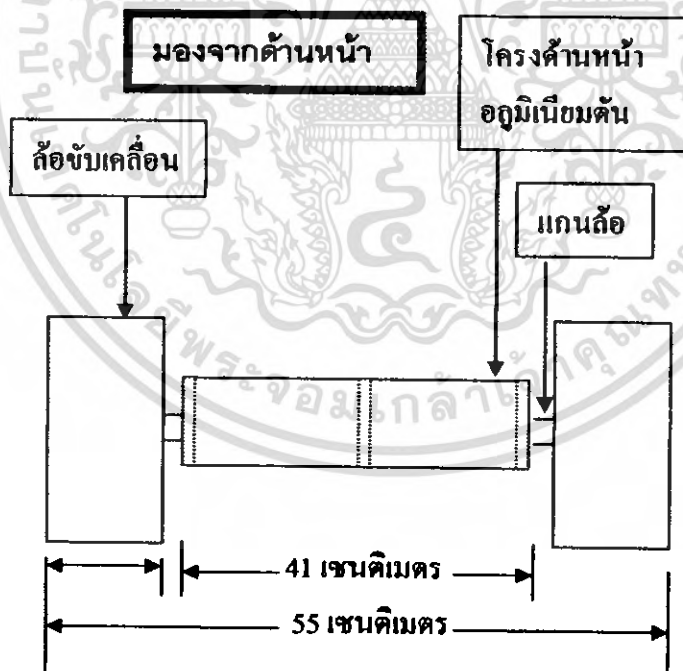
รูปที่ 3.10 แสดงโครงสร้างการทำงานโดยละเอียดของหุ่นยนต์

### 3.4.1 โครงสร้างตัวถังของหุ่นยนต์

โครงสร้างหลักของหุ่นยนต์ทำจากอลูมิเนียมตันหนา 0.6 เซนติเมตรเพื่อความแข็งแรง โดยหุ่นยนต์จะมีขนาดประมาณ 63x50 cm. ใช้ล้อในการขับเคลื่อน เพื่อลดช่วงความห่างทำให้สามารถผ่านพื้นผิวที่เป็นขั้นได้ โดยล้อมีขนาดเส้นผ่านศูนย์กลาง เซนติเมตร กว้าง 7 เซนติเมตร ใช้มอเตอร์ข้างละ 1 ตัวส่งกำลังผ่านโซ่และเฟืองขนาดเล็กขับเคลื่อนทั้งหมด

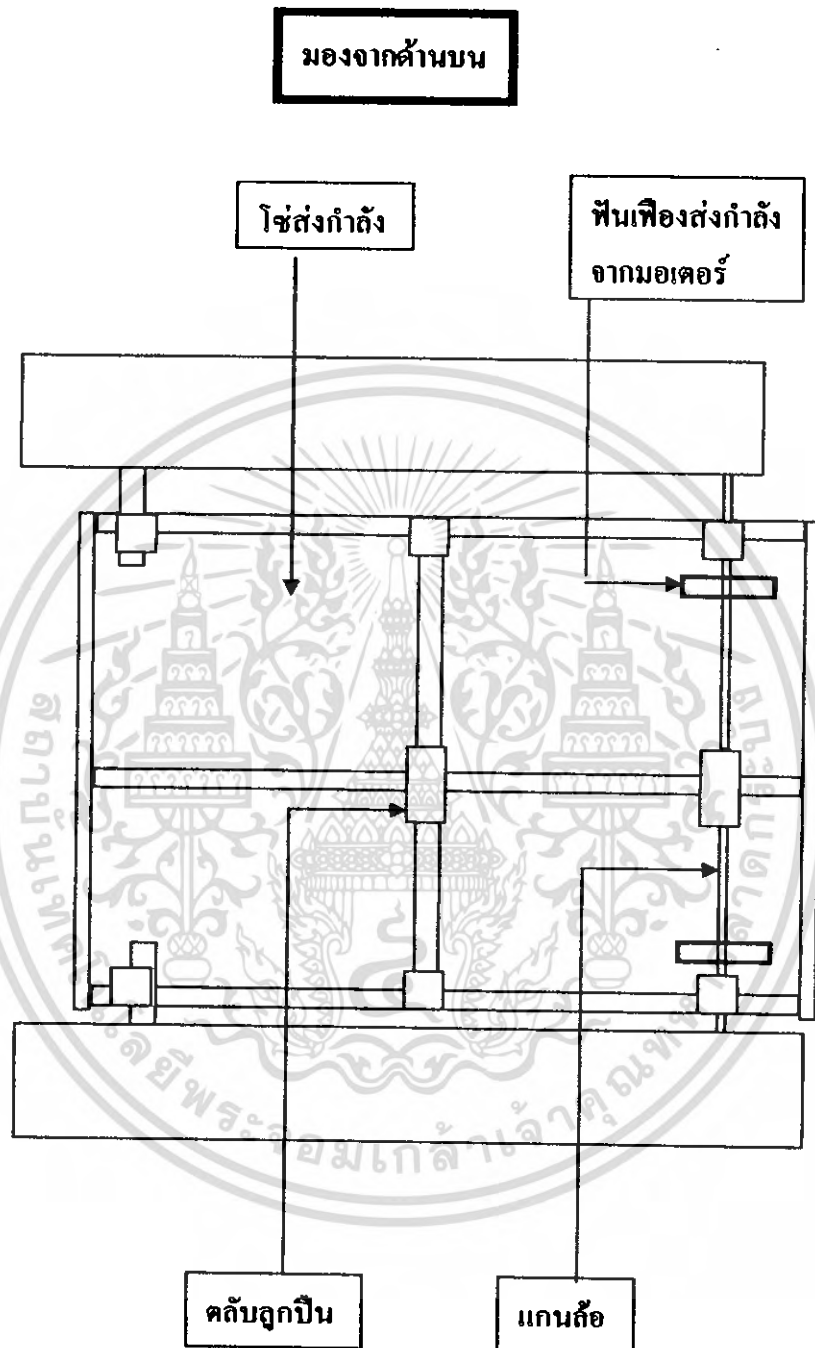


รูปที่ 3.11 แสดงโครงสร้างของหุ่นยนต์มองจากด้านข้าง



รูปที่ 3.12 แสดงโครงสร้างของหุ่นยนต์มองจากด้านหน้า

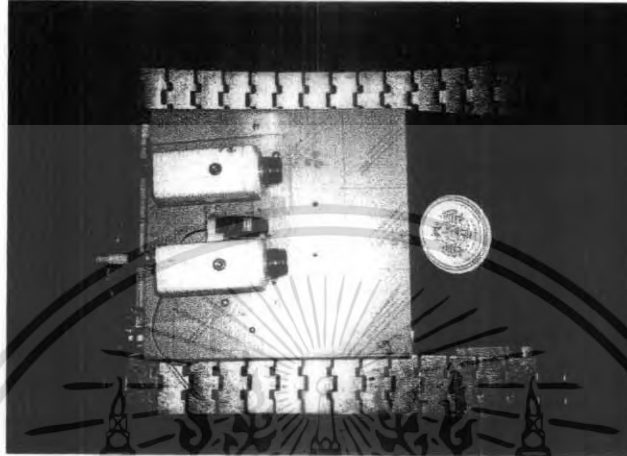
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 แสดงโครงสร้างของหุ่นยนต์มองจากด้านบน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4.2 รูปถ่ายของหุ่นยนต์



รูปที่3.14 รูปถ่ายจากด้านบนของหุ่นยนต์



รูปที่3.15 รูปถ่ายจากด้านหน้าของหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

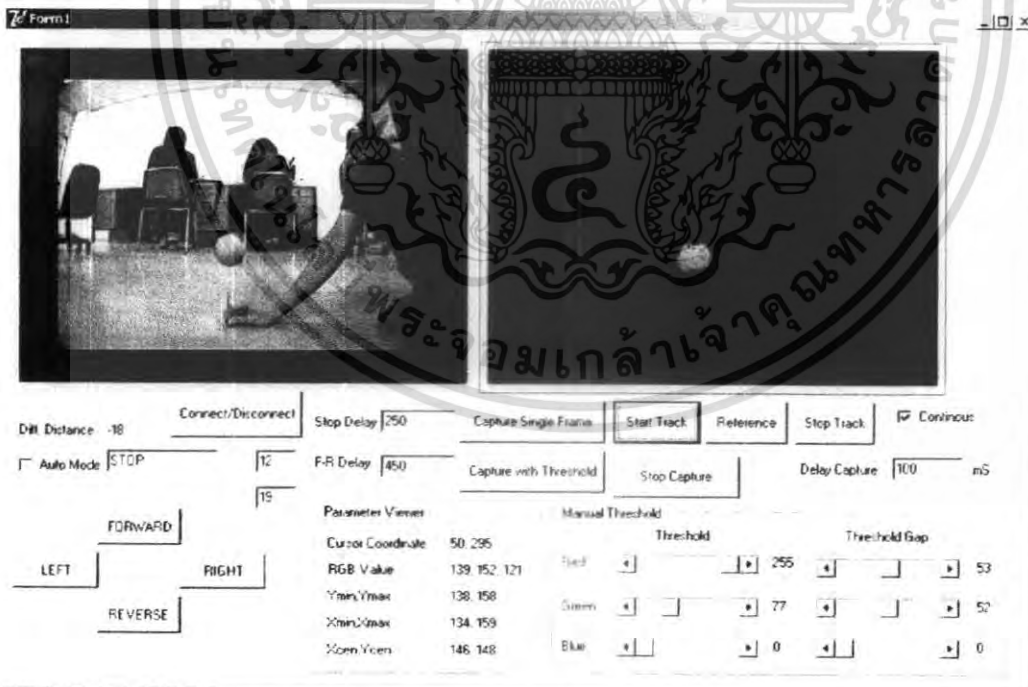
### ผลการทดสอบ

#### 4.1 การทดลองที่ 1. การติดตามวัตถุสี (Object Tracking)

##### 4.1.1 การเคลื่อนที่ของหุ่นยนต์

การบังคับการเคลื่อนที่จะใช้การติดตามวัตถุสีแดงในการกำหนดทิศทาง หรือจะใช้ปุ่มบนหน้าจอกอมพิวเตอร์ที่กำหนดทิศทางการเคลื่อนที่โดยใช้เมาส์คลิกที่ปุ่ม (ปุ่ม FORWARD, REVERSE, LEFT, RIGHT) จากการควบคุมจะพบว่าหุ่นยนต์สามารถเคลื่อนที่ได้ 360 องศารอบทิศทางโดยการหมุนรอบตัวเองอาศัยหลักการหมุนล้อทั้งสองข้างสลับกัน เมื่อจะทำการเลี้ยว ล้อข้างจะเดินหนาและอีกข้างหนึ่งจะถอยหลังเพื่อทำให้การคล่องตัวของตัวหุ่นยนต์ หรือล้อทั้งสองข้างหมุนไม่เท่ากัน จึงจะเลี้ยวได้ และตัวหุ่นยนต์สามารถขึ้นเนินที่มีความลาดชันประมาณ 30 องศาได้

##### 4.1.2 การแสดงภาพบนหน้าจอกอมพิวเตอร์



รูปที่ 4.1 แสดงภาพหน้าต่างของโปรแกรม

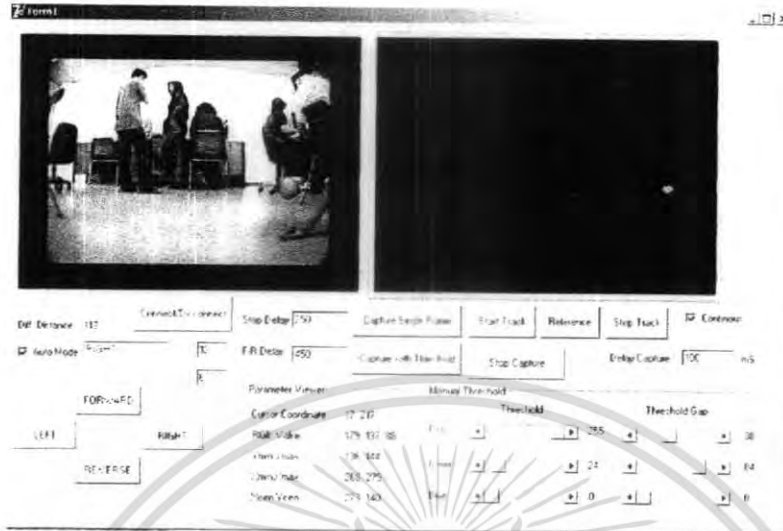
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.2 การเคลื่อนที่ของหุ่นยนต์โดยการติดตามวัตถุ



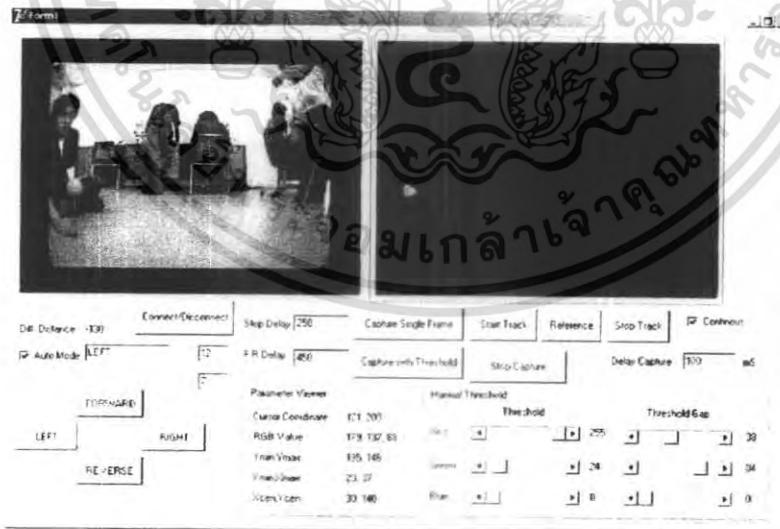
รูปที่ 4.2 การทดสอบการทำจับภาพวัตถุและทำเทรลโฮส

ทำการตรวจสอบวัตถุทรงกลม โดยการจับภาพและทำการเทรลโฮสให้ได้สีแดงซึ่งเป็นสีของวัตถุ และไม่มีสีของสิ่งแวดล้อมเข้ามารบกวน โดยจุดนี้จะทำเป็นจุดอ้างอิงในการบอกระยะที่หุ่นยนต์หยุด การทดลองใช้ระยะที่ 60 ซม. เมื่อหุ่นยนต์เข้าใกล้วัตถุในระยะนี้จะทำให้หุ่นยนต์หยุดการเคลื่อนที่



รูปที่ 4.3 ทดสอบการเคลื่อนที่ ไปทางขวาของหุ่นยนต์

จากนั้นเมื่อทำการเคลื่อนที่วัตถุตรงกลางออกไปทางขวาของหุ่นยนต์จะทำให้หุ่นยนต์เลี้ยวขวาตามวัตถุจนกว่าวัตถุนั้นจะกลับมาอยู่ตรงกลางของภาพอีกครั้ง หุ่นยนต์จึงหยุดการเคลื่อนที่



รูปที่ 4.4 ทดสอบการเคลื่อนที่ ไปทางซ้ายของหุ่นยนต์

จากนั้นทำการเคลื่อนวัตถุทดสอบไปทางซ้ายของหุ่นยนต์ จะทำให้หุ่นยนต์เลี้ยวซ้ายตาม

วัตถุ จนกว่าจะเห็นวัตถุอยู่กลางของภาพ หุ่นยนต์จึงหยุดการเคลื่อนที่

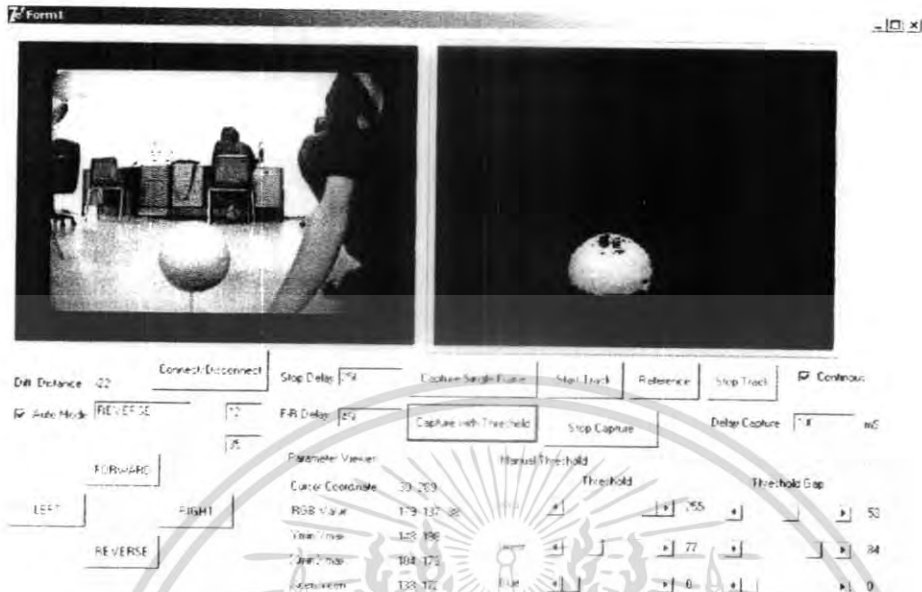
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 ทดสอบการเคลื่อนที่ไปข้างหน้าของหุ่นยนต์

จากนั้นเมื่อทำการนำวัตถุทดสอบเคลื่อนที่ออกห่างจากหุ่นยนต์ จะทำให้วัตถุจะเห็นในภาพมีขนาดเล็กลงจากค่าที่อ้างอิงครั้งแรก จึงทำให้หุ่นยนต์เคลื่อนที่เข้าหาวัตถุทดสอบจนกว่าขนาดของวัตถุจะมีขนาดใกล้เคียงกับขนาดที่อ้างอิงในครั้งแรก หุ่นยนต์จะหยุดการเคลื่อนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 ทดสอบการเคลื่อนที่ถอยหลังของหุ่นยนต์

จากนั้นเมื่อทำการนำวัตถุทดสอบเคลื่อนที่เข้าใกล้หุ่นยนต์ จะทำให้วัตถุจะเห็นในภาพมีขนาดใหญ่ขึ้นจากค่าที่อ้างอิงครั้งแรก จึงทำให้หุ่นยนต์เคลื่อนที่ถอยหลังจากวัตถุทดสอบจนกว่าขนาดของวัตถุจะมีขนาดใกล้เคียงกับขนาดที่อ้างอิงในครั้งแรก หุ่นยนต์จะหยุดการเคลื่อนที่

### 4.3 การทดลองที่ 2. ผลการทดลองวัดระยะโดยทฤษฎีสเตอริโอ สโคปิก (Stereoscopic)

จากสมการของสเตอริโอ

$$Z = \frac{f B}{(x_1 - x_2)}$$

เพราะฉะนั้นจะได้ว่า

$$f = \frac{Z(x_1 - x_2)}{B}$$

ตารางที่ 4.1 แสดงผลการทดลองการหาค่า  $f$

Z	B	$x_1$	$x_2$	f
60 cm	10 cm	201	62	834
90 cm	10 cm	187	96	819
120 cm	10 cm	177	106	852
150 cm	10 cm	166	107	885
180 cm	10 cm	155	109	828
210 cm	10 cm	155	114	861
240 cm	10 cm	154	118	864
300 cm	10 cm	151	122	870
				$f_{AV} = 725.75$

ตารางที่ 4.2 ผลการทดลองการหาระยะทาง (Z) ของกล้องสเตอริโอ

$f_{AV} = 725.75$

B	$x_1$	$x_2$	ระยะทางจริง	Z	% Error
10 cm	201	62	60 cm	59 cm	1.66 %
10 cm	187	96	90 cm	90 cm	0 %
10 cm	177	106	120 cm	117 cm	2.5 %
10 cm	166	107	150 cm	148 cm	1.33 %
10 cm	155	109	180 cm	175 cm	2.77 %
10 cm	155	114	210 cm	200 cm	4.76 %
10 cm	154	118	240 cm	237 cm	1.25 %
10 cm	151	122	300 cm	295 cm	1.67 %

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่หรือนำไปใช้  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.3.1 ผลการทดลองการวัดระยะทฤษฎี สเตอริโอ สโคปิก (Stereoscopic)

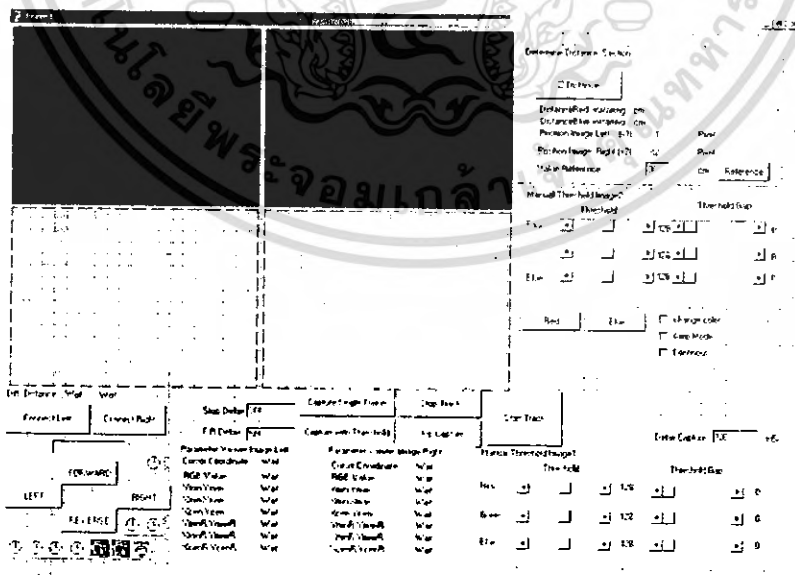
จากผลการทดลองสังเกตได้ว่าการผิดพลาดประมาณ 0 ถึง 10 เปอร์เซ็นต์ โดยวัตถุซึ่งอยู่ระยะไกลค่าความผิดพลาดยิ่งสูงขึ้น ความผิดพลาดที่เกิดขึ้นเกิดเนื่องจากการกั้นระดับสีที่อาจทำให้วัตถุที่ต้องการมีขนาดผิดเพี้ยนไปจากเดิมทำให้จุดกึ่งกลางของวัตถุในแต่ละภาพมีค่าแตกต่างจากจุดเดิมเล็กน้อยได้

สรุปการทดลองนี้แม้จะมีความผิดพลาดอยู่บ้างแต่ถือว่ายังยอมรับได้ ทำให้การหาระยะวัตถุมีความแม่นยำขึ้นกว่าการตรวจสอบขนาดเล็กหรือใหญ่ของวัตถุ

คุณสมบัติที่ต้องการเพื่อให้เกิดความแม่นยำที่มากขึ้น

- จุดอ้างอิงตำแหน่งของวัตถุทั้งสองภาพต้องเป็นจุดอ้างอิงตำแหน่งเดียวกันถึงจะหาระยะได้อย่างถูกต้อง
- กล้องและเลนส์ที่ใช้ทั้งสองต้องมีคุณสมบัติเหมือนกันทุกประการ
- ระยะห่างของกล้องทั้งสองตัวควรอยู่ใกล้กันเพื่อให้สามารถมองวัตถุในช่วงที่เอียงเล็กน้อย

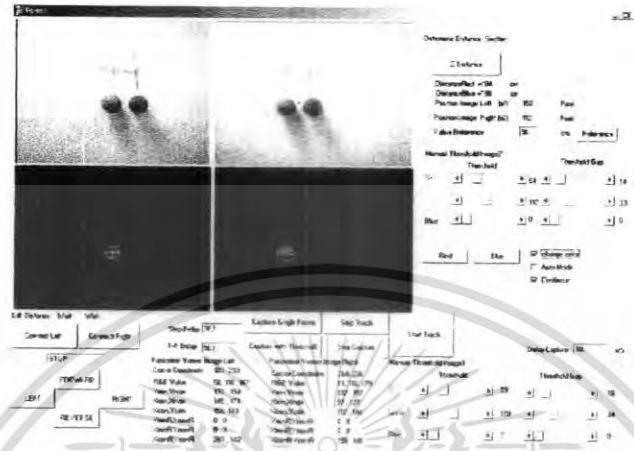
#### 4.4 รูปแบบของหน้าต่างโปรแกรมการเกิดภาพสเตอริโอสโคปิก



รูปที่ 4.7 หน้าต่างโปรแกรมการเกิดภาพสเตอริโอสโคปิก

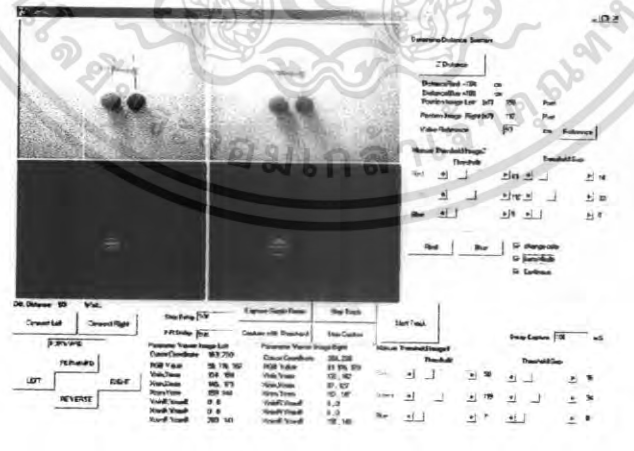
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.5 ผลการทดลองการเคลื่อนที่ของหุ่นยนต์ด้วยทฤษฎีสเตอริโอโคปิก (Stereoscopic)



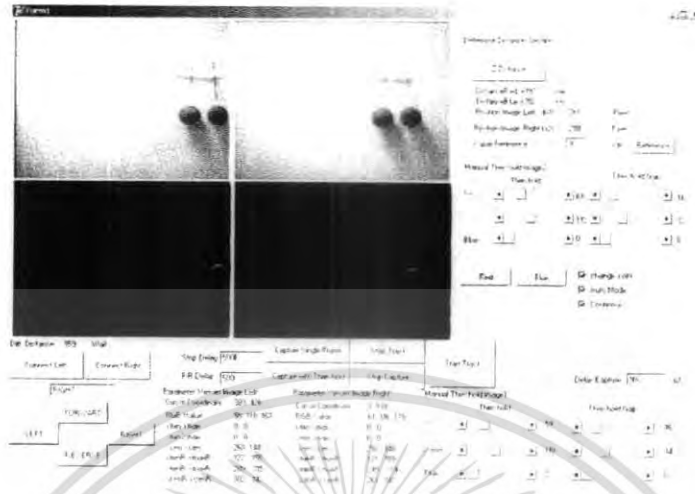
รูปที่ 4.8 การทำเทรสเตอร์ของวัตถุสีในโปรแกรมสเตอริโอโคปิก

ทำการตรวจสอบวัตถุสีทรงกลม โดยการจับภาพและทำการเทรสเตอร์ให้ได้สีแดง ซึ่งเป็นสีของวัตถุและวัตถุสีน้ำเงิน และไม่มีสีของสิ่งแวดล้อมเข้ามารบกวน และทำการบันทึกค่าที่ได้ทั้งสองสี พร้อมกับกำหนดระยะอ้างอิงเป็นหน่วยเซนติเมตร ซึ่งจะเป็นระยะที่จะกำหนดการหยุดของหุ่นยนต์เมื่อหุ่นยนต์เข้าใกล้วัตถุ



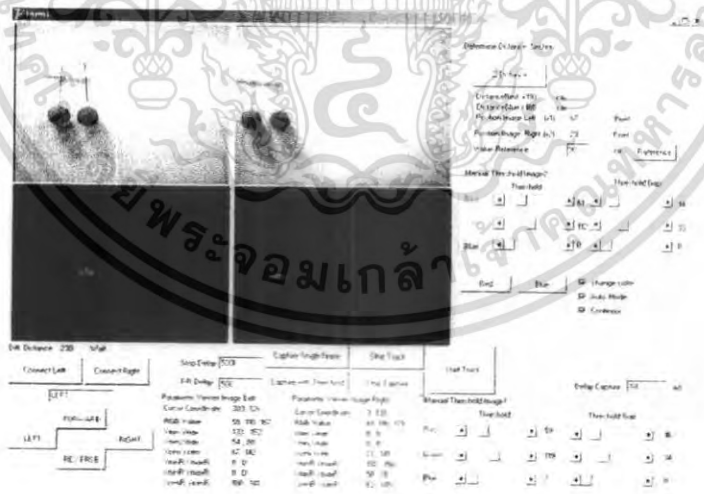
รูปที่ 4.9 ตำแหน่งของวัตถุที่ทำให้หุ่นยนต์เคลื่อนที่ไปข้างหน้า

จากนั้นเมื่อทำการนำวัตถุสีทดสอบเคลื่อนที่ออกจากหุ่นยนต์จะทำให้ระยะห่างระหว่างหุ่นยนต์กับวัตถุห่างมากขึ้นรถก็จะวิ่งตามจนถึงระยะที่ทำการอ้างอิง เมื่อถึงระยะอ้างอิงหุ่นยนต์ก็จะหยุดเดิน เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้ดูแลเห็นชอบจะออกผลการคำนวณว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 ตำแหน่งของวัตถุที่ทำให้หุ่นยนต์เคลื่อนที่ไปทางด้านขวา

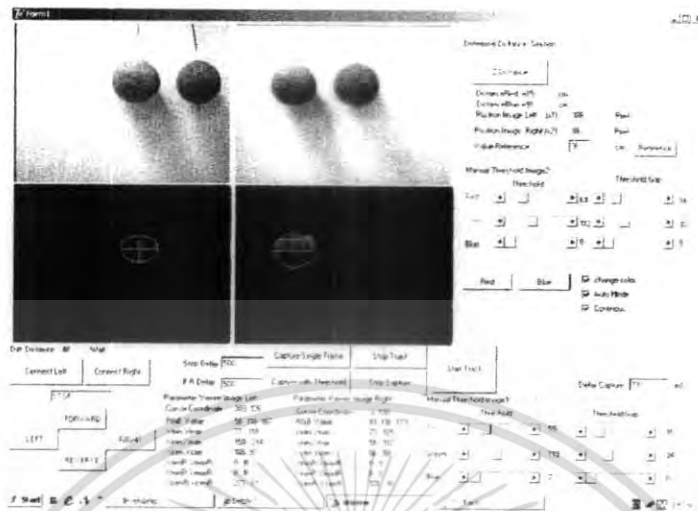
จากนั้นทำการเลื่อนวัตถุทดสอบไปทางขวาของหุ่นยนต์ จะทำให้หุ่นยนต์เลี้ยวขวาตามวัตถุ จนกว่าจะเห็นวัตถุอยู่กลางของจอภาพ หุ่นยนต์จึงหยุดการเคลื่อนที่



รูปที่ 4.11 ตำแหน่งของวัตถุที่ทำให้หุ่นยนต์เคลื่อนที่ไปทางด้านซ้าย

จากนั้นทำการเลื่อนวัตถุทดสอบไปทางซ้ายของหุ่นยนต์ จะทำให้หุ่นยนต์เลี้ยวซ้ายตามวัตถุ จนกว่าจะเห็นวัตถุอยู่กลางของจอภาพ หุ่นยนต์จึงหยุดการเคลื่อนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.12 ตำแหน่งของวัตถุที่ทำให้หุ่นยนต์หยุดการเคลื่อนที่

เมื่อหุ่นยนต์มาถึงในระยะที่กำหนดจากค่าที่อ้างอิงไว้จะทำให้หุ่นยนต์หยุดการเคลื่อนที่  
จนกว่าจะมีการเคลื่อนที่ของวัตถุหุ่นยนต์จึงจะมีการเคลื่อนที่ตาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### สรุปผลการทดลองและแนวทางการพัฒนาต่อ

#### 5.1 สรุปผลการทดลองที่ 1 การติดตามวัตถุสี (Object Tracking)

จากการทดลองโครงการนี้สามารถควบคุมหุ่นยนต์ได้ 2 แบบ คือแบบควบคุมโดยคน และแบบควบคุมตามวัตถุอัตโนมัติ โดยอาศัยการตรวจจับสีของวัตถุ

จากการทดลองแบบควบคุมโดยคนสามารถควบคุมผ่านทางโปรแกรมโดยตรงได้เลข โดยควบคุมผ่านพอร์ตอนุกรม สามารถเคลื่อนที่ได้ทุกสภาพพื้นผิว และพื้นที่ลาดเอียงได้ไม่เกิน 60 องศา สิ่งขีดขวางสูงไม่เกิน 10 ซม.

จากการทดลองแบบควบคุมตามวัตถุอัตโนมัติ สามารถติดตามได้วัตถุ และเลี้ยวตามได้ โดยการอาศัยการตรวจจับสีของวัตถุ เมื่อทำการจับภาพวัตถุได้แล้วนำภาพมาทำการแยกองค์ประกอบสีเพื่อให้เหลือแต่สีของวัตถุสีเดียว แต่เนื่องจากสภาพแวดล้อมมีองค์ประกอบของ 3 สี ประกอบกันจึงทำให้การแยกวัตถุออกจากสิ่งแวดล้อมทำได้ยาก จึงต้องอาศัยการแยกสีด้วยมือ จึงทำให้แยกวัตถุออกจากสิ่งแวดล้อมได้ โดยเมื่อสีใดที่เป็นสีของวัตถุก็จะให้เป็นแสดงค่าสีเป็น 255 หรือเป็นสีของวัตถุ ที่ชัดที่สุด แล้วสีใดมีองค์ประกอบของสีน้อยสีของวัตถุที่ทดสอบ ก็ให้แสดงค่าสีเป็น 0 หรือดำ ทำให้เหลือแค่วัตถุเท่านั้นที่อยู่บนจอภาพ และเมื่อยังมีสีจากสิ่งแวดล้อมเข้ามาอีกทำการปรับเทรสโอดของแต่ละสีจนให้เหลือแต่สีของวัตถุที่ทดสอบและสีของสิ่งแวดล้อมให้หายไป จากนั้นทำการสร้าง เส้นรอบวัตถุเพื่อเป็นวิเคราะห์หาจุดศูนย์กลางและขนาดของวัตถุ เพื่อทำเป็นจุดและขนาดอ้างอิงในการตรวจสอบว่าวัตถุเคลื่อนที่ออกห่าง เข้าใกล้ อยู่ทางซ้าย หรืออยู่ทางขวาของหุ่นยนต์ โดยเทียบจากจุดกึ่งกลางของภาพนั้น เพื่อให้หุ่นยนต์เคลื่อนที่ตามวัตถุได้อย่างอัตโนมัติ และถ้าวัตถุหลุดออกจากภาพก็จะบังคับให้หุ่นยนต์หยุดเองแบบอัตโนมัติ แต่เนื่องจากสีจากสิ่งแวดล้อมอาจจะมีมารบกวนบางจึงทำให้การเคลื่อนที่ของหุ่นยนต์มีความผิดพลาดได้ จึงจำเป็นที่จะต้องใช้สีของวัตถุที่ตรวจสอบเป็นสีที่แตกต่างจากสิ่งแวดล้อมมากๆ เพื่อป้องกันการเกิดสัญญาณรบกวนที่จะทำให้หุ่นยนต์เคลื่อนที่ผิดพลาด และจากการทดลองการเคลื่อนที่ของหุ่นยนต์สามารถเคลื่อนที่ตามได้ทุกสีและทุกขนาดของวัตถุที่ตรวจสอบ แต่จะไม่สามารถบอกถึงระยะห่างระหว่างวัตถุกับหุ่นยนต์ได้ ซึ่งเป็นขีดจำกัดของการใช้กล้องตัวเดียว ซึ่งจะใช้กล้องสองในเทอมถัดไป ซึ่งสามารถทราบได้ถึงพิคัด 3 มิติได้

## 5.2 การทดลองที่ 2. ผลการทดลองวัดระยะโดยทฤษฎีสเตอร์ิโอสโคปิก (Stereoscopic)

1. ภาพที่เกิดจากกล้องทั้งสองนั้นจะมีขนาดไม่เท่ากันเนื่องจากเลนส์ของกล้องทั้งสองนั้นมีระยะโฟกัสภาพไม่เท่ากัน
2. สีที่เกิดจากกล้องทั้งสองนั้นไม่เหมือนกัน
3. การให้วัตถุอยู่บนกล้องทั้งสองตัว จะต้องวางวัตถุห่างจากกล้อง
4. การหาจุด ตำแหน่ง  $x_1$  และ  $x_2$  บนระนาบของภาพที่เกิดจากกล้อง เพื่อมาหาระยะทาง จะต้องเป็นจุดบนวัตถุเดียวกันเพื่อให้ได้ระยะทางที่ถูกต้องที่สุด แต่การทำงานของโปรแกรมจะใช้จุดที่ใกล้เคียงกันจึงได้ระยะทางที่อาจเกิดความผิดพลาดประมาณ  $\pm 5\%$
5. มุมมองของกล้องจะมีขนาดแคบลง

## 5.3 การแก้ปัญหาของการวัดระยะโดยทฤษฎีสเตอร์ิโอสโคปิก (Stereoscopic)

1. ควรใช้กล้องรุ่นเดียวกันและเลนส์ที่มีระยะโฟกัสขนาดเท่ากันด้วยและวางตำแหน่งกล้องให้ขนานกันในระนาบเดียวกันจะได้ขนาดของภาพและสีเท่ากัน
2. การวางตำแหน่งระยะห่างระหว่างกล้องทั้งสองให้ได้ขนาดใกล้กันมากที่สุด โดยการทดลองได้ใช้ระยะห่างระหว่างกล้องทั้งสองที่ 10 เซนติเมตร จึงทำให้การมองเห็นตำแหน่งจุด  $x_1$  และ  $x_2$  ของวัตถุทั้งสองภาพนั้นเป็นจุดเดียวกัน โปรแกรมนั้นก็จะสามารถคำนวณระยะทางระหว่างวัตถุกับตัวหุ่นยนต์ได้ถูกต้อง

## เอกสารอ้างอิง

- [1] นคร ภักดีชาติ, ชัยวัฒน์ ลิ้มพรจิตรวิไล, ไมโครคอนโทรลเลอร์ MSC51 ด้วยโปรแกรมภาษา C ฉบับ P89V51RD2 กรุงเทพฯ : อินโนเวทีฟ เอ็กเพอริเม้นต์ จำกัด, 2524.
- [2] ชูชาติ ปิณฑวิรุจน์, รศ.ดร.การประมวลผลภาพเชิงเลข Digital Image Processing เล่ม 1. กรุงเทพฯ : สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ก.

unit stereool;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, Buttons, CPDrv, ExtCtrls, tscap32\_rt, StdCtrls, Menus;

type

TForm1 = class(TForm)

Comm1: TCommPortDriver;

SpeedButton1: TSpeedButton;

SpeedButton2: TSpeedButton;

SpeedButton3: TSpeedButton;

SpeedButton4: TSpeedButton;

Timer1: TTimer;

tscap321: TtsCap32;

Button1: TButton;

Image1: TImage;

Button2: TButton;

tscap32PopupMenu1: TtsCap32PopupMenu;

Timer2: TTimer;

CheckBox1: TCheckBox;

Edit1: TEdit;

Label1: TLabel;

Label2: TLabel;

Button3: TButton;

GroupBox1: TGroupBox;

ScrollBar1: TScrollBar;

ScrollBar2: TScrollBar;

Label3: TLabel;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Label4: TLabel;  
Label5: TLabel;  
Label6: TLabel;  
ScrollBar3: TScrollBar;  
ScrollBar4: TScrollBar;  
ScrollBar5: TScrollBar;  
ScrollBar6: TScrollBar;  
Label7: TLabel;  
Label8: TLabel;  
Label9: TLabel;  
Label10: TLabel;  
Label11: TLabel;  
Button4: TButton;  
Timer3: TTimer;  
GroupBox2: TGroupBox;  
Label16: TLabel;  
Label17: TLabel;  
Label18: TLabel;  
Label19: TLabel;  
Label12: TLabel;  
Label14: TLabel;  
Label13: TLabel;  
Label15: TLabel;  
Label22: TLabel;  
Label23: TLabel;  
Label20: TLabel;  
Label21: TLabel;  
Button5: TButton;  
Button6: TButton;  
CheckBox2: TCheckBox;



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Label24: TLabel;

Edit2: TEdit;

Label25: TLabel;

Timer4: TTimer;

Edit3: TEdit;

Label26: TLabel;

Edit4: TEdit;

Label27: TLabel;

Button7: TButton;

Image2: TImage;

tsCap32PopupMenu2: TtsCap32PopupMenu;

tsCap322: TtsCap32;

Label30: TLabel;

Label31: TLabel;

Button8: TButton;

Label35: TLabel;

Label32: TLabel;

Label42: TLabel;

Label43: TLabel;

Label44: TLabel;

Label45: TLabel;

Label46: TLabel;

Label47: TLabel;

Button9: TButton;

Label28: TLabel;

GroupBox3: TGroupBox;

Label33: TLabel;

GroupBox4: TGroupBox;

Label34: TLabel;

Label36: TLabel;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Label37: TLabel;  
Label38: TLabel;  
Label39: TLabel;  
Edit7: TEdit;  
Label40: TLabel;  
Label41: TLabel;  
GroupBox5: TGroupBox;  
ScrollBar7: TScrollBar;  
ScrollBar8: TScrollBar;  
ScrollBar9: TScrollBar;  
ScrollBar10: TScrollBar;  
ScrollBar11: TScrollBar;  
ScrollBar12: TScrollBar;  
Label57: TLabel;  
Label58: TLabel;  
Label59: TLabel;  
Label60: TLabel;  
Label61: TLabel;  
Label62: TLabel;  
Label63: TLabel;  
Label64: TLabel;  
Label65: TLabel;  
Label66: TLabel;  
Label68: TLabel;  
Label69: TLabel;  
Label70: TLabel;  
Button10: TButton;  
Button11: TButton;  
Label71: TLabel;  
Label72: TLabel;



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Label73: TLabel;  
Label49: TLabel;  
Label50: TLabel;  
Label51: TLabel;  
Label52: TLabel;  
Label53: TLabel;  
Label54: TLabel;  
Label55: TLabel;  
Label56: TLabel;  
Label74: TLabel;  
Label75: TLabel;  
Label76: TLabel;  
Label77: TLabel;  
Label78: TLabel;  
Timer5: TTimer;  
Timer6: TTimer;  
GroupBox6: TGroupBox;  
RadioButton1: TRadioButton;  
RadioButton2: TRadioButton;  
Edit5: TEdit;  
Label29: TLabel;  
GroupBox7: TGroupBox;  
SpeedButton5: TSpeedButton;  
SpeedButton6: TSpeedButton;  
procedure SpeedButton5Up(Sender: TObject; Button: TMouseButton;  
Shift: TShiftState; X, Y: Integer);  
procedure SpeedButton1Up(Sender: TObject; Button: TMouseButton;  
Shift: TShiftState; X, Y: Integer);  
procedure SpeedButton2Down(Sender: TObject; Button: TMouseButton;  
Shift: TShiftState; X, Y: Integer);



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure SpeedButton2Up(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure SpeedButton4Down(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure SpeedButton4UP(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure SpeedButton3Down(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure SpeedButton3Up(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure Timer1Timer(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Timer2Timer(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure ScrollBar1Change(Sender: TObject);
procedure ScrollBar2Change(Sender: TObject);
procedure ScrollBar3Change(Sender: TObject);
procedure ScrollBar4Change(Sender: TObject);
procedure ScrollBar5Change(Sender: TObject);
procedure ScrollBar6Change(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure Image1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure Image1MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure Timer3Timer(Sender: TObject);
procedure Button5Click(Sender: TObject);
procedure Button6Click(Sender: TObject);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure Timer4Timer(Sender: TObject);
procedure Button7Click(Sender: TObject);
procedure Image2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure Image2MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure Button8Click(Sender: TObject);
procedure Button9Click(Sender: TObject);
procedure ScrollBar7Change(Sender: TObject);
procedure ScrollBar8Change(Sender: TObject);
procedure ScrollBar9Change(Sender: TObject);
procedure ScrollBar10Change(Sender: TObject);
procedure ScrollBar11Change(Sender: TObject);
procedure ScrollBar12Change(Sender: TObject);
procedure Button10Click(Sender: TObject);
procedure Button11Click(Sender: TObject);
procedure Button12Click(Sender: TObject);
procedure Timer5Timer(Sender: TObject);
procedure Timer7Timer(Sender: TObject);
procedure Timer8Timer(Sender: TObject);
procedure Timer6Timer(Sender: TObject);
procedure SpeedButton6Up(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure SpeedButton6Down(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure SpeedButton5Down(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure SpeedButton1Down(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

private
    procedure Thresholding: ! Private declarations !
public
    { Public declarations }
end;
var
    Form1: TForm1;
implementation
    {$R *.dfm}
type
    TRGB=array[0..1023] of TRGBTriple;
    PRGB=^TRGB;
    TCube = array[1..8] of Integer;
const my_forward=8;
    my_reverse=2;
    my_left=6;
    my_right=4;
    my_stop=0;
    cam_right=1;
    cam_left=3;
    cam_control=9;
var
    direction,camdirection:byte;
    ThBMP:TBitmap;
    ThBMP1:TBitmap;
    LineShow:boolean;
    RefRed,RefGreen,RefBlue:byte;
    Ymin,Ymax,Xmin,Xmax,Ymin1,Ymax1,Xmin1,Xmax1:word;
    YYmin,YYmax,XXmin,XXmax,YYmin1,YYmax1,XXmin1,XXmax1:word;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Manual.change.change1.LR:boolean;  
 Xcen.Ycen,Xcen1.Ycen1,XXcen.YYcen,XXcen1.YYcen1:word;  
 centerLEFT,centerRIGHT:Integer;  
 ZZ,XX1:Integer;  
 refL.refLEFT.refRIGHT,result,XX:Integer;  
 Xleft:Integer;  
 Yleft:Integer;  
 Xright:Integer;  
 Yright:Integer;  
 a:Integer;  
 b:Integer;  
 ar:TCube;  
 br:TCube;  
 f:integer;  
 d:integer;  
 distance:integer;  
 x1:integer;  
 x2:integer;  
 x3,x4,x5,x6:integer;  
 zref,distanceblue:integer;  
 Sredmax,Sblucmax,SGreenmax:integer;  
 Sredmin,Sbluemin,SGreenmin:integer;  
 Sredmax1,Sbluemax1,SGreenmax1:integer;  
 Sredmin1,Sbluemin1,SGreenmin1:integer;  
 SSredmax,SSblucmax,SSGreenmax:integer;  
 SSredmin,SSbluemin,SSGreenmin:integer;  
 SSredmax1,SSbluemax1,SSGreenmax1:integer;  
 SSredmin1,SSbluemin1,SSGreenmin1:integer;



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
RedChk_Min.RedChk_Max.GreenChk_Min.GreenChk_Max.BlueChk_Min.BlueChk_Max:small
int;
```

```
RedChk_Min1.RedChk_Max1.GreenChk_Min1.GreenChk_Max1.BlueChk_Min1.BlueChk_Max
1:smallint;
```

```
RedChk.GreenChk,BlueChk:boolean;
```

```
RedChk1.GreenChk1,BlueChk1.rfrist:boolean;
```

```
caseturn:byte;
```

```
chkdt:boolean;
```

```
//*****Thresholding*****//
```

```
procedure TForm1.Thresholding; //sub programe
```

```
var
```

```
PThBMP,PThBMP1:PRGB;
```

```
x,y,x1,y1:word;
```

```
First:boolean;
```

```
begin
```

```
if Manual then
```

```
begin
```

```
ThBMP.Free;
```

```
ThBMP1.Free;//Image 2
```

```
Ymin:=0;
```

```
Ymax:=0;
```

```
Xmin:=0;
```

```
Xmax:=0;
```

```
Ymin1:=0; // Image 2
```

```
Ymax1:=0;
```

```
Xmin1:=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Xmax1:=0;
ThBMP:=TBitmap.Create;
ThBMP1:=TBitmap.Create; //Image 2
ThBMP.LoadFromFile('Test.BMP');
ThBMP1.LoadFromFile('Test1.BMP'); // Image 2
RedChk_Min:=ScrollBar1.Position-ScrollBar4.Position;
RedChk_Max:=ScrollBar1.Position+ScrollBar4.Position;
GreenChk_Min:=ScrollBar2.Position-ScrollBar5.Position;
GreenChk_Max:=ScrollBar2.Position+ScrollBar5.Position;
BlueChk_Min:=ScrollBar3.Position-ScrollBar6.Position;
BlueChk_Max:=ScrollBar3.Position+ScrollBar6.Position;
if RedChk_Min<0 then RedChk_Min :=0;
if RedChk_Max>255 then RedChk_Max :=255;
if GreenChk_Min<0 then GreenChk_Min :=0;
if GreenChk_Max>255 then GreenChk_Max :=255;
if BlueChk_Min<0 then BlueChk_Min :=0;
if BlueChk_Max>255 then BlueChk_Max :=255;
First:=true;
for y:=0 to ThBMP.Height-1 do
begin
  PThBMP:=ThBMP.ScanLine[y];
  for x:=0 to ThBMP.Width-1 do
  begin
    if ScrollBar4.Position=0 then
      RedChk:=PThBMP[x].rgbRed>=RedChk_Min
    else
      RedChk:=(PThBMP[x].rgbRed>=RedChk_Min) and
(PThBMP[x].rgbRed<=RedChk_Max);
    if ScrollBar5.Position=0 then
      GreenChk:=PThBMP[x].rgbGreen>=GreenChk_Min

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else
  GreenChk:=(PThBMP[x].rgbtGreen>=GreenChk_Min) and
(PThBMP[x].rgbtGreen<=GreenChk_Max);
  if ScrollBar6.Position=0 then
    BlueChk:=PThBMP[x].rgbtBlue>=BlueChk_Min
  else
    BlueChk:=(PThBMP[x].rgbtBlue>=BlueChk_Min) and
(PThBMP[x].rgbtBlue<=BlueChk_Max);
  if RedChk and GreenChk and BlueChk then
begin
  if First then
begin
  Ymin:=y;
  Xmin:=x;
  First:=false;
end
else
begin
  if y<Ymin then Ymin:=y;
  if y>Ymax then Ymax:=y;
  if x<Xmin then Xmin:=x;
  if x>Xmax then Xmax:=x;
end;
end
else
begin
  PThBMP[x].rgbtRed:=0;
  PThBMP[x].rgbtGreen:=0;
  PThBMP[x].rgbtBlue:=0;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

end;

end:

//*****Image 2*****//

RedChk_Min1:=ScrollBar7.Position-ScrollBar10.Position;
RedChk_Max1:=ScrollBar7.Position+ScrollBar10.Position;
GreenChk_Min1:=ScrollBar8.Position-ScrollBar11.Position;
GreenChk_Max1:=ScrollBar8.Position+ScrollBar11.Position;
BlueChk_Min1:=ScrollBar9.Position-ScrollBar12.Position;
BlueChk_Max1:=ScrollBar9.Position+ScrollBar12.Position;
if RedChk_Min1<0 then RedChk_Min1 :=0;
if RedChk_Max1>255 then RedChk_Max1 :=255;
if GreenChk_Min1<0 then GreenChk_Min1 :=0;
if GreenChk_Max1>255 then GreenChk_Max1 :=255;
if BlueChk_Min1<0 then BlueChk_Min1 :=0;
if BlueChk_Max1>255 then BlueChk_Max1 :=255;
First:=true;
for y1:=0 to ThBMP1.Height-1 do
begin
PThBMP1:=ThBMP1.ScanLine[y1];
for x1:=0 to ThBMP1.Width-1 do
begin
if ScrollBar10.Position=0 then
RedChk1:=PThBMP1[x1].rgbRed>=RedChk_Min1
else
RedChk1:=(PThBMP1[x1].rgbRed>=RedChk_Min1) and
(PThBMP1[x1].rgbRed<=RedChk_Max1);
if ScrollBar11.Position=0 then
GreenChk1:=PThBMP1[x1].rgbGreen>=GreenChk_Min1
else

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
GreenChk1:=(PThBMP1[x1].rgbtGreen> GreenChk_Min) and  
(PThBMP1[x1].rgbtGreen<=GreenChk_Max1);
```

```
if ScrollBar12.Position=0 then
```

```
BlueChk1:=PThBMP1[x1].rgbtBlue>=BlueChk_Min1
```

```
else
```

```
BlueChk1:=(PThBMP1[x1].rgbtBlue>=BlueChk_Min1) and
```

```
(PThBMP1[x1].rgbtBlue<=BlueChk_Max1);
```

```
if RedChk1 and GreenChk1 and BlueChk1 then
```

```
begin
```

```
if First then
```

```
begin
```

```
Ymin1:=y1;
```

```
Xmin1:=x1;
```

```
First:=false;
```

```
end
```

```
else
```

```
begin
```

```
if y1<Ymin1 then Ymin1:=y1;
```

```
if y1>Ymax1 then Ymax1:=y1;
```

```
if x1<Xmin1 then Xmin1:=x1;
```

```
if x1>Xmax1 then Xmax1:=x1;
```

```
end;
```

```
end
```

```
else
```

```
begin
```

```
PThBMP1[x1].rgbtRed:=0;
```

```
PThBMP1[x1].rgbtGreen:=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        PThBMP1[x1].rgbBlue:=0;
    end;
end;
end:
Image1.Picture.Graphic:=ThBMP;
Image2.Picture.Graphic:=ThBMP1;
Label14.Caption:=IntToStr(Ymin)+'.'+IntToStr(Ymax);
Label15.Caption:=IntToStr(Xmin)+'.'+IntToStr(Xmax);
Label42.Caption:=IntToStr(Ymin1)+'.'+IntToStr(Ymax1);
Label43.Caption:=IntToStr(Xmin1)+'.'+IntToStr(Xmax1);
end;
end:
//*****//
procedure TForm1.SpeedButton5Up(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
begin
    Timer5.Enabled:=false;
    Comm1.SendByte(my_stop);
end;
//*****//
procedure TForm1.SpeedButton1Up(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
begin
    Timer1.Enabled:=false;
    Comm1.SendByte(my_stop);
end;
//*****//
procedure TForm1.SpeedButton2Down(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);

```

**begin**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if Comm1.Connect then
begin
    Comm1.SendByte(my_left);
    direction:=my_left;
    Timer1.Enabled:=true;
end;
end;
//*****//
procedure TForm1.SpeedButton2Up(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
begin
    Timer1.Enabled:=false;
    Comm1.SendByte(my_stop);
end;
//*****//
procedure TForm1.SpeedButton4Down(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
begin
    if Comm1.Connect then
    begin
        Comm1.SendByte(my_reverse);
        direction:=my_reversc;
        Timer1.Enabled:=true;
    end;
end;
//*****//
procedure TForm1.SpeedButton4UP(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
begin
    Timer1.Enabled:=false;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Comm1.SendByte(my_stop);
end;
//*****//
procedure TForm1.SpeedButton3Down(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
begin
if Comm1.Connect then
begin
Comm1.SendByte(my_right);
direction:=my_right;
Timer1.Enabled:=true;
end;
end;
//*****//
procedure TForm1.SpeedButton3Up(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
begin
Timer1.Enabled:=false;
Comm1.SendByte(my_stop);
end;
//*****//
procedure TForm1.Timer1Timer(Sender: TObject);
begin
case direction of
my_forward: Comm1.SendByte(my_forward);
my_reverse: Comm1.SendByte(my_reverse);
my_left : Comm1.SendByte(my_left);
my_right : Comm1.SendByte(my_right);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

end;

end;

//*****//
procedure TForm1.Button1Click(Sender: TObject);
begin
    tscap321.Connected := Not tscap321.Connected;

end;

//*****//
procedure TForm1.Button2Click(Sender: TObject);
begin
    Timer3.Enabled:=true;
end;

//*****//
procedure TForm1.Timer2Timer(Sender: TObject); //capture picture
begin
    if CheckBox1.Checked=false then
        begin
            Timer2.Enabled:=false;
        end
    else
        begin
            tscap321.SaveAsBMP:='Test.bmp';
            tscap322.SaveAsBMP:='Test1.bmp';
        end;
        Thresholding;

end;

//*****//

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure TForm1.Button3Click(Sender: TObject);
begin
    Timer3.Enabled:=false;
    Image1.Picture.Graphic:=ThBMP;
    Image2.Picture.Graphic:=ThBMP1;
    LineShow:=false;
end;
//*****//
procedure TForm1.FormCreate(Sender: TObject);
begin
    Image1.Canvas.Brush.Color:=clgray;
    Image1.Canvas.FillRect(Image1.ClientRect);
    Image2.Canvas.Brush.Color:=clgray;
    Image2.Canvas.FillRect(Image2.ClientRect);
    rfrist:=true;
    caseturn:=1;
end;
//*****//
procedure TForm1.ScrollBar1Change(Sender: TObject);
begin
    Label5.Caption:=IntToStr(ScrollBar1.Position);
    Thresholding;
end;
//*****//
procedure TForm1.ScrollBar2Change(Sender: TObject);
begin
    Label6.Caption:=IntToStr(ScrollBar2.Position);
    Thresholding;
end;
//*****//

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure TForm1.ScrollBar3Change(Sender: TObject);
begin
    Label8.Caption:=IntToStr(ScrollBar3.Position);
    Thresholding;
end;
//*****//
procedure TForm1.ScrollBar4Change(Sender: TObject);
begin
    Label9.Caption:=IntToStr(ScrollBar4.Position);
    Thresholding;
end;
//*****//
procedure TForm1.ScrollBar5Change(Sender: TObject);
begin
    Label10.Caption:=IntToStr(ScrollBar5.Position);
    Thresholding;
end;
//*****//
procedure TForm1.ScrollBar6Change(Sender: TObject);
begin
    Label11.Caption:=IntToStr(ScrollBar6.Position);
    Thresholding;
end;
//*****//
procedure TForm1.Button4Click(Sender: TObject);
begin
    Timer2.Interval:=StrToInt(Edit1.Text);
    Timer2.Enabled:=true;
end;
//*****//

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure TForm1.Image1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
var
  My_Point:TPoint;
begin
  My_Point:=Point(X,Y);
  Label17.Caption:=IntToStr(My_Point.X)+'.'+IntToStr(My_Point.Y);
end;
//*****//
procedure TForm1.Image1MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
var
  My_Point:TPoint;
  PCursor:PRGB;
begin
  My_Point:=Point(X,Y);
  PCursor:=Image1.Picture.Bitmap.ScanLine[My_Point.Y];

  RefRed:=PCurSor[My_Point.X].rgbRed;
  RefGreen:=PCurSor[My_Point.X].rgbGreen;
  RefBlue:=PCurSor[My_Point.X].rgbBlue;

  Manual:=false;
  ScrollBar1.Position:=RefRed;
  ScrollBar2.Position:=RefGreen;
  ScrollBar3.Position:=RefBlue;

  Manual:=true;
  Label35.Caption:=IntToStr(My_Point.X);
  Label19.Caption:=IntToStr(RefRed)+'.'+IntToStr(RefGreen)+'.'+IntToStr(RefBlue);
  Label17.Caption:=IntToStr(My_Point.X)+'.'+IntToStr(My_Point.Y);

  Xleft:=My_Point.X;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Yleft:=My_Point.Y;
end:
;*****//
procedure TForm1.Timer3Timer(Sender: TObject); //tracking
var

StrCenter:string;
c1,c2:integer;
begin
if not(LineShow) then
begin
Xcen:=((Xmax-Xmin) div 2)+Xmin; //find center
Ycen:=((Ymax-Ymin) div 2)+Ymin;
StrCenter:=IntToStr(Xcen)+' '+IntToStr(Ycen); //show center
Label23.Caption:=StrCenter;
Label35.Caption:=IntToStr(Xcen);
//*****Image 1*****//
Image1.Picture.Bitmap.Canvas.Brush.Style:=bsClear;
Image1.Picture.Bitmap.Canvas.Pen.Color:=clRed;
Image1.Picture.Bitmap.Canvas.Pen.Width : 3;
Image1.Picture.Bitmap.Canvas.Ellipse(Xmin,Ymin,Xmax,Ymax);
Image1.Picture.Bitmap.Canvas.MoveTo(Xcen,Ymin);
Image1.Picture.Bitmap.Canvas.LineTo(Xcen,Ymax);
Image1.Picture.Bitmap.Canvas.MoveTo(Xmin,Ycen);
Image1.Picture.Bitmap.Canvas.LineTo(Xmax,Ycen);
LineShow:=true;
//*****Image 2*****//
Xcen1:=((Xmax1-Xmin1) div 2)+Xmin1; //find center
Ycen1:=((Ymax1-Ymin1) div 2)+Ymin1;
Label44.Caption:=IntToStr(Xcen1)+' '+IntToStr(Ycen1);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Label32.Caption:=IntToStr(Xcen1);
Image2.Picture.Bitmap.Canvas.Brush.Style:= bsClear;
Image2.Picture.Bitmap.Canvas.Pen.Color:= clRed;
Image2.Picture.Bitmap.Canvas.Pen.Width : 3;
Image2.Picture.Bitmap.Canvas.Ellipse(Xmin1,Ymin1,Xmax1,Ymax1);
Image2.Picture.Bitmap.Canvas.MoveTo(Xcen1,Ymin1);
Image2.Picture.Bitmap.Canvas.LineTo(Xcen1,Ymax1);
Image2.Picture.Bitmap.Canvas.MoveTo(Xmin1,Ycen1);
Image2.Picture.Bitmap.Canvas.LineTo(Xmax1,Ycen1);
LineShow:=true;
//*****distance*****//
d:=10;
f:=850;
x1:=Xcen;
x2:=Xcen1;
x4:=XXcen;
x5:=XXcen1;
x6:=(x4-x5);
x3:=(x1-x2);
centerLEFT:=Xcen-round((ThBMP.Width)/2);
centerRIGHT:=Xcen1-round((ThBMP1.Width)/2);
if x3=0 then
else
begin
if (Xcen=0) or (Xcen1=0) then
begin
distance:=0;
label28.Caption:=inttostr(distance);
end
else

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

begin
distance:=(f*d)div(x3);
label28.Caption:=inttostr(distance);
end;
end;
if x6=0 then
else
begin
if ((XXcen=0) and (XXcen1=0)) then
begin
distanceblue:=999999;
label72.Caption:=inttostr(distanceblue);
end
else
begin
distanceblue:=(f*d)div(x6);
label72.Caption:=inttostr(distanceblue);
end;
end;
end
else
begin
Image1.Picture.Graphic:=ThBMP;
Image2.Picture.Graphic:=ThBMP1;
LineShow:=false;
end;
//***** Auto Tracking Robot*****//
if (checkbox2.Checked=true) and (radiobutton1.Checked=true) then

```

```
begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if not((Xmin=0) and (Ymin=0) and (Xmax = 0) and (Ymax=0)) then
```

```
begin
```

```
refLEFT:=ThBMP.Width-xcen:
```

```
refRIGHT:=xcen1;
```

```
result:=refLEFT-refRIGHT;
```

```
label24.Caption:=IntToStr(result):
```

```
//*****check Right*****//
```

```
XX:=Xcen-round((ThBMP.Width)/2);
```

```
label24.Caption:=IntToStr(XX);
```

```
//////////check Right//////////
```

```
if (XX>80) then
```

```
begin
```

```
if Comm1.Connect then
```

```
begin
```

```
Comm1.SendByte(my_right);
```

```
Timer4.Interval:=StrToInt(Edit3.Text);
```

```
Timer4.Enabled:=true;
```

```
Edit2.Text:='RIGHT';
```

```
end;
```

```
end;
```

```
//////////check left//////////
```

```
if (XX<(-80)) then
```

```
begin
```

```
if Comm1.Connect then
```

```
begin
```

```
Comm1.SendByte(my_left);
```

```
Timer4.Interval:=StrToInt(Edit3.Text);
```

```
Timer4.Enabled:=true;
```

```
Edit2.Text:='LEFT';
```

```
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

end;

//////////interval stop//////////

if (XX>=-80) and (XX<=80) then
begin
  if Comm1.Connect then
  begin

    Comm1.SendByte(my_stop);
    Timer4.Interval:=StrToInt(Edit3.Text);
    Timer4.Enabled:=true;
    Edit2.Text:='STOP';
    //*****FORWARD AND REVERSE*****//

    if (distance>(zref+20))then //forward
    begin
      Comm1.SendByte(my_reverse);
      Timer4.Interval:=StrToInt(Edit4.Text);
      Timer4.Enabled:=true;
      Edit2.Text:='FORWARD';
    end

    else if (distance<zref-20) then //reverse

    begin

      Comm1.SendByte(my_forward);
      Timer4.Interval:=StrToInt(Edit4.Text);
      Timer4.Enabled:=true;
      Edit2.Text:='REVERSE';
    end
  end
end

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else if (distance<=zref+20) and (distance>=-zref-20) then
begin
Comm1.SendByte(my_stop);
Timer4.Interval:=StrToInt(Edit4.Text);
Timer4.Enabled:=true;
Edit2.Text:='STOP';
end;
end;
end;
end
else
begin
if Comm1.Connect then
begin
Comm1.SendByte(my_stop);
direction:=my_stop;
Edit2.Text:='STOP';
end;
end;
end;

```

\*\*\*\*\*

end;

\*\*\*\*\* Avoid \*\*\*\*\*

if (checkbox2.Checked=true) and (radiobutton2.Checked=true) then

begin

XX:=Xcen-round((ThBMP.Width)/2);

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

XX1:=Xcen1-round((ThBMP1.Width),2);
label24.Caption:=IntToStr(XX);
label78.Caption:=-IntToStr(XX1);
if (distance<=zref+10)and (distance>0) or chkd1 then
begin
    chkd1:=true;
    case caseturn of
        1:
            if not(Timer6.Enabled) then
            begin
                Comm1.SendByte(my_right);
                Comm1.SendByte(my_right);
                Comm1.SendByte(my_right);
                Timer4.Interval:=StrToInt(Edit3.Text);
                Timer4.Enabled:=true;
                Edit2.Text:='RIGHT';
                Label29.Caption:='Now Turn Right';
                Timer6.Interval:=1000;
                Timer6.Enabled:=true;
            end;
        2:
            if not(Timer6.Enabled) then
            begin
                Comm1.SendByte(my_reverse);
                Comm1.SendByte(my_reverse);
                Comm1.SendByte(my_reverse);
                Timer4.Interval:=StrToInt(Edit4.Text);
                Timer4.Enabled:=true;
                Edit2.Text:='FORWARD';
                Label29.Caption:='Now Turn Forward';
            end;
    end;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (distance<=zrcf+10)and (distance>0) then
begin
    caseturn:=3;
    comm1.SendByte(my_stop);
    comm1.SendByte(my_stop);
    comm1.SendByte(my_stop);
end;
end:

```

3:

```

if not(Timer6.Enabled) then
begin
    Comm1.SendByte(my_left);
    Comm1.SendByte(my_left);
    Comm1.SendByte(my_left);
    Timer4.Interval:=StrToInt(Edit3.Text);
    Timer4.Enabled:=true;
    Edit2.Text:='LEFT';
    Label29.Caption:='Now Turn Left';
    Timer6.Interval:=1000;
    Timer6.Enabled:=true;
end;

```

4:

```

if not(Timer6.Enabled) then
begin

    Comm1.SendByte(my_reverse);
    Comm1.SendByte(my_reverse);
    Comm1.SendByte(my_reverse);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Timer4.Interval:=StrToInt(Edit4.Text);
Timer4.Enabled:=true;
Edit2.Text:='FORWARD';
Label29.Caption:='Now Turn Forward';
if (distance<=zref-10)and (distance>0) then
begin
    casturn:=1;
    comm1.SendByte(my_stop);
    comm1.SendByte(my_stop);
    comm1.SendByte(my_stop);
    chkd:=false;
end;
end;
end:
end:
end:
end:
//*****end Timer 3*****//
procedure TForm1.Button5Click(Sender: TObject);
begin
    tscap321.SaveAsBMP:='Test.bmp';
    Image1.Picture.LoadFromFile('Test.bmp');
    tscap322.SaveAsBMP:='Test1.bmp';
    Image2.Picture.LoadFromFile('Test1.bmp');
end;
//*****//
procedure TForm1.Button6Click(Sender: TObject);
begin
    Timer1.Enabled:=false;
    Timer2.Enabled:=false;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Timer3.Enabled:=false;
Timer4.Enabled:=false;
if Comm1.Connect then
begin
    Comm1.SendByte(my_stop);
    direction:=my_stop;
    Edit2.Text:='STOP';
end;
end;
//*****//
procedure TForm1.Timer4Timer(Sender: TObject);
begin
    if Comm1.Connect then
    begin
        Comm1.SendByte(my_stop);
        Edit2.Text:='STOP';
    end;
end;
//*****//
procedure TForm1.Button7Click(Sender: TObject);
begin
    if checkbox1.Checked=false then
        zref:=strtoint(Edit7.Text);
end;
//*****//
procedure TForm1.Image2MouseMove(Sender: TObject; Shift: TShiftState; X,
    Y: Integer);
var
    My_Point1:TPoint;
begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

My_Point1 := Point(X,Y);
Label30.Caption := IntToStr(My_Point1.X) + ',' + IntToStr(My_Point1.Y);
end;

//*****

procedure TForm1.Image2MouseDown(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
var
My_Point: TPoint;
PCursor: PRGB;
begin
My_Point := Point(X,Y);
PCursor := Image2.Picture.Bitmap.ScanLine[My_Point.Y];
RefRed := PCursor[My_Point.X].rgbRed;
RefGreen := PCursor[My_Point.X].rgbGreen;
RefBlue := PCursor[My_Point.X].rgbBlue;
Manual := false;
ScrollBar7.Position := RefRed;
ScrollBar8.Position := RefGreen;
ScrollBar9.Position := RefBlue;
Manual := true;
Label70.Caption := IntToStr(RefRed) + ',' + IntToStr(RefGreen) + ',' + IntToStr(RefBlue);
Label32.Caption := IntToStr(My_Point.X);
Xright := My_Point.X;
Yright := My_Point.Y;
end;

//*****

procedure TForm1.Button8Click(Sender: TObject);
begin
tscap322.Connected := Not tscap322.Connected;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/**.....*/
```

```
procedure TForm1.Button9Click(Sender: TObject);
```

```
begin
```

```
  d:=10;
```

```
  f:=850;
```

```
  distance:=(f*d)div(Xleft-Xright);
```

```
  label28.Caption:=inttostr(distance);
```

```
end;
```

```
/**.....*/
```

```
procedure TForm1.ScrollBar7Change(Sender: TObject);
```

```
begin
```

```
  Label60.Caption:=IntToStr(ScrollBar7.Position);
```

```
  Thresholding;
```

```
end;
```

```
/**.....*/
```

```
procedure TForm1.ScrollBar8Change(Sender: TObject);
```

```
begin
```

```
  Label61.Caption:=IntToStr(ScrollBar8.Position);
```

```
  Thresholding;
```

```
end;
```

```
/**.....*/
```

```
procedure TForm1.ScrollBar9Change(Sender: TObject);
```

```
begin
```

```
  Label62.Caption:=IntToStr(ScrollBar9.Position);
```

```
  Thresholding;
```

```
end;
```

```
/**.....*/
```

```
procedure TForm1.ScrollBar10Change(Sender: TObject);
```

```
begin
```

```
  Label63.Caption:=IntToStr(ScrollBar10.Position);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Thresholding;
end;
//*****//
procedure TForm1.ScrollBar11Change(Sender: TObject);
begin
Label64.Caption:=IntToStr(ScrollBar11.Position);
Thresholding;
end;
//*****//
procedure TForm1.ScrollBar12Change(Sender: TObject);
begin
Label65.Caption:=IntToStr(ScrollBar12.Position);
Thresholding;
end;
//*****save value color*****//
procedure TForm1.Button10Click(Sender: TObject);
begin
//*****IMAGE LEFT*****//
Sredmax:=strtoint(label5.Caption);
SGreenmax:=strtoint(label6.Caption);
Sbluemax:=strtoint(label8.Caption);

Sredmin:=strtoint(label9.Caption);
SGreenmin:=strtoint(label10.Caption);
Sbluemin:=strtoint(label11.Caption);
//*****IMAGE RIGHT*****//
Sredmax1:=strtoint(label60.Caption);
SGreenmax1:=strtoint(label61.Caption);
Sbluemax1:=strtoint(label62.Caption);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Sredmin1:=strtoint(label63.Caption);
SGreenmin1:=strtoint(label64.Caption);
Sbluemin1:=strtoint(label65.Caption);
//*****//

Thresholding;

end;

procedure TForm1.Button11Click(Sender: TObject);
begin
//*****//
SSredmax:=strtoint(label5.Caption);
SSGreenmax:=strtoint(label6.Caption);
SSblucmax:=strtoint(label8.Caption);

SSredmin:=strtoint(label9.Caption);
SSGreenmin:=strtoint(label10.Caption);
SSbluemin:=strtoint(label11.Caption);
//*****//
SSredmax1:=strtoint(label60.Caption);
SSGreenmax1:=strtoint(label61.Caption);
SSbluemax1:=strtoint(label62.Caption);

SSredmin1:=strtoint(label63.Caption);
SSGreenmin1:=strtoint(label64.Caption);
SSbluemin1:=strtoint(label65.Caption);
//*****//

Thresholding;

end;
//*****//

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure TForm1.Button12Click(Sender: TObject);
```

```
begin
```

```
end;
```

```
//*****//
```

```
procedure TForm1.Timer5Timer(Sender: TObject);
```

```
begin
```

```
  case camdirection of
```

```
    cam_control: Comm1.SendByte(cam_control);
```

```
    cam_left : Comm1.SendByte(cam_left);
```

```
    cam_right : Comm1.SendByte(cam_right);
```

```
  end;
```

```
end;
```

```
*****//
```

```
procedure TForm1.Timer7Timer(Sender: TObject);
```

```
begin
```

```
end;
```

```
//*****//
```

```
procedure TForm1.Timer8Timer(Sender: TObject);
```

```
begin
```

```
end;
```

```
//*****//
```

```
procedure TForm1.Timer6Timer(Sender: TObject);
```

```
begin
```

```
  Timer6.Enabled:=false;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

case caseturn of

1:

begin

\*\*\*\*\*

comm1.SendByte(my\_stop);

comm1.SendByte(my\_stop);

comm1.SendByte(my\_stop);

Edit2.Text:='STOP';

Label29.Caption:='Finish Turn Right';

caseturn:=2;

end;

(\*2:

begin

\*\*\*\*\*

comm1.SendByte(my\_stop);

Edit2.Text:='STOP';

Label29.Caption:='Finish Turn Forward';

caseturn:=3;

end;\*)

3:

begin

\*\*\*\*\*

comm1.SendByte(my\_stop);

comm1.SendByte(my\_stop);

comm1.SendByte(my\_stop);

Edit2.Text:='STOP';

Label29.Caption:='Finish Turn Left';

caseturn:=4;

end;

end;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

end;

/\*\*  
\*\*\*\*\*  
\*\*\*/

procedure TForm1.SpeedButton6Up(Sender: TObject; Button: TMouseButton;

Shift: TShiftState; X, Y: Integer);

begin

Timer5.Enabled:=false;

Comm1.SendByte(my\_stop);

end;

procedure TForm1.SpeedButton6Down(Sender: TObject; Button: TMouseButton;

Shift: TShiftState; X, Y: Integer);

begin

if Comm1.Connect then

begin

Comm1.SendByte(cam\_right);

camdirection:=cam\_right;

Timer5.Enabled:=true;

end;

end;

procedure TForm1.SpeedButton5Down(Sender: TObject;

Button: TMouseButton; Shift: TShiftState; X, Y: Integer);

begin

if Comm1.Connect then

begin

Comm1.SendByte(cam\_left);

camdirection:=cam\_left;

Timer5.Enabled:=true;

end;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

end;

procedure TForm1.SpeedButton1Down(Sender: TObject;

Button: TMouseButton; Shift: TShiftState; X, Y: Integer);

begin

if Comm1.Connect then

begin

Comm1.SendByte(my\_forward);

direction:=my\_forward;

Timer1.Enabled:=true;

end;

end;

end.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ข.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Features

Compatible with MCS-51® Products  
8K Bytes of In-System Programmable (ISP) Flash Memory  
– Endurance: 1000 Write/Erase Cycles  
4.0V to 5.5V Operating Range  
Fully Static Operation: 0 Hz to 33 MHz  
Three-level Program Memory Lock  
256 x 8-bit Internal RAM  
32 Programmable I/O Lines  
Three 16-bit Timer/Counters  
Eight Interrupt Sources  
Full Duplex UART Serial Channel  
Low-power Idle and Power-down Modes  
Interrupt Recovery from Power-down Mode  
Watchdog Timer  
Dual Data Pointer  
Power-off Flag

## Description

The AT89S52 is a low-power, high-performance CMOS 8-bit microcontroller with 8K bytes of in-system programmable Flash memory. The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard 80C51 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with in-system programmable Flash on a monolithic chip, the Atmel AT89S52 is a powerful microcontroller which provides a highly-flexible and cost-effective solution to many embedded control applications.

The AT89S52 provides the following standard features: 8K bytes of Flash, 256 bytes RAM, 32 I/O lines, Watchdog timer, two data pointers, three 16-bit timer/counters, a 4-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S52 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 interrupt or hardware reset.



**8-bit  
Microcontroller  
with 8K Bytes  
In-System  
Programmable  
Flash**

**AT89S52**

Rev. 1919A-07/01

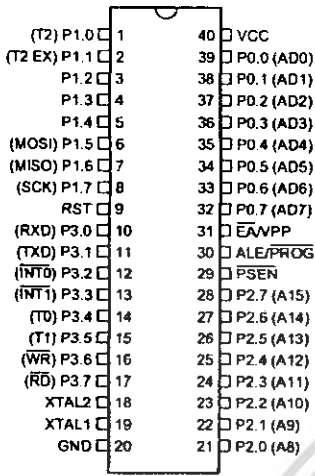


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

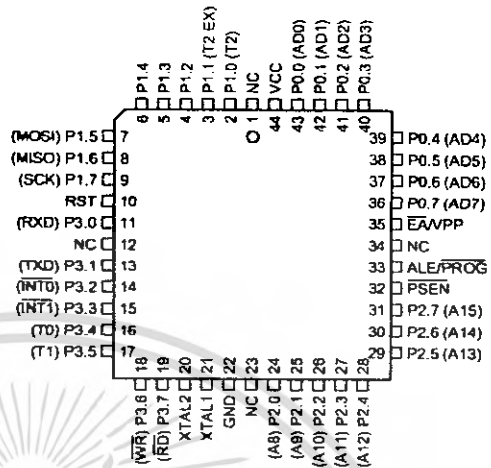


# in Configurations

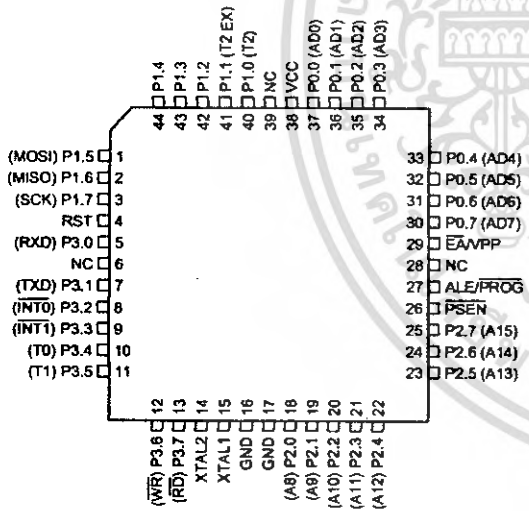
### PDIP



### PLCC



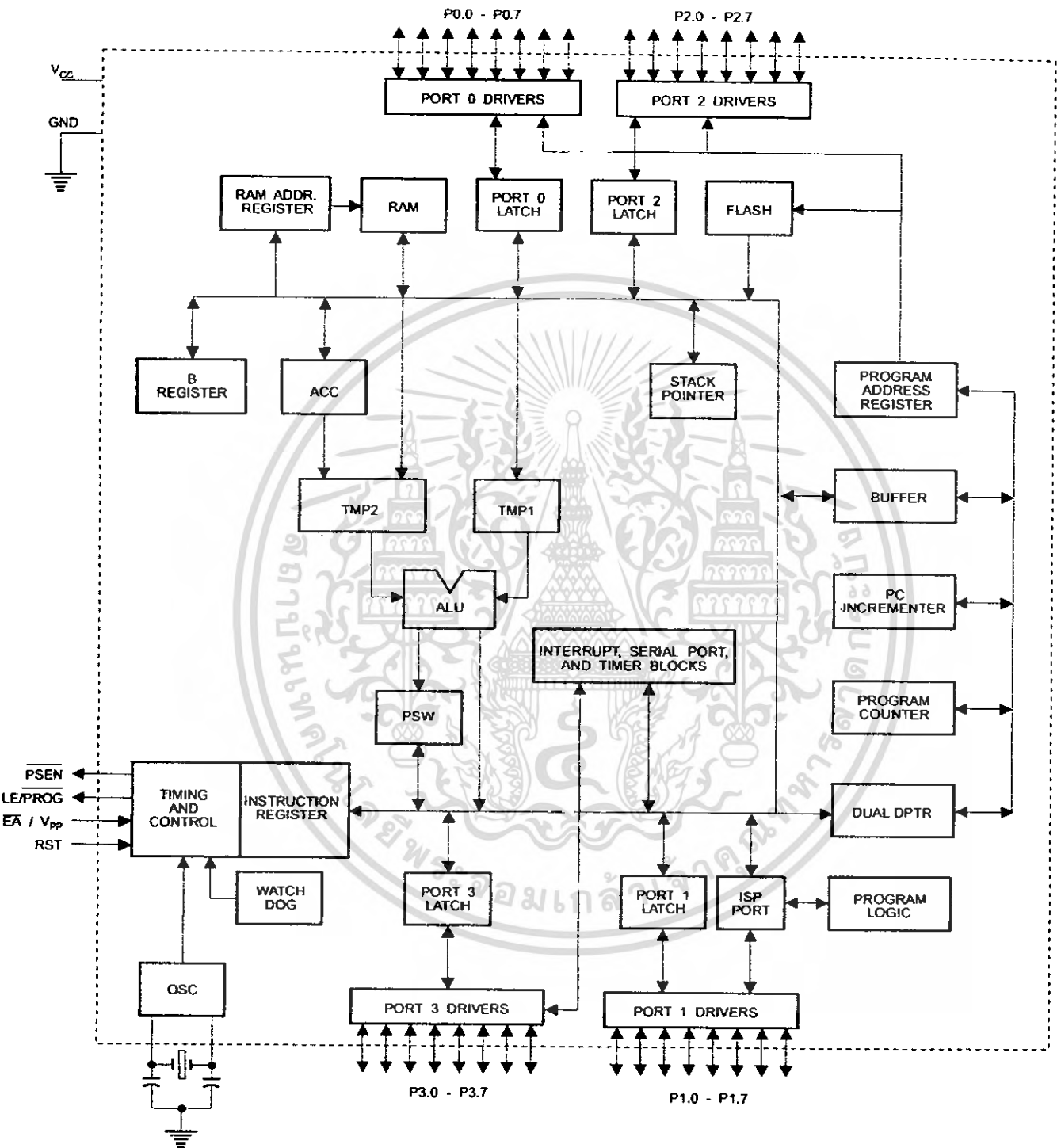
### TQFP



## AT89S52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



## Pin Description

**VCC**  
Supply voltage.

**ND**  
Ground.

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

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

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

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

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

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

Port Pin	Alternate Functions
P1.0	T2 (external count input to Timer/Counter 2), clock-out
P1.1	T2EX (Timer/Counter 2 capture/reload trigger and direction control)
P1.5	MOSI (used for In-System Programming)
P1.6	MISO (used for In-System Programming)
P1.7	SCK (used for In-System Programming)

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

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to

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

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

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

Port 3 also serves the functions of various special features of the AT89S52, as shown in the following table.

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

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

**RST**  
Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device. This pin drives High for 96 oscillator periods after the Watchdog times out. The DISRTO bit in SFR AUXR (address 8EH) can be used to disable this feature. In the default state of bit DISRTO, the RESET HIGH out feature is enabled.

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

In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external data memory.

If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is

## AT89S52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# AT89S52

quickly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

## $\overline{PSEN}$

Program Store Enable ( $\overline{PSEN}$ ) is the read strobe to external program memory.

When the AT89S52 is executing code from external program memory,  $\overline{PSEN}$  is activated twice each machine cycle, except that two  $\overline{PSEN}$  activations are skipped during each access to external data memory.

## $\overline{VPP}$

External Access Enable.  $\overline{EA}$  must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH.

Note, however, that if lock bit 1 is programmed,  $\overline{EA}$  will be internally latched on reset.

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

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

## XTAL1

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

## XTAL2

Output from the inverting oscillator amplifier.

Table 1. AT89S52 SFR Map and Reset Values

0F8H								0FFH
0F0H	B 00000000							0F7H
0E8H								0EFH
0E0H	ACC 00000000							0E7H
0D8H								0DFH
0D0H	PSW 00000000							0D7H
0C8H	T2CON 00000000	T2MOD XXXXXX00	RCAP2L 00000000	RCAP2H 00000000	TL2 00000000	TH2 00000000		0CFH
0C0H								0C7H
0B8H	IP XX000000							0BFH
0B0H	P3 11111111							0B7H
0A8H	IE 0X000000							0AFH
0A0H	P2 11111111		AUXR1 XXXXXXXX0				WDRST XXXXXXXXXX	0A7H
98H	SCON 00000000	SBUF XXXXXXXXXX						9FH
90H	P1 11111111							97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000	AUXR XXX00XX0	8FH
80H	P0 11111111	SP 00000111	DP0L 00000000	DP0H 00000000	DP1L 00000000	DP1H 00000000	PCON 0XXX0000	87H





## Special Function Registers

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

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

user software should not write 1s to these unlisted locations, since they may be used in future products to invoke

new features. In that case, the reset or inactive values of the new bits will always be 0.

**Timer 2 Registers:** Control and status bits are contained in registers T2CON (shown in Table 2) and T2MOD (shown in Table 3) for Timer 2. The register pair (RCAP2H, RCAP2L) are the Capture/Reload registers for Timer 2 in 16-bit capture mode or 16-bit auto-reload mode.

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

Table 2. T2CON – Timer/Counter 2 Control Register

T2CON Address = 0C8H		Reset Value = 0000 0000B						
Bit Addressable								
Bit	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/ $\overline{T2}$	CP/ $\overline{RL2}$
	7	6	5	4	3	2	1	0
Symbol	Function							
TF2	Timer 2 overflow flag set by a Timer 2 overflow and must be cleared by software. TF2 will not be set when either RCLK = 1 or TCLK = 1.							
EXF2	Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX and EXEN2 = 1. When Timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software. EXF2 does not cause an interrupt in up/down counter mode (DCEN = 1).							
RCLK	Receive clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its receive clock in serial port Modes 1 and 3. RCLK = 0 causes Timer 1 overflow to be used for the receive clock.							
TCLK	Transmit clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its transmit clock in serial port Modes 1 and 3. TCLK = 0 causes Timer 1 overflows to be used for the transmit clock.							
EXEN2	Timer 2 external enable. When set, allows a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. EXEN2 = 0 causes Timer 2 to ignore events at T2EX.							
TR2	Start/Stop control for Timer 2. TR2 = 1 starts the timer.							
$\overline{T2}$	Timer or counter select for Timer 2. C/ $\overline{T2}$ = 0 for timer function. C/ $\overline{T2}$ = 1 for external event counter (falling edge triggered).							
CP/ $\overline{RL2}$	Capture/Reload select. CP/ $\overline{RL2}$ = 1 causes captures to occur on negative transitions at T2EX if EXEN2 = 1. CP/ $\overline{RL2}$ = 0 causes automatic reloads to occur when Timer 2 overflows or negative transitions occur at T2EX when EXEN2 = 1. When either RCLK or TCLK = 1, this bit is ignored and the timer is forced to auto-reload on Timer 2 overflow.							

## AT89S52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Table 3a. AUXR: Auxiliary Register**

AUXR	Address = 8EH	Reset Value = XXX00XX0B						
Not Bit Addressable								
Bit	7	6	5	4	3	2	1	0
	–	–	–	WDIDLE	DISRTO	–	–	DISALE
Reserved for future expansion								
DISALE	Disable/Enable ALE							
DISALE	Operating Mode							
0	ALE is emitted at a constant rate of 1/6 the oscillator frequency							
1	ALE is active only during a MOVX or MOVC instruction							
DISRTO	Disable/Enable Reset out							
DISRTO								
0	Reset pin is driven High after WDT times out							
1	Reset pin is input only							
WDIDLE	Disable/Enable WDT in IDLE mode							
WDIDLE								
0	WDT continues to count in IDLE mode							
1	WDT halts counting in IDLE mode							

**Internal Data Pointer Registers:** To facilitate accessing both internal and external data memory, two banks of 16-bit Data Pointer Registers are provided: DP0 at SFR address locations 82H-83H and DP1 at 84H-85H. Bit DPS = 0 selects DP0 and DPS = 1 selects DP1. The user should always initialize the DPS bit to the

appropriate value before accessing the respective Data Pointer Register.

**Power Off Flag:** The Power Off Flag (POF) is located at bit 4 (PCON.4) in the PCON SFR. POF is set to "1" during power up. It can be set and reset under software control and is not affected by reset.

**Table 3b. AUXR1: Auxiliary Register 1**

AUXR1	Address = A2H	Reset Value = XXXXXXX0B						
Not Bit Addressable								
Bit	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	DPS
Reserved for future expansion								
DPS	Data Pointer Register Select							
DPS								
0	Selects DPTR Registers DP0L, DP0H							
1	Selects DPTR Registers DP1L, DP1H							



## Memory Organization

CS-51 devices have a separate address space for Program and Data Memory. Up to 64K bytes each of external Program and Data Memory can be addressed.

### Program Memory

If the  $\overline{EA}$  pin is connected to GND, all program fetches are directed to external memory.

In the AT89S52, if  $\overline{EA}$  is connected to  $V_{CC}$ , program fetches to addresses 0000H through 1FFFH are directed to internal memory and fetches to addresses 2000H through 7FFFH are to external memory.

### Data Memory

The AT89S52 implements 256 bytes of on-chip RAM. The upper 128 bytes occupy a parallel address space to the Special Function Registers. This means that the upper 128 bytes have the same addresses as the SFR space but are physically separate from SFR space.

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

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

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

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

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

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

## AT89S52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Watchdog Timer  
One-time Enabled with Reset-out)**

The WDT is intended as a recovery method in situations where the CPU may be subjected to software upsets. The WDT consists of a 13-bit counter and the Watchdog Timer Reset (WDTRST) SFR. The WDT is defaulted to disable on exiting reset. To enable the WDT, a user must write 01EH and 0E1H in sequence to the WDTRST register (SFR location 0A6H). When the WDT is enabled, it will increment every machine cycle while the oscillator is running. The WDT timeout period is dependent on the external clock frequency. There is no way to disable the WDT except through reset (either hardware reset or WDT overflow reset). When WDT overflows, it will drive an output RESET HIGH pulse at the RST pin.

**Reservicing the WDT**

To enable the WDT, a user must write 01EH and 0E1H in sequence to the WDTRST register (SFR location 0A6H). When the WDT is enabled, the user needs to service it by writing 01EH and 0E1H to WDTRST to avoid a WDT overflow. The 13-bit counter overflows when it reaches 8191 (FFFH), and this will reset the device. When the WDT is enabled, it will increment every machine cycle while the oscillator is running. This means the user must reset the WDT at least every 8191 machine cycles. To reset the WDT the user must write 01EH and 0E1H to WDTRST. *WDTRST is a write-only register. The WDT counter cannot be read or written.* When WDT overflows, it will generate an output RESET pulse at the RST pin. The RESET pulse duration is 96xTOSC, where TOSC=1/FOSC. To make the best use of the WDT, it should be serviced in those sections of code that will periodically be executed within the time required to prevent a WDT reset.

**WDT During Power-down and Idle**

In Power-down mode the oscillator stops, which means the WDT also stops. While in Power-down mode, the user does not need to service the WDT. There are two methods of exiting Power-down mode: by a hardware reset or via a self-activated external interrupt which is enabled prior to entering Power-down mode. When Power-down is exited by hardware reset, servicing the WDT should occur as it normally does whenever the AT89S52 is reset. Exiting Power-down with an interrupt is significantly different. The interrupt is held low long enough for the oscillator to stabilize. When the interrupt is brought high, the interrupt is serviced. To prevent the WDT from resetting the device while the interrupt pin is held low, the WDT is not started until the interrupt is pulled high. It is suggested that the WDT be reset during the interrupt service for the interrupt prior to exit Power-down mode.

To ensure that the WDT does not overflow within a few states of exiting Power-down, it is best to reset the WDT just before entering Power-down mode.

Before going into the IDLE mode, the WDIDLE bit in SFR AUXR is used to determine whether the WDT continues to count if enabled. The WDT keeps counting during IDLE (WDIDLE bit = 0) as the default state. To prevent the WDT from resetting the AT89S52 while in IDLE mode, the user should always set up a timer that will periodically exit IDLE, service the WDT, and reenter IDLE mode.

With WDIDLE bit enabled, the WDT will stop to count in IDLE mode and resumes the count upon exit from IDLE.

**UART**

The UART in the AT89S52 operates the same way as the UART in the AT89C51 and AT89C52. For further information on the UART operation, refer to the ATMEL Web site (<http://www.atmel.com>). From the home page, select 'Products', then '8051-Architecture Flash Microcontroller', then 'Product Overview'.

**Timer 0 and 1**

Timer 0 and Timer 1 in the AT89S52 operate the same way as Timer 0 and Timer 1 in the AT89C51 and AT89C52. For further information on the timers' operation, refer to the ATMEL Web site (<http://www.atmel.com>). From the home page, select 'Products', then '8051-Architecture Flash Microcontroller', then 'Product Overview'.

**Timer 2**

Timer 2 is a 16-bit Timer/Counter that can operate as either a timer or an event counter. The type of operation is selected by bit C/T2 in the SFR T2CON (shown in Table 2). Timer 2 has three operating modes: capture, auto-reload (up or down counting), and baud rate generator. The modes are selected by bits in T2CON, as shown in Table 3. Timer 2 consists of two 8-bit registers, TH2 and TL2. In the Timer function, the TL2 register is incremented every machine cycle. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

Table 3. Timer 2 Operating Modes

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



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

### Capture Mode

In the capture mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 is a 16-bit timer counter which upon overflow sets bit TF2 in T2CON.

Figure 5. Timer in Capture Mode

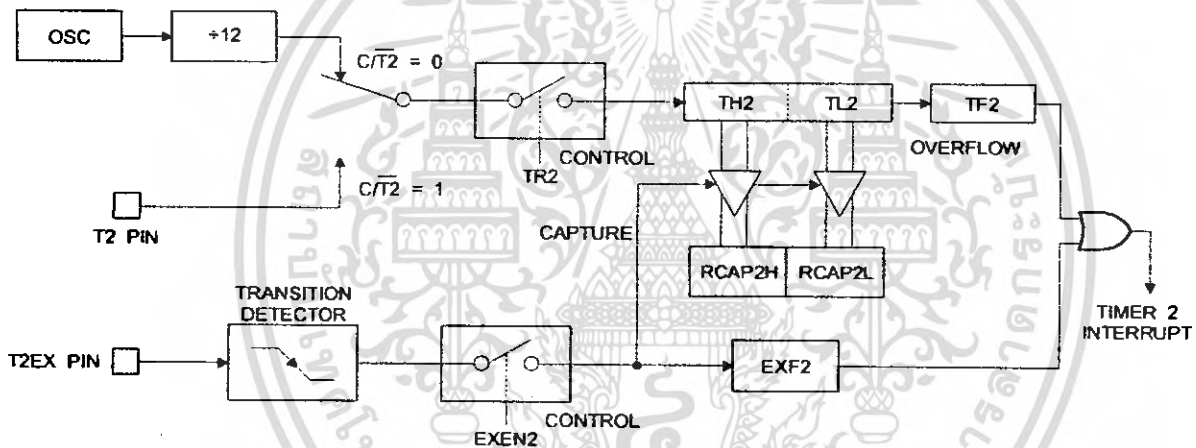


Figure 6 shows Timer 2 automatically counting up when DCEN=0. In this mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 counts up to 0FFFH and then sets the TF2 bit upon overflow. The overflow also causes the timer registers to be reloaded with the 16-bit value in RCAP2H and RCAP2L. The values in the registers in Capture Mode RCAP2H and RCAP2L are preset in software. If EXEN2 = 1, a 16-bit reload can be triggered either by an overflow or by a 1-to-0 transition at external input T2EX. This transition also sets the EXF2 bit. Both the TF2 and EXF2 bits can generate an interrupt if enabled.

Setting the DCEN bit enables Timer 2 to count up or down, as shown in Figure 6. In this mode, the T2EX pin controls

This bit can then be used to generate an interrupt. If EXEN2 = 1, Timer 2 performs the same operation, but a 1-to-0 transition at external input T2EX also causes the current value in TH2 and TL2 to be captured into RCAP2H and RCAP2L, respectively. In addition, the transition at T2EX causes bit EXF2 in T2CON to be set. The EXF2 bit, like TF2, can generate an interrupt. The capture mode is illustrated in Figure 5.

### Auto-reload (Up or Down Counter)

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

the direction of the count. A logic 1 at T2EX makes Timer 2 count up. The timer will overflow at 0FFFFH and set the TF2 bit. This overflow also causes the 16-bit value in RCAP2H and RCAP2L to be reloaded into the timer registers, TH2 and TL2, respectively.

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

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

Figure 6. Timer 2 Auto Reload Mode (DCEN = 0)

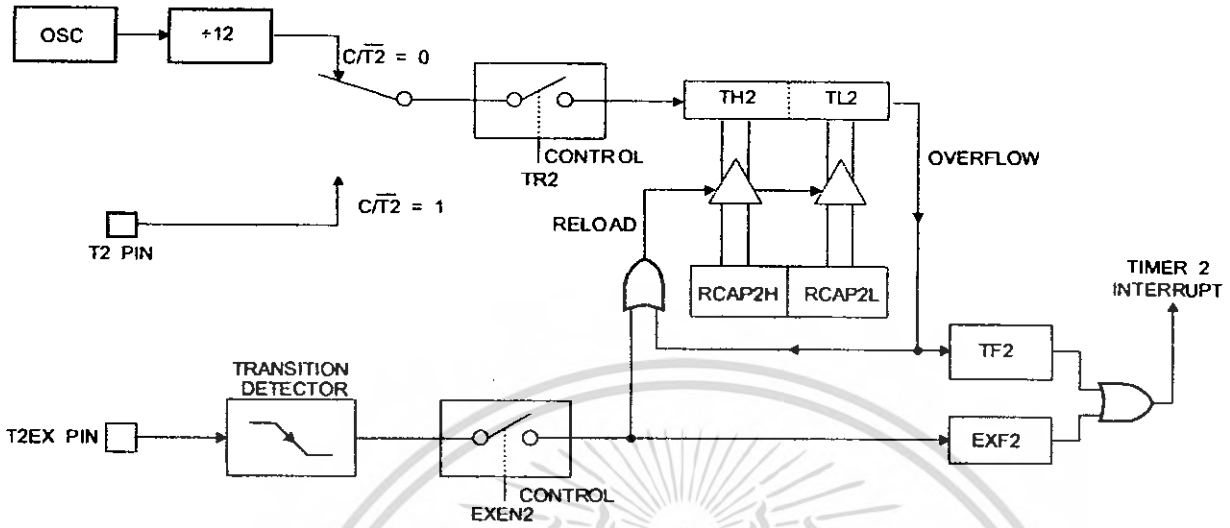


Table 4. T2MOD – Timer 2 Mode Control Register

T2MOD Address = 0C9H

Reset Value = XXXX XX00B

Not Bit Addressable

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

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



Figure 7. Timer 2 Auto Reload Mode (DCEN = 1)

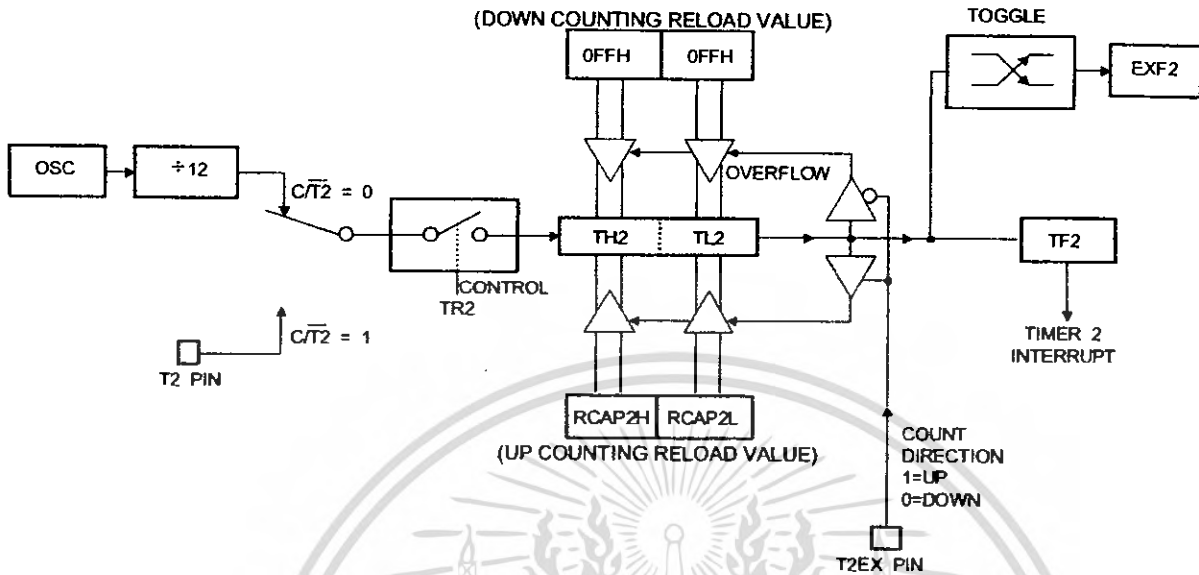
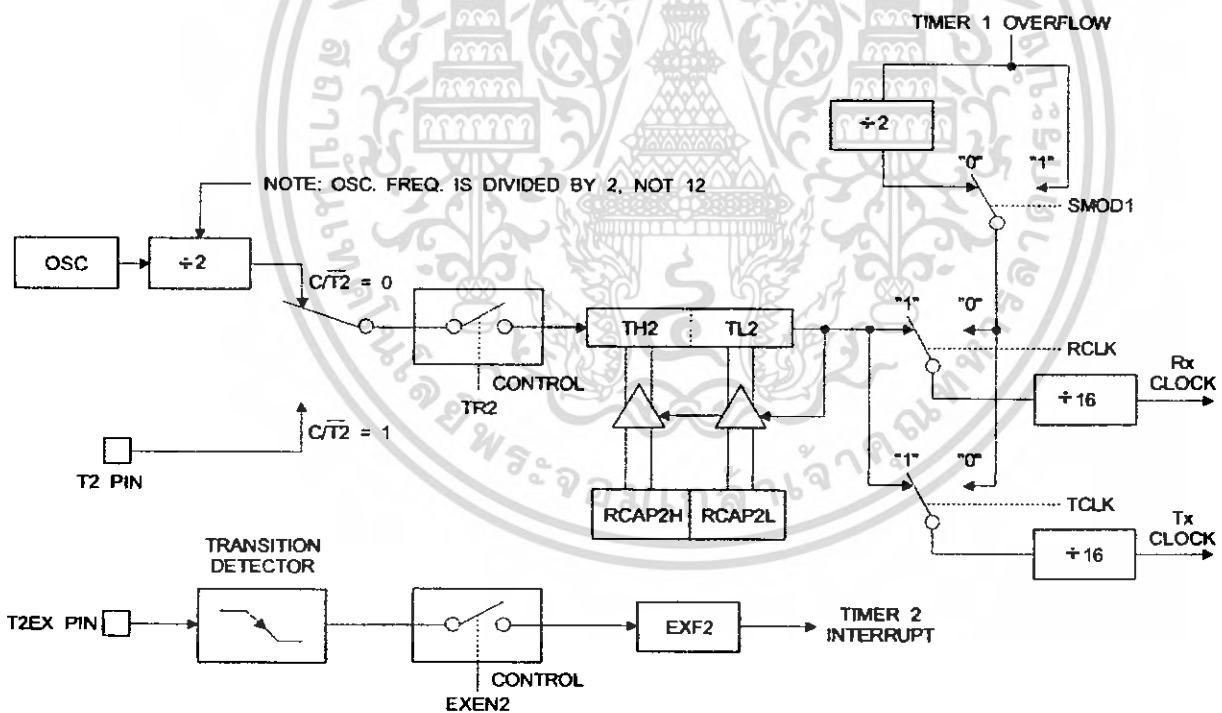


Figure 8. Timer 2 in Baud Rate Generator Mode



## AT89S52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Baud Rate Generator**

Timer 2 is selected as the baud rate generator by setting RCLK and/or RCLK in T2CON (Table 2). Note that the baud rates for transmit and receive can be different if Timer 2 is used for the receiver or transmitter and Timer 1 is used for the other function. Setting RCLK and/or TCLK puts Timer 2 into its baud rate generator mode, as shown in Figure 8.

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

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

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

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

increments every state time (at 1/2 the oscillator frequency). The baud rate formula is given below.

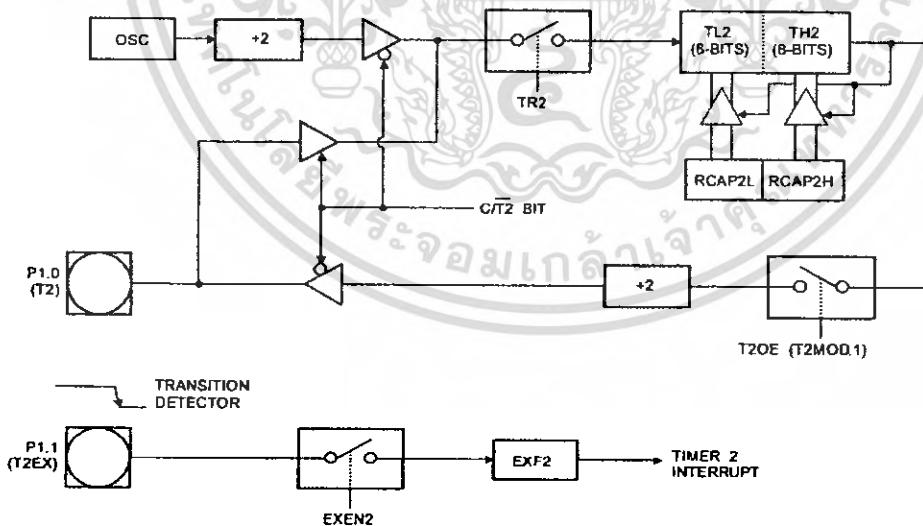
$$\frac{\text{Modes 1 and 3}}{\text{Baud Rate}} = \frac{\text{Oscillator Frequency}}{32 \times \{65536 - \text{RCAP2H}, \text{RCAP2L}\}}$$

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

Timer 2 as a baud rate generator is shown in Figure 8. This figure is valid only if RCLK or TCLK = 1 in T2CON. Note that a rollover in TH2 does not set TF2 and will not generate an interrupt. Note too, that if EXEN2 is set, a 1-to-0 transition in T2EX will set EXF2 but will not cause a reload from (RCAP2H, RCAP2L) to (TH2, TL2). Thus, when Timer 2 is in use as a baud rate generator, T2EX can be used as an extra external interrupt.

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

Figure 9. Timer 2 in Clock-Out Mode



## Programmable Clock Out

A 50% duty cycle clock can be programmed to come out on pin 1, as shown in Figure 9. This pin, besides being a regular I/O pin, has two alternate functions. It can be programmed to input the external clock for Timer/Counter 2 or output a 50% duty cycle clock ranging from 61 Hz to 4 kHz at a 16 MHz operating frequency.

To configure the Timer/Counter 2 as a clock generator, bit 7 (T2CON.1) must be cleared and bit T2OE (T2MOD.1) must be set. Bit TR2 (T2CON.2) starts and stops the timer.

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

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

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

## Interrupts

The AT89S52 has a total of six interrupt vectors: two external interrupts (INT0 and INT1), three timer interrupts (Timer 0, 1, and 2), and the serial port interrupt. These interrupts are all shown in Figure 10.

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

As Table 5 shows that bit position IE.6 is unimplemented. In the AT89S52, bit position IE.5 is also unimplemented. User software should not write 1s to these bit positions, since they may be used in future AT89 products.

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

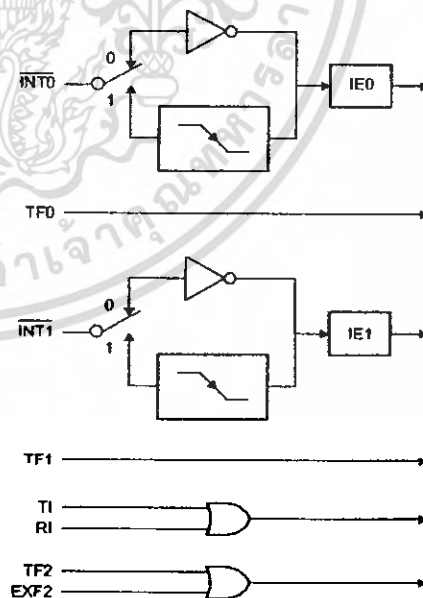
Timer 0 and Timer 1 flags, TF0 and TF1, are set at the end of the cycle in which the timers overflow. The values are then polled by the circuitry in the next cycle. However, Timer 2 flag, TF2, is set at S2P2 and is polled in the next cycle in which the timer overflows.

Table 5. Interrupt Enable (IE) Register

Symbol	Position	Function
EA	IE.7	Disables all interrupts. If EA = 0, no interrupt is acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
—	IE.6	Reserved.
ET2	IE.5	Timer 2 interrupt enable bit.
ES	IE.4	Serial Port interrupt enable bit.
ET1	IE.3	Timer 1 interrupt enable bit.
EX1	IE.2	External interrupt 1 enable bit.
ET0	IE.1	Timer 0 interrupt enable bit.
EX0	IE.0	External interrupt 0 enable bit.

User software should never write 1s to unimplemented bits, because they may be used in future AT89 products.

Figure 10. Interrupt Sources



## AT89S52

## Oscillator Characteristics

XTAL1 and XTAL2 are the input and output, respectively, of an on-chip inverting amplifier that can be configured for use as an on-chip oscillator, as shown in Figure 11. Either a quartz crystal or ceramic resonator may be used. To drive the oscillator from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven, as shown in Figure 12. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry goes through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

## Idle Mode

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

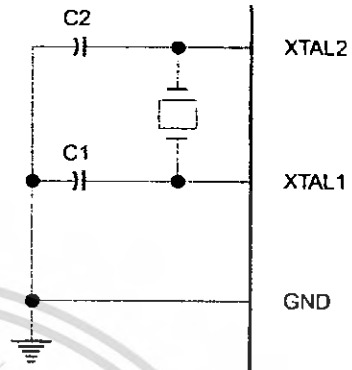
When idle mode is terminated by a hardware reset, the device normally resumes program execution from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware does not restrict access to internal RAM in this event, but access to port pins is not inhibited.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

## Power-down Mode

In Power-down mode, the oscillator is stopped, and the instruction that invokes Power-down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the Power-down mode is terminated. Exit from Power-down mode can be initiated either by a hardware reset or by an enabled external interrupt. The interrupt redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before  $V_{CC}$  is restored to its normal operating level and must be held

active long enough to allow the oscillator to restart and stabilize.

Figure 11. Oscillator Connections



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

Figure 12. External Clock Drive Configuration

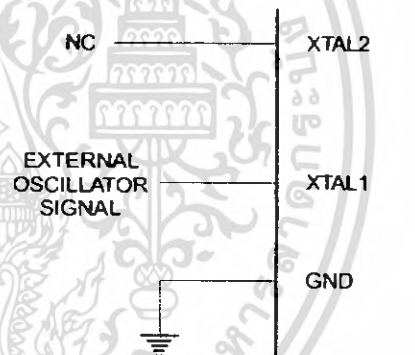


Table 6. Status of External Pins During Idle and Power-down Modes

Mode	Program Memory	ALE	PSEN	PORT0	PORT1	PORT2	PORT3
Normal	Internal	1	1	Data	Data	Data	Data
Normal	External	1	1	Float	Data	Address	Data
Power-down	Internal	0	0	Data	Data	Data	Data
Power-down	External	0	0	Float	Data	Data	Data





## Program Memory Lock Bits

The AT89S52 has three lock bits that can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the following table.

Table 7. Lock Bit Protection Modes

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

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

## Programming the Flash – Parallel Mode

The AT89S52 is shipped with the on-chip Flash memory ready to be programmed. The programming interface provides a high-voltage (12-volt) program enable signal and is compatible with conventional third-party Flash or ROM programmers.

The AT89S52 code memory array is programmed byte-by-byte.

**Programming Algorithm:** Before programming the AT89S52, the address, data, and control signals should be set up according to the Flash programming mode table and Figures 13 and 14. To program the AT89S52, take the following steps:

- Input the desired memory location on the address lines.
- Input the appropriate data byte on the data lines.
- Activate the correct combination of control signals.
- Raise  $\overline{EA}/V_{pp}$  to 12V.
- Pulse ALE/PROG once to program a byte in the Flash array or the lock bits. The byte-write cycle is self-timed and typically takes no more than 50  $\mu$ s.

Repeat steps 1 through 5, changing the address and data for the entire array or until the end of the object file is reached.

**Data Polling:** The AT89S52 features Data Polling to indicate the end of a byte write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written data on P0.7. Once the write cycle has been completed, true data is valid on all outputs, and the next cycle may begin. Data Polling may begin any time after a write cycle has been initiated.

**Ready/Busy:** The progress of byte programming can also be monitored by the RDY/BSY output signal. P3.0 is pulled low after ALE goes high during programming to indicate **READY**. P3.0 is pulled high again when programming is done to indicate **BUSY**.

**Program Verify:** If lock bits LB1 and LB2 have not been programmed, the programmed code data can be read back via the address and data lines for verification. The status of the individual lock bits can be verified directly by reading them back.

**Reading the Signature Bytes:** The signature bytes are read by the same procedure as a normal verification of locations 000H, 100H, and 200H, except that P3.6 and P3.7 must be pulled to a logic low. The values returned are as follows.

- (000H) = 1EH indicates manufactured by Atmel
- (100H) = 52H indicates 89S52
- (200H) = 06H

**Chip Erase:** In the parallel programming mode, a chip erase operation is initiated by using the proper combination of control signals and by pulsing ALE/PROG low for a duration of 200 ns - 500 ns.

In the serial programming mode, a chip erase operation is initiated by issuing the Chip Erase instruction. In this mode, chip erase is self-timed and takes about 500 ms.

During chip erase, a serial read from any address location will return 00H at the data output.

## Programming the Flash – Serial Mode

The Code memory array can be programmed using the serial ISP interface while RST is pulled to  $V_{cc}$ . The serial interface consists of pins SCK, MOSI (input) and MISO (output). After RST is set high, the Programming Enable instruction needs to be executed first before other operations can be executed. Before a reprogramming sequence can occur, a Chip Erase operation is required.

The Chip Erase operation turns the content of every memory location in the Code array into FFH.

Either an external system clock can be supplied at pin XTAL1 or a crystal needs to be connected across pins XTAL1 and XTAL2. The maximum serial clock (SCK)

## AT89S52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

quency should be less than 1/16 of the crystal frequency. With a 33 MHz oscillator clock, the maximum SCK frequency is 2 MHz.

## Serial Programming Algorithm

program and verify the AT89S52 in the serial programming mode, the following sequence is recommended:

**Power-up sequence:**

Apply power between VCC and GND pins.

Set RST pin to "H".

If a crystal is not connected across pins XTAL1 and XTAL2, apply a 3 MHz to 33 MHz clock to XTAL1 pin and wait for at least 10 milliseconds.

Enable serial programming by sending the Programming Enable serial instruction to pin MOSI/P1.5. The frequency of the shift clock supplied at pin SCK/P1.7 needs to be less than the CPU clock at XTAL1 divided by 16.

The Code array is programmed one byte at a time by supplying the address and data together with the

appropriate Write instruction. The write cycle is self-timed and typically takes less than 1 ms at 5V.

4. Any memory location can be verified by using the Read instruction which returns the content at the selected address at serial output MISO/P1.6.
5. At the end of a programming session, RST can be set low to commence normal device operation.

**Power-off sequence (if needed):**

Set XTAL1 to "L" (if a crystal is not used).

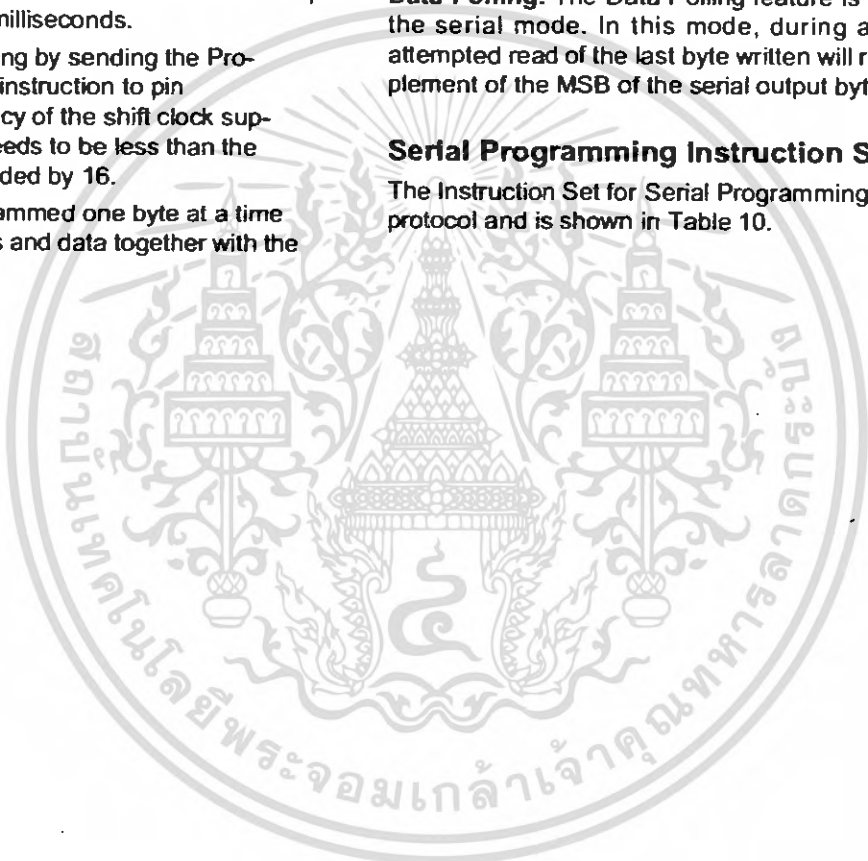
Set RST to "L".

Turn V<sub>CC</sub> power off.

**Data Polling:** The Data Polling feature is also available in the serial mode. In this mode, during a write cycle an attempted read of the last byte written will result in the complement of the MSB of the serial output byte on MISO.

## Serial Programming Instruction Set

The Instruction Set for Serial Programming follows a 4-byte protocol and is shown in Table 10.





## Programming Interface – Parallel Mode

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

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

Table 8. Flash Programming Modes

Mode	V <sub>CC</sub>	RST	PSEN	ALE/ PROG	EA/ V <sub>PP</sub>	P2.6	P2.7	P3.3	P3.6	P3.7	P0.7-0 Data	P2.4-0	P1.7-0
												Address	
Write Code Data	5V	H	L		12V	L	H	H	H	H	D <sub>IN</sub>	A12-8	A7-0
Read Code Data	5V	H	L	H	H	L	L	L	H	H	D <sub>OUT</sub>	A12-8	A7-0
Write Lock Bit 1	5V	H	L		12V	H	H	H	H	H	X	X	X
Write Lock Bit 2	5V	H	L		12V	H	H	H	L	L	X	X	X
Write Lock Bit 3	5V	H	L		12V	H	L	H	H	L	X	X	X
Read Lock Bits 2, 3	5V	H	L	H	H	H	H	L	H	L	P0.2, P0.3, P0.4	X	X
Chip Erase	5V	H	L		12V	H	L	H	L	L	X	X	X
Read Atmel ID	5V	H	L	H	H	L	L	L	L	L	1EH	X 0000	00H
Read Device ID	5V	H	L	H	H	L	L	L	L	L	52H	X 0001	00H
Read Device ID	5V	H	L	H	H	L	L	L	L	L	06H	X 0010	00H

- Notes:
1. Each PROG pulse is 200 ns - 500 ns for Chip Erase.
  2. Each PROG pulse is 200 ns - 500 ns for Write Code Data.
  3. Each PROG pulse is 200 ns - 500 ns for Write Lock Bits.
  4. RDY/BSY signal is output on P3.0 during programming.
  5. X = don't care.

Figure 13. Programming the Flash Memory (Parallel Mode)

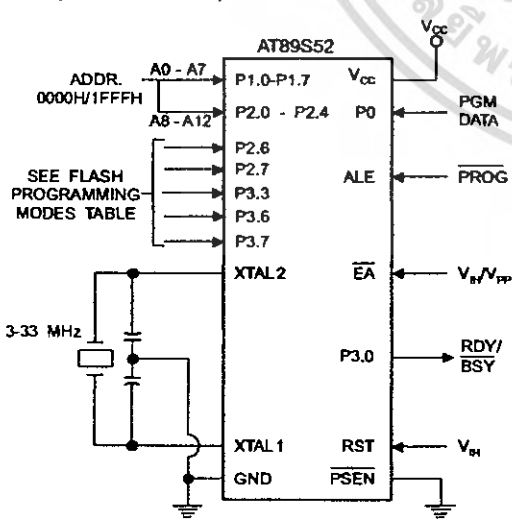
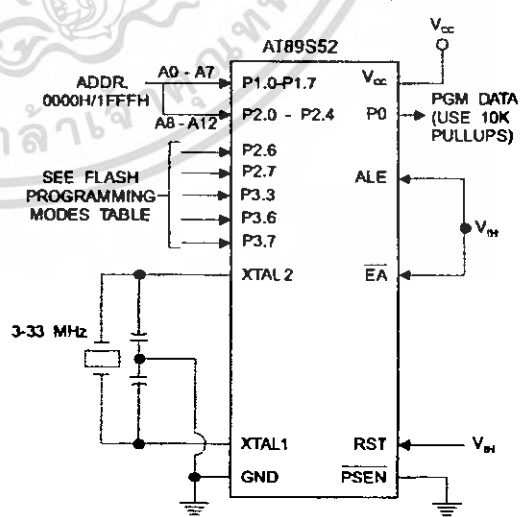


Figure 14. Verifying the Flash Memory (Parallel Mode)



## AT89S52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Flash Programming and Verification Characteristics (Parallel Mode)

= 20°C to 30°C, V<sub>CC</sub> = 4.5 to 5.5V

Symbol	Parameter	Min	Max	Units
V <sub>PP</sub>	Programming Supply Voltage	11.5	12.5	V
I <sub>PP</sub>	Programming Supply Current		10	mA
I <sub>CC</sub>	V <sub>CC</sub> Supply Current		30	mA
f <sub>CLC</sub>	Oscillator Frequency	3	33	MHz
t <sub>AVGL</sub>	Address Setup to $\overline{\text{PROG}}$ Low	48t <sub>CLC</sub>		
t <sub>GHAX</sub>	Address Hold After $\overline{\text{PROG}}$	48t <sub>CLC</sub>		
t <sub>DVGL</sub>	Data Setup to $\overline{\text{PROG}}$ Low	48t <sub>CLC</sub>		
t <sub>HDX</sub>	Data Hold After $\overline{\text{PROG}}$	48t <sub>CLC</sub>		
t <sub>ESH</sub>	P2.7 (ENABLE) High to V <sub>PP</sub>	48t <sub>CLC</sub>		
t <sub>SHGL</sub>	V <sub>PP</sub> Setup to $\overline{\text{PROG}}$ Low	10		μs
t <sub>HSL</sub>	V <sub>PP</sub> Hold After $\overline{\text{PROG}}$	10		μs
t <sub>GLGH</sub>	$\overline{\text{PROG}}$ Width	0.2	1	μs
t <sub>AVQV</sub>	Address to Data Valid		48t <sub>CLC</sub>	
t <sub>ELQV</sub>	$\overline{\text{ENABLE}}$ Low to Data Valid		48t <sub>CLC</sub>	
t <sub>EQZ</sub>	Data Float After $\overline{\text{ENABLE}}$	0	48t <sub>CLC</sub>	
t <sub>GHL</sub>	$\overline{\text{PROG}}$ High to $\overline{\text{BUSY}}$ Low		1.0	μs
t <sub>WC</sub>	Byte Write Cycle Time		50	μs

Figure 15. Flash Programming and Verification Waveforms – Parallel Mode

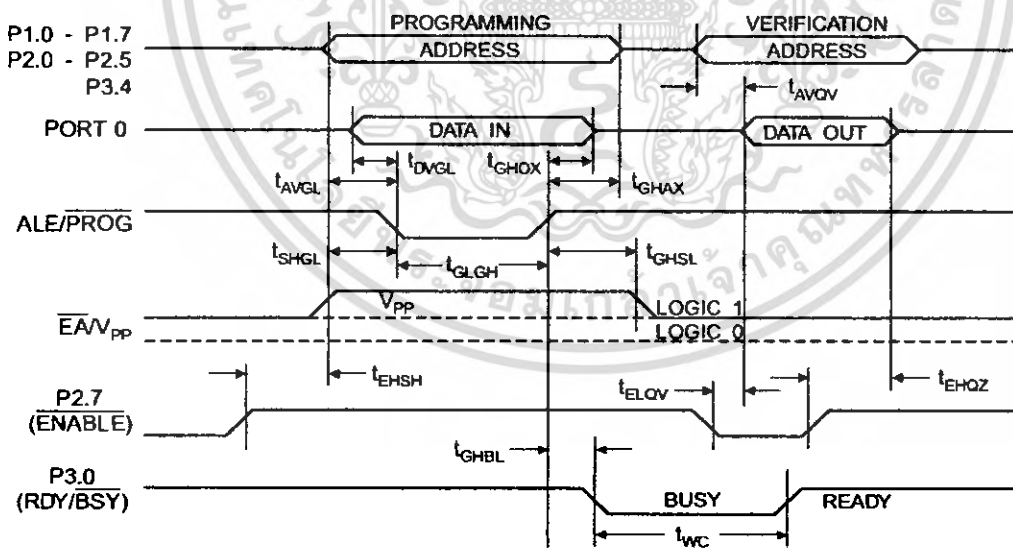
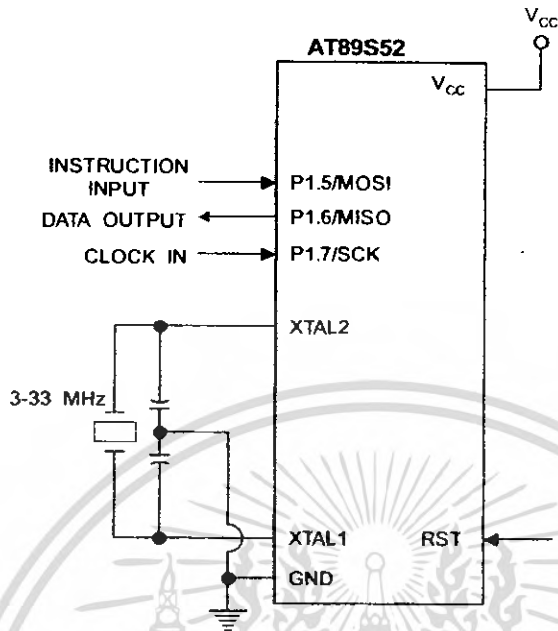


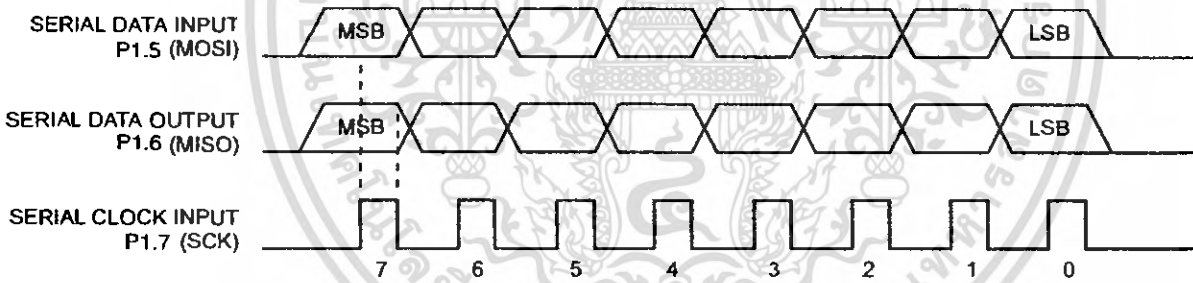


Figure 16. Flash Memory Serial Downloading



### Flash Programming and Verification Waveforms – Serial Mode

Figure 17. Serial Programming Waveforms



## AT89S52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 9. Serial Programming Instruction Set

Instruction	Instruction Format				Operation
	Byte 1	Byte 2	Byte 3	Byte 4	
Programming Enable	1010 1100	0101 0011	xxxx xxxx	xxxx xxxx 0110 1001 (Output)	Enable Serial Programming while RST is high
Chip Erase	1010 1100	100x xxxx	xxxx xxxx	xxxx xxxx	Chip Erase Flash memory array
Read Program Memory (Byte Mode)	0010 0000	xxx A12 A11 A10 A9 A8	A7 A6 A5 A4 A3 A2 A1 A0	DDDD DDDD DDDD DDDD	Read data from Program memory in the byte mode
Write Program Memory (Byte Mode)	0100 0000	xxx A12 A11 A10 A9 A8	A7 A6 A5 A4 A3 A2 A1 A0	DDDD DDDD DDDD DDDD	Write data to Program memory in the byte mode
Write Lock Bits <sup>(2)</sup>	1010 1100	1110 00 B1 B2	xxxx xxxx	xxxx xxxx	Write Lock bits. See Note (2).
Read Lock Bits	0010 0100	xxxx xxxx	xxxx xxxx	xx B2 B1 xx	Read back current status of the lock bits (a programmed lock bit reads back as a '1')
Read Signature Bytes <sup>(1)</sup>	0010 1000	xxx A5 A4 A3 A2 A1 A0	xxx xxxx	Signature Byte	Read Signature Byte
Read Program Memory (Page Mode)	0011 0000	xxx A12 A11 A10 A9 A8	Byte 0	Byte 1... Byte 255	Read data from Program memory in the Page Mode (256 bytes)
Write Program Memory (Page Mode)	0101 0000	xxx A12 A11 A10 A9 A8	Byte 0	Byte 1... Byte 255	Write data to Program memory in the Page Mode (256 bytes)

Notes: 1. The signature bytes are not readable in Lock Bit Modes 3 and 4.

- 2. B1 = 0, B2 = 0 → Mode 1, no lock protection
- B1 = 0, B2 = 1 → Mode 2, lock bit 1 activated
- B1 = 1, B2 = 0 → Mode 3, lock bit 2 activated
- B1 = 1, B2 = 1 → Mode 4, lock bit 3 activated

Each of the lock bits needs to be activated sequentially before Mode 4 can be executed.

When Reset signal is high, SCK should be low for at least 64 system clocks before it goes high to clock in the enable data bytes. No pulsing of Reset signal is necessary. SCK should be no faster than 1/16 of the system clock at V<sub>CC</sub>1.

For Page Read/Write, the data always starts from byte 0 to 255. After the command byte and upper address byte are latched, each byte thereafter is treated as data until all 256 bytes are shifted in/out. Then the next instruction will be ready to be decoded.





## Serial Programming Characteristics

Figure 18. Serial Programming Timing

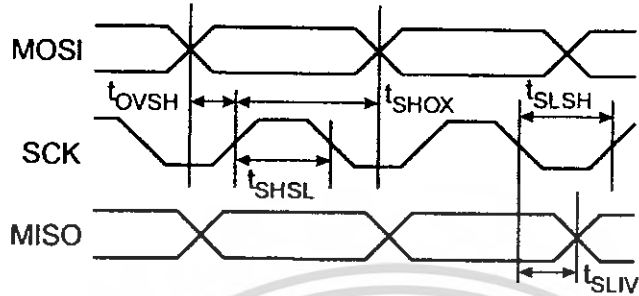


Table 10. Serial Programming Characteristics,  $T_A = -40^\circ\text{C}$  to  $85^\circ\text{C}$ ,  $V_{CC} = 4.0 - 5.5\text{V}$  (Unless otherwise noted)

Symbol	Parameter	Min	Typ	Max	Units
$f_{CLCL}$	Oscillator Frequency	0		33	MHz
$T_{CLCL}$	Oscillator Period	30			ns
$t_{SL}$	SCK Pulse Width High	$2 t_{CLCL}$			ns
$t_{SH}$	SCK Pulse Width Low	$2 t_{CLCL}$			ns
$t_{OVSH}$	MOSI Setup to SCK High	$t_{CLCL}$			ns
$t_{HOX}$	MOSI Hold after SCK High	$2 t_{CLCL}$			ns
$t_{SLIV}$	SCK Low to MISO Valid	10	16	32	ns
$t_{ERASE}$	Chip Erase Instruction Cycle Time			500	ms
$t_{WC}$	Serial Byte Write Cycle Time			$64 t_{CLCL} + 400$	$\mu\text{s}$

## AT89S52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Absolute Maximum Ratings\*

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

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

## 3 Characteristics

Values shown in this table are valid for  $T_A = -40^\circ\text{C}$  to  $85^\circ\text{C}$  and  $V_{CC} = 4.0\text{V}$  to  $5.5\text{V}$ , unless otherwise noted.

Symbol	Parameter	Condition	Min	Max	Units
	Input Low Voltage	(Except $\overline{EA}$ )	-0.5	$0.2 V_{CC} - 0.1$	V
	Input Low Voltage ( $\overline{EA}$ )		-0.5	$0.2 V_{CC} - 0.3$	V
	Input High Voltage	(Except XTAL1, RST)	$0.2 V_{CC} + 0.9$	$V_{CC} + 0.5$	V
	Input High Voltage (XTAL1, RST)		$0.7 V_{CC}$	$V_{CC} + 0.5$	V
	Output Low Voltage <sup>(1)</sup> (Ports 1,2,3)	$I_{OL} = 1.6 \text{ mA}$		0.45	V
	Output Low Voltage <sup>(1)</sup> (Port 0, ALE, PSEN)	$I_{OL} = 3.2 \text{ mA}$		0.45	V
	Output High Voltage (Ports 1,2,3, ALE, PSEN)	$I_{OH} = -60 \mu\text{A}, V_{CC} = 5\text{V} \pm 10\%$	2.4		V
		$I_{OH} = -25 \mu\text{A}$	$0.75 V_{CC}$		V
		$I_{OH} = -10 \mu\text{A}$	$0.9 V_{CC}$		V
	Output High Voltage (Port 0 in External Bus Mode)	$I_{OH} = -800 \mu\text{A}, V_{CC} = 5\text{V} \pm 10\%$	2.4		V
		$I_{OH} = -300 \mu\text{A}$	$0.75 V_{CC}$		V
		$I_{OH} = -80 \mu\text{A}$	$0.9 V_{CC}$		V
	Logical 0 Input Current (Ports 1,2,3)	$V_{IN} = 0.45\text{V}$		-50	$\mu\text{A}$
	Logical 1 to 0 Transition Current (Ports 1,2,3)	$V_{IN} = 2\text{V}, V_{CC} = 5\text{V} \pm 10\%$		-650	$\mu\text{A}$
	Input Leakage Current (Port 0, $\overline{EA}$ )	$0.45 < V_{IN} < V_{CC}$		$\pm 10$	$\mu\text{A}$
RST	Reset Pull-down Resistor		10	30	k $\Omega$
	Pin Capacitance	Test Freq. = 1 MHz, $T_A = 25^\circ\text{C}$		10	pF
	Power Supply Current	Active Mode, 12 MHz		25	mA
		Idle Mode, 12 MHz		6.5	mA
	Power-down Mode <sup>(1)</sup>	$V_{CC} = 5.5\text{V}$		50	$\mu\text{A}$

Notes: 1. Under steady state (non-transient) conditions,  $I_{OL}$  must be externally limited as follows:

Maximum  $I_{OL}$  per port pin: 10 mA

Maximum  $I_{OL}$  per 8-bit port:

Port 0: 26 mA      Ports 1, 2, 3: 15 mA

Maximum total  $I_{OL}$  for all output pins: 71 mA

If  $I_{OL}$  exceeds the test condition,  $V_{OL}$  may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.

2. Minimum  $V_{CC}$  for Power-down is 2V.





## 3 Characteristics

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

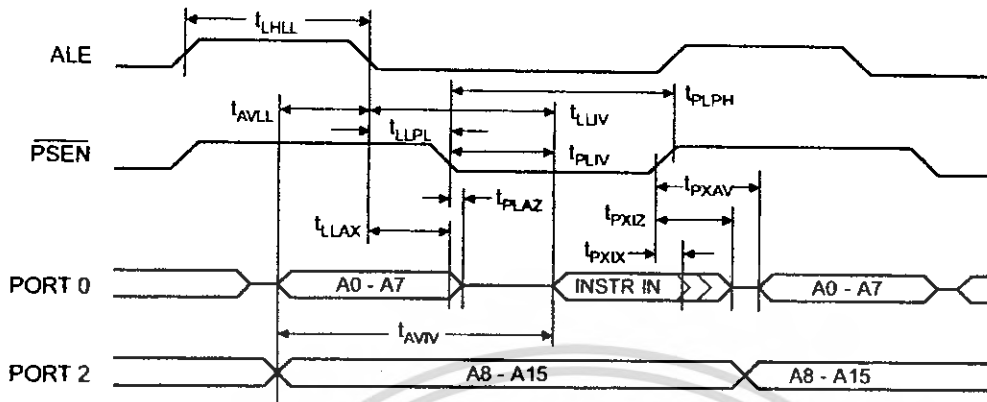
### External Program and Data Memory Characteristics

Symbol	Parameter	12 MHz Oscillator		Variable Oscillator		Units
		Min	Max	Min	Max	
$t_{\text{CLC}}$	Oscillator Frequency			0	33	MHz
$t_{\text{HLL}}$	ALE Pulse Width	127		$2t_{\text{CLC}}-40$		ns
$t_{\text{VLL}}$	Address Valid to ALE Low	43		$t_{\text{CLC}}-25$		ns
$t_{\text{MAX}}$	Address Hold After ALE Low	48		$t_{\text{CLC}}-25$		ns
$t_{\text{IV}}$	ALE Low to Valid Instruction In		233		$4t_{\text{CLC}}-65$	ns
$t_{\text{PL}}$	ALE Low to $\overline{\text{PSEN}}$ Low	43		$t_{\text{CLC}}-25$		ns
$t_{\text{PH}}$	$\overline{\text{PSEN}}$ Pulse Width	205		$3t_{\text{CLC}}-45$		ns
$t_{\text{IV}}$	$\overline{\text{PSEN}}$ Low to Valid Instruction In		145		$3t_{\text{CLC}}-60$	ns
$t_{\text{IX}}$	Input Instruction Hold After $\overline{\text{PSEN}}$	0		0		ns
$t_{\text{IZ}}$	Input Instruction Float After $\overline{\text{PSEN}}$		59		$t_{\text{CLC}}-25$	ns
$t_{\text{AV}}$	$\overline{\text{PSEN}}$ to Address Valid	75		$t_{\text{CLC}}-8$		ns
$t_{\text{IV}}$	Address to Valid Instruction In		312		$5t_{\text{CLC}}-80$	ns
$t_{\text{AZ}}$	$\overline{\text{PSEN}}$ Low to Address Float		10		10	ns
$t_{\text{RH}}$	RD Pulse Width	400		$6t_{\text{CLC}}-100$		ns
$t_{\text{WH}}$	$\overline{\text{WR}}$ Pulse Width	400		$6t_{\text{CLC}}-100$		ns
$t_{\text{DV}}$	RD Low to Valid Data In		252		$5t_{\text{CLC}}-90$	ns
$t_{\text{DX}}$	Data Hold After $\overline{\text{RD}}$	0		0		ns
$t_{\text{DZ}}$	Data Float After $\overline{\text{RD}}$		97		$2t_{\text{CLC}}-28$	ns
$t_{\text{DV}}$	ALE Low to Valid Data In		517		$8t_{\text{CLC}}-150$	ns
$t_{\text{DV}}$	Address to Valid Data In		585		$9t_{\text{CLC}}-165$	ns
$t_{\text{ML}}$	ALE Low to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ Low	200	300	$3t_{\text{CLC}}-50$	$3t_{\text{CLC}}+50$	ns
$t_{\text{WL}}$	Address to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ Low	203		$4t_{\text{CLC}}-75$		ns
$t_{\text{WX}}$	Data Valid to $\overline{\text{WR}}$ Transition	23		$t_{\text{CLC}}-30$		ns
$t_{\text{WH}}$	Data Valid to $\overline{\text{WR}}$ High	433		$7t_{\text{CLC}}-130$		ns
$t_{\text{DX}}$	Data Hold After $\overline{\text{WR}}$	33		$t_{\text{CLC}}-25$		ns
$t_{\text{AZ}}$	$\overline{\text{RD}}$ Low to Address Float		0		0	ns
$t_{\text{LH}}$	$\overline{\text{RD}}$ or $\overline{\text{WR}}$ High to ALE High	43	123	$t_{\text{CLC}}-25$	$t_{\text{CLC}}+25$	ns

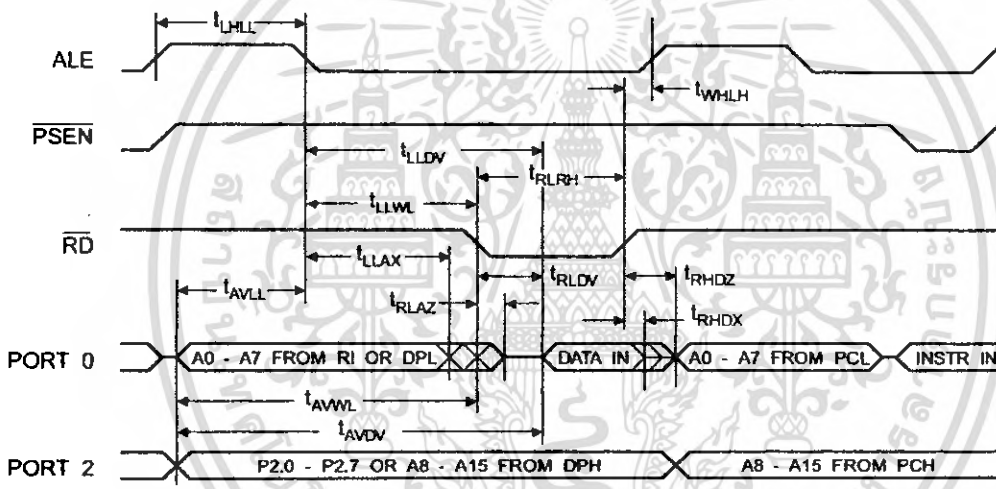
## AT89S52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

External Program Memory Read Cycle

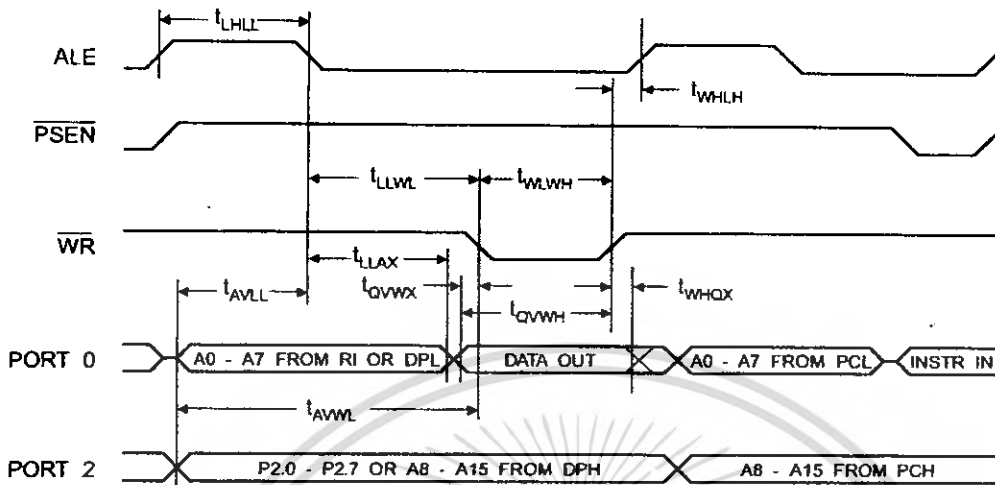


External Data Memory Read Cycle

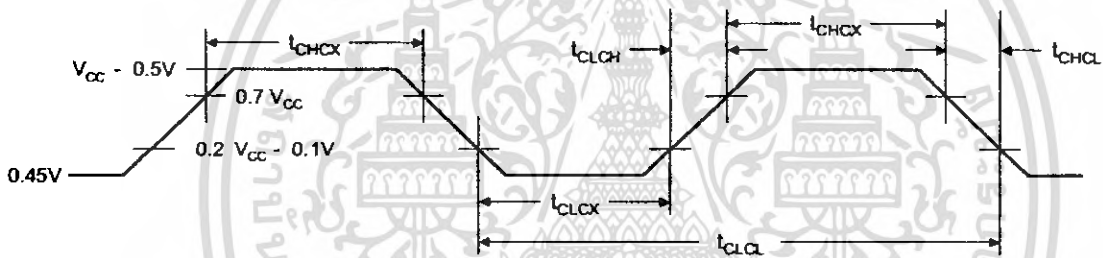


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### External Data Memory Write Cycle



### External Clock Drive Waveforms



### External Clock Drive

Symbol	Parameter	Min	Max	Units
CLCL	Oscillator Frequency	0	33	MHz
CL	Clock Period	30		ns
CHCX	High Time	12		ns
CLCX	Low Time	12		ns
CH	Rise Time		5	ns
CL	Fall Time		5	ns

## AT89S52

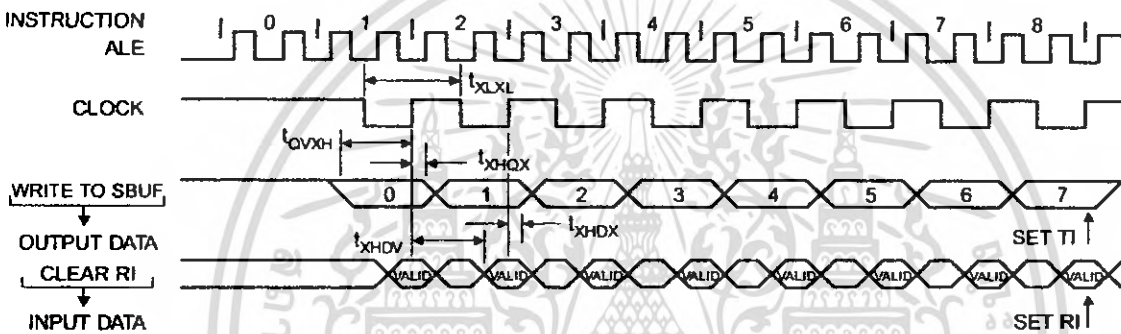
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Serial Port Timing: Shift Register Mode Test Conditions

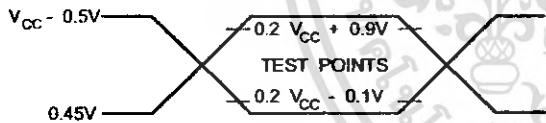
The values in this table are valid for  $V_{CC} = 4.0V$  to  $5.5V$  and Load Capacitance =  $80\text{ pF}$ .

Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
$t_{XL}$	Serial Port Clock Cycle Time	1.0		$12t_{CLCL}$		$\mu s$
$t_{VXH}$	Output Data Setup to Clock Rising Edge	700		$10t_{CLCL} - 133$		ns
$t_{HDX}$	Output Data Hold After Clock Rising Edge	50		$2t_{CLCL} - 80$		ns
$t_{HDV}$	Input Data Hold After Clock Rising Edge	0		0		ns
$t_{HDV}$	Clock Rising Edge to Input Data Valid		700		$10t_{CLCL} - 133$	ns

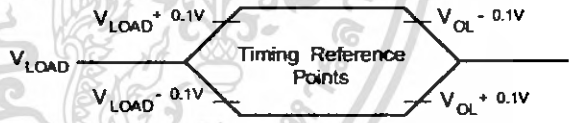
Shift Register Mode Timing Waveforms



Testing Input/Output Waveforms (1)



Float Waveforms (1)



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.

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





## Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
24	4.0V to 5.5V	AT89S52-24AC	44A	Commercial (0° C to 70° C)
		AT89S52-24JC	44J	
		AT89S52-24PC	40P6	
		AT89S52-24AI	44A	Industrial (-40° C to 85° C)
		AT89S52-24JI	44J	
		AT89S52-24PI	40P6	
33	4.5V to 5.5V	AT89S52-33AC	44A	Commercial (0° C to 70° C)
		AT89S52-33JC	44J	
		AT89S52-33PC	40P6	

= Preliminary Availability



Package Type	
A	44-lead, Thin Plastic Gull Wing Quad Flatpack (TQFP)
J	44-lead, Plastic J-headed Chip Carrier (PLCC)
P6	40-pin, 0.600" Wide, Plastic Dual Inline Package (POIP)

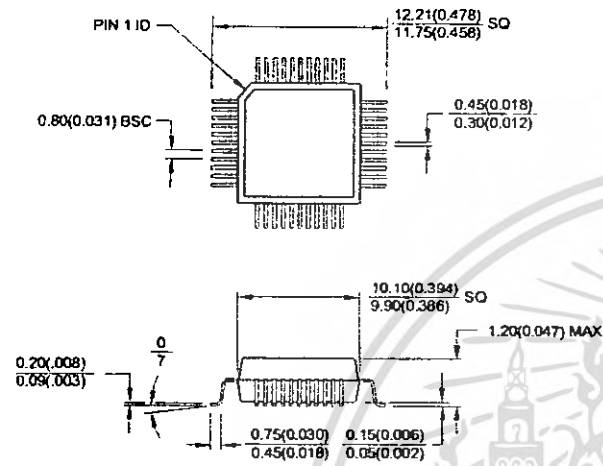
## AT89S52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



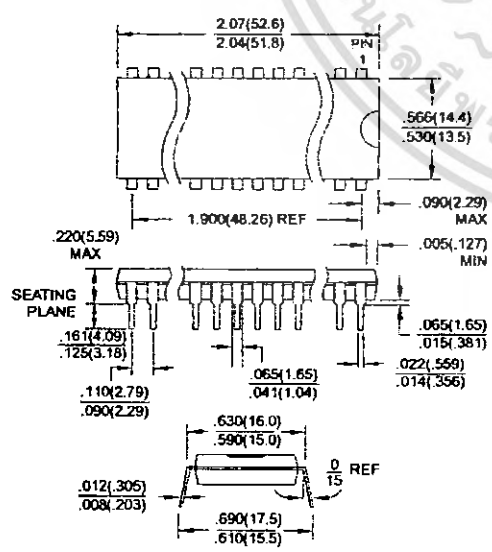
## Packaging Information

**44A, 44-lead, Thin (1.0 mm) Plastic Gull Wing Quad Flat Package (TQFP)**  
 Dimensions in Millimeters and (Inches)\*

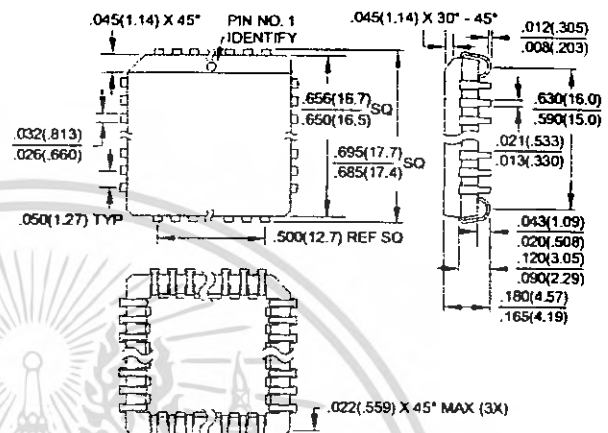


\*Controlling dimension: millimeters

**40P6, 40-pin, 0.600" Wide, Plastic Dual Inline Package (PDIP)**  
 Dimensions in Inches and (Millimeters)  
 JEDEC STANDARD MS-011 AC



**44J, 44-lead, Plastic J-leaded Chip Carrier (PLCC)**  
 Dimensions in Inches and (Millimeters)



## AT89S52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



## Atmel Headquarters

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

**Europe**  
Atmel SarL  
Route des Arsenaux 41  
Casa Postale 80  
CH-1705 Fribourg  
Switzerland  
TEL (41) 26-426-5555  
FAX (41) 26-426-5500

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

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

## Atmel Product Operations

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

**Atmel Grenoble**  
Avenue de Rochepleine  
BP 123  
38521 Saint-Egreve Cedex, France  
TEL (33) 4-7658-3000  
FAX (33) 4-7658-3480

**Atmel Heilbronn**  
Theresienstrasse 2  
POB 3535  
D-74025 Heilbronn, Germany  
TEL (49) 71 31 67 25 94  
FAX (49) 71 31 67 24 23

**Atmel Nantes**  
La Chantrerie  
BP 70602  
44306 Nantes Cedex 3, France  
TEL (33) 0 2 40 18 18 18  
FAX (33) 0 2 40 18 19 60

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

**Atmel Smart Card ICs**  
Scottish Enterprise Technology Park  
East Kilbride, Scotland G75 0QR  
TEL (44) 1355-357-000  
FAX (44) 1355-242-743

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

**e-mail**  
literature@atmel.com

**Web Site**  
<http://www.atmel.com>

**BBS**  
1-(408) 436-4309

### Atmel Corporation 2001.

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

Atmel® is the registered trademark of Atmel.

Intel® is the registered trademark of Intel Corporation. Terms and product names in this document may be trademarks of others.



Printed on recycled paper.

Rev.1919A-07/01/XM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TOSHIBA PHOTOCOUPLER GaAs IRED & PHOTO-TRANSISTOR

# TLP521-1, TLP521-2, TLP521-4

PROGRAMMABLE CONTROLLERS

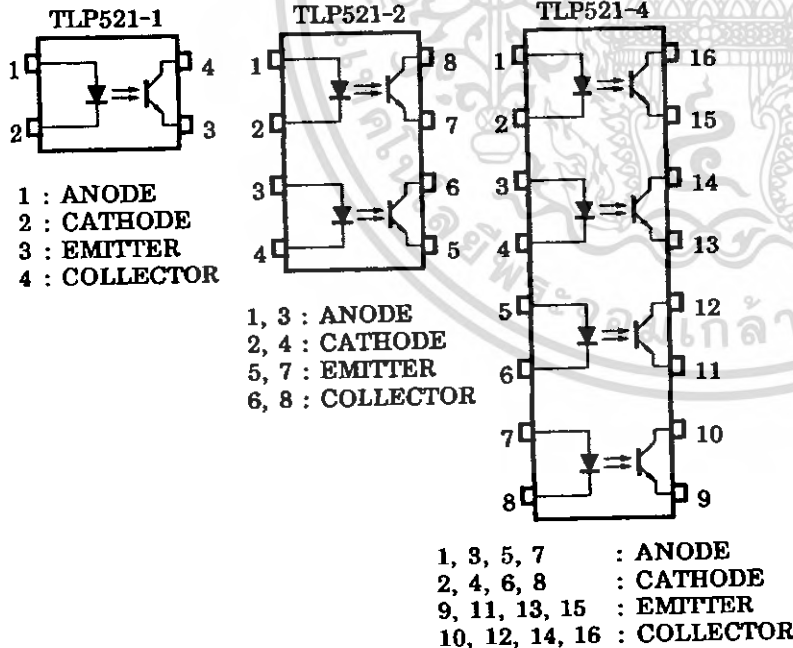
AC/DC-INPUT MODULE

SOLID STATE RELAY

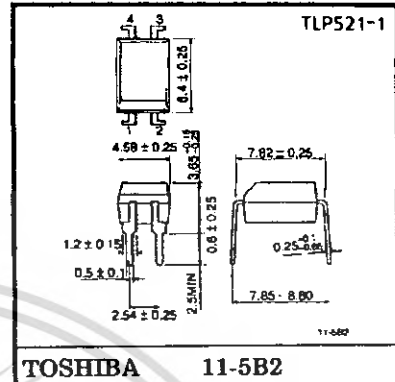
The TOSHIBA TLP521-1, -2 and -4 consist of a photo-transistor optically coupled to a gallium arsenide infrared emitting diode. The TLP521-2 offers two isolated channels in an eight lead plastic DIP package, while the TLP521-4 provides four isolated channels in a sixteen plastic DIP package.

- Collector-Emitter Voltage : 55V (Min.)
- Current Transfer Ratio : 50% (Min.)
- Rank GB : 100% (Min.)
- Isolation Voltage : 2500Vrms (Min.)
- UL Recognized
  - made in Japan : UL1577, File No. E67349
  - made in Thailand : UL1577, File No. E152349

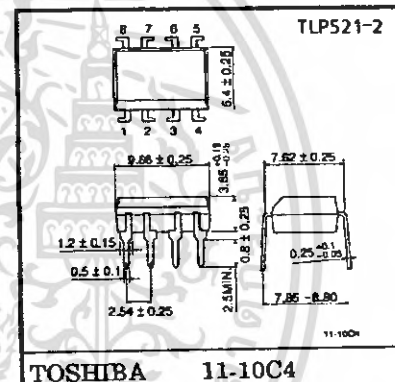
PIN CONFIGURATIONS (TOP VIEW)



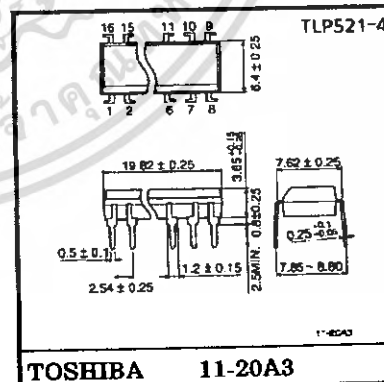
Unit in mm



Weight : 0.26g



Weight : 0.54g



Weight : 1.1g

961001EBC2

TOSHIBA is continually working to improve the quality and the reliability of its products. Nevertheless, semiconductor devices in general can malfunction or fail due to their inherent electrical sensitivity and vulnerability to physical stress. It is the responsibility of the buyer, when utilizing TOSHIBA products, to observe standards of safety, and to avoid situations in which a malfunction or failure of a TOSHIBA product could cause loss of human life, bodily injury or damage to property. In developing your designs, please ensure that TOSHIBA products are used within specified operating ranges as set forth in the most recent products specifications. Also, please keep in mind the precautions and conditions set forth in the TOSHIBA Semiconductor Reliability Handbook.

1998-02-27 1/8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## MAXIMUM RATINGS (Ta = 25°C)

	CHARACTERISTIC	SYMBOL	RATING		UNIT
			TLP521-1	TLP521-2 TLP521-4	
LED	Forward Current	I <sub>F</sub>	70	50	mA
	Forward Current Derating	ΔI <sub>F</sub> /°C	-0.93 (Ta ≥ 50°C)	-0.5 (Ta ≥ 25°C)	mA/°C
	Pulse Forward Current	I <sub>FP</sub>	1 (100μ pulse, 100pps)		A
	Reverse Voltage	V <sub>R</sub>	5		V
	Junction Temperature	T <sub>J</sub>	125		°C
DETECTOR	Collector-Emitter Voltage	V <sub>CEO</sub>	55		V
	Emitter-Collector Voltage	V <sub>ECO</sub>	7		V
	Collector Current	I <sub>C</sub>	50		mA
	Collector Power Dissipation (1 Circuit)	P <sub>C</sub>	150	100	mW
	Collector Power Dissipation Derating (1 Circuit, Ta ≥ 25°C)	ΔP <sub>C</sub> /°C	-1.5	-1.0	mW/°C
	Junction Temperature	T <sub>J</sub>	125		°C
	Storage Temperature Range	T <sub>stg</sub>	-55~125		°C
	Operating Temperature Range	T <sub>opr</sub>	-55~100		°C
	Lead Soldering Temperature	T <sub>sol</sub>	260 (10 sec.)		°C
	Total Package Power Dissipation	P <sub>T</sub>	250	150	mW
Total Package Power Dissipation Derating (Ta ≥ 25°C)	ΔP <sub>T</sub> /°C	-2.5	-1.5	mW/°C	
Isolation Voltage	BV <sub>S</sub>	2500 (AC, 1 min., R.H. ≤ 60%) (Note 1)		V <sub>rms</sub>	

(Note 1) Device considered a two terminal device : LED side pins shorted together and DETECTOR side pins shorted together.

## RECOMMENDED OPERATING CONDITIONS

CHARACTERISTIC	SYMBOL	MIN.	TYP.	MAX.	UNIT
Supply Voltage	V <sub>CC</sub>	—	5	24	V
Forward Current	I <sub>F</sub>	—	16	20	mA
Collector Current	I <sub>C</sub>	—	1	10	mA
Operating Temperature	T <sub>opr</sub>	-25	—	85	°C

961001EBC2

- Gallium arsenide (GaAs) is a substance used in the products described in this document. GaAs dust and fumes are toxic. Do not break, cut or pulverize the product, or use chemicals to dissolve them. When disposing of the products, follow the appropriate regulations. Do not dispose of the products with other industrial waste or with domestic garbage.
- The products described in this document are subject to foreign exchange and foreign trade control laws.
- The information contained herein is presented only as a guide for the applications of our products. No responsibility is assumed by TOSHIBA CORPORATION for any infringements of intellectual property or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any intellectual property or other rights of TOSHIBA CORPORATION or others.
- The information contained herein is subject to change without notice.

1998-02-27 2/8

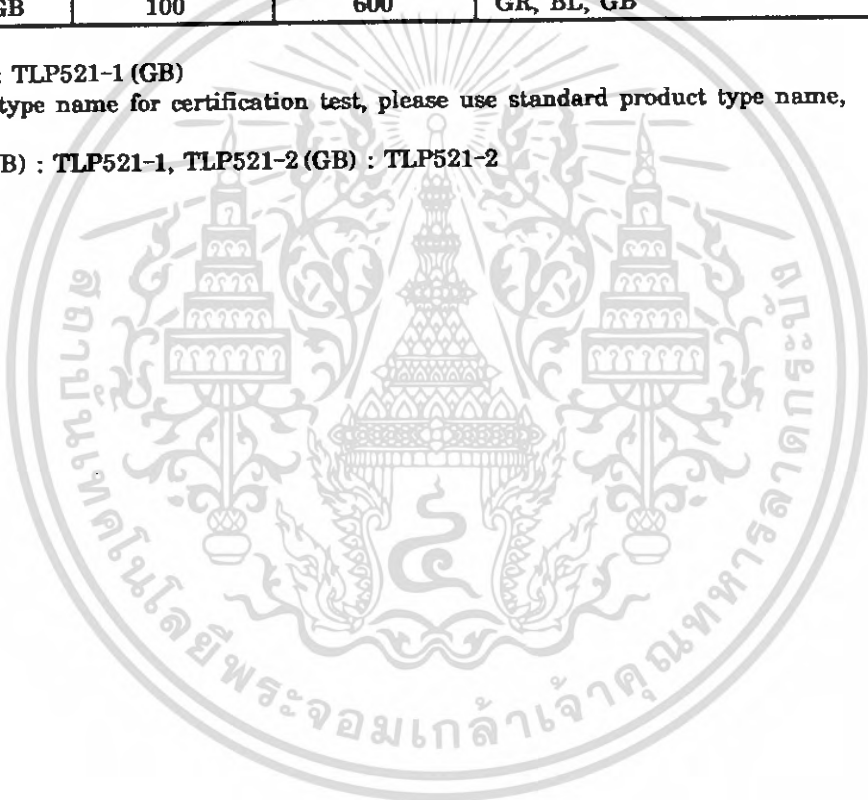
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TYPE	CLASSIFICATION (*1)	CURRENT TRANSFER RATIO (%) ( $I_C / I_F$ )		MARKING OF CLASSIFICATION
		$I_F = 5mA, V_{CE} = 5V, T_a = 25^\circ C$		
		MIN.	MAX.	
TLP521	A	50	600	BLANK, Y, Y <sup>■</sup> , G, G <sup>■</sup> , B, B <sup>■</sup> , GB
	Rank Y	50	150	Y, Y <sup>■</sup>
	Rank GR	100	300	G, G <sup>■</sup>
	Rank BL	200	600	B, B <sup>■</sup>
	Rank GB	100	600	G, G <sup>■</sup> , B, B <sup>■</sup> , GB
TLP521-2	A	50	600	BLANK, GR, BL, GB
TLP521-4	Rank GB	100	600	GR, BL, GB

\*1 : Ex. Rank GB : TLP521-1 (GB)

(Note) Application type name for certification test, please use standard product type name, i.e.

TLP521-1 (GB) : TLP521-1, TLP521-2 (GB) : TLP521-2



INDIVIDUAL ELECTRICAL CHARACTERISTICS (Ta = 25°C)

CHARACTERISTIC		SYMBOL	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
LED	Forward Voltage	V <sub>F</sub>	I <sub>F</sub> = 10mA	1.0	1.15	1.3	V
	Reverse Current	I <sub>R</sub>	V <sub>R</sub> = 5V	—	—	10	μA
	Capacitance	C <sub>T</sub>	V = 0, f = 1MHz	—	30	—	pF
DETECTOR	Collector-Emitter Breakdown Voltage	V <sub>(BR)CEO</sub>	I <sub>C</sub> = 0.5mA	55	—	—	V
	Emitter-Collector Breakdown Voltage	V <sub>(BR)ECO</sub>	I <sub>E</sub> = 0.1mA	7	—	—	V
	Collector Dark Current	I <sub>CEO</sub>	V <sub>CE</sub> = 24V	—	10	100	nA
			V <sub>CE</sub> = 24V, Ta = 85°C	—	2	50	μA
Capacitance (Collector to Emitter)	C <sub>C</sub>	V = 0, f = 1MHz	—	10	—	pF	

COUPLED ELECTRICAL CHARACTERISTICS (Ta = 25°C)

CHARACTERISTIC	SYMBOL	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
Current Transfer Ratio	I <sub>C</sub> /I <sub>F</sub>	I <sub>F</sub> = 5mA, V <sub>CE</sub> = 5V Rank GB	50	—	600	%
			100	—	600	
Saturated CTR	I <sub>C</sub> /I <sub>F(sat)</sub>	I <sub>F</sub> = 1mA, V <sub>CE</sub> = 0.4V Rank GB	—	60	—	%
			30	—	—	
Collector-Emitter Saturation Voltage	V <sub>CE(sat)</sub>	I <sub>C</sub> = 2.4mA, I <sub>F</sub> = 8mA	—	—	0.4	V
		I <sub>C</sub> = 0.2mA, I <sub>F</sub> = 1mA	—	0.2	—	
		Rank GB	—	—	0.4	

ISOLATION CHARACTERISTICS (Ta = 25°C)

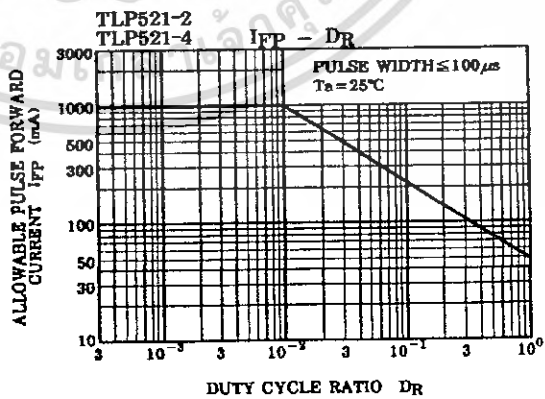
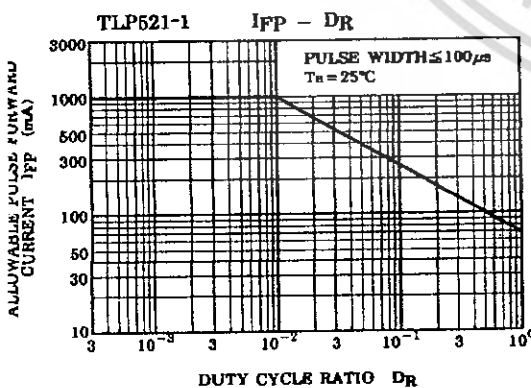
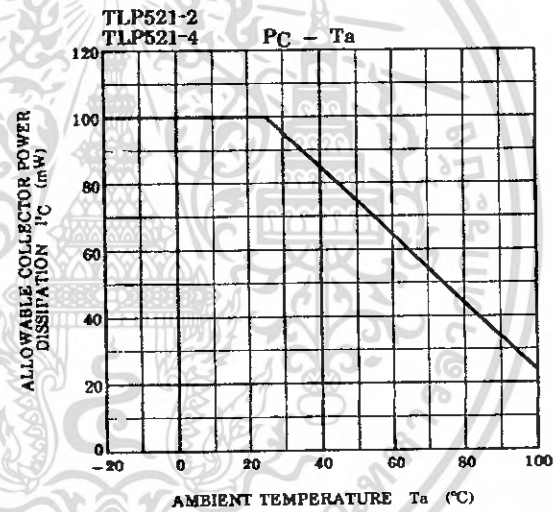
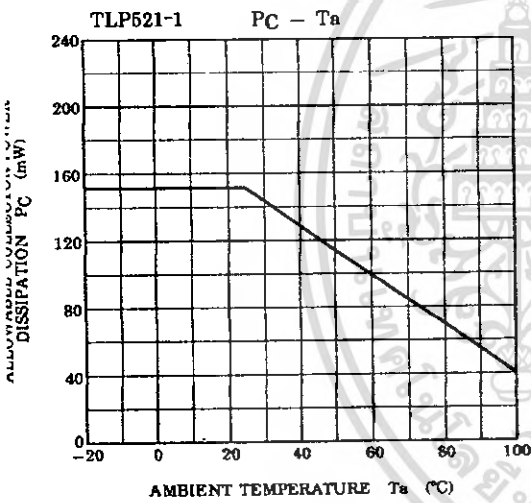
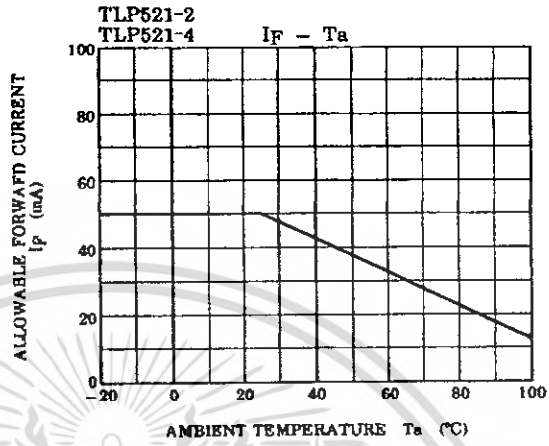
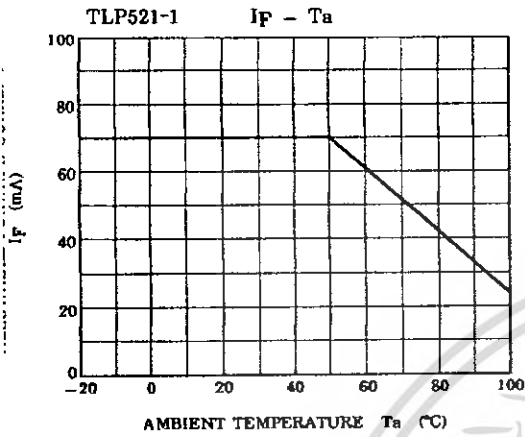
CHARACTERISTIC	SYMBOL	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
Capacitance (Input to Output)	C <sub>S</sub>	V <sub>S</sub> = 0, f = 1MHz	—	0.8	—	pF
Isolation Resistance	R <sub>S</sub>	V <sub>S</sub> = 500V, R.H. ≤ 60%	—	10 <sup>11</sup>	—	Ω
Isolation Voltage	BV <sub>S</sub>	AC, 1 minute	2500	—	—	V <sub>rms</sub>
		AC, 1 second, in oil	—	5000	—	
		DC, 1 minute, in oil	—	5000	—	V <sub>dc</sub>

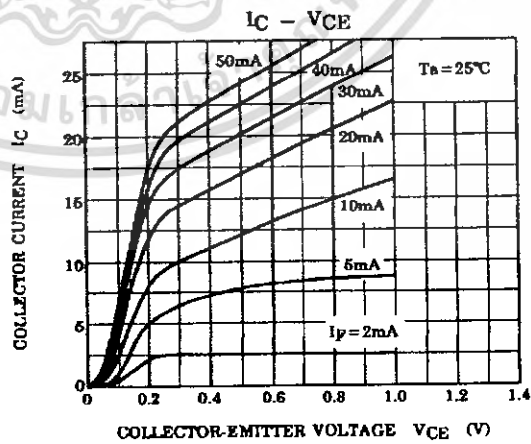
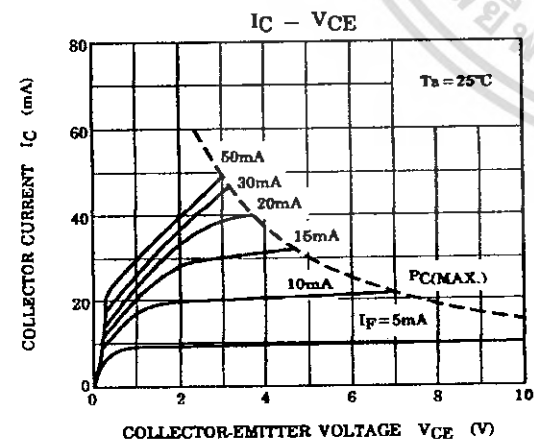
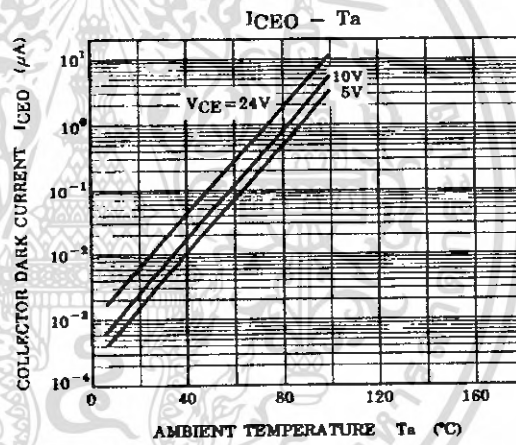
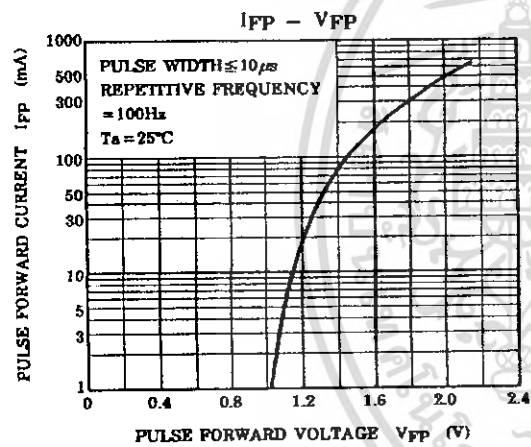
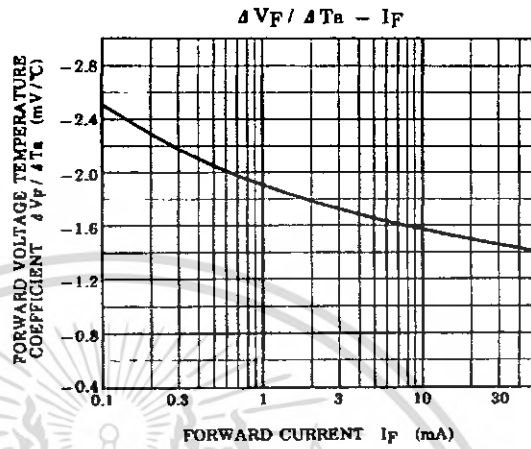
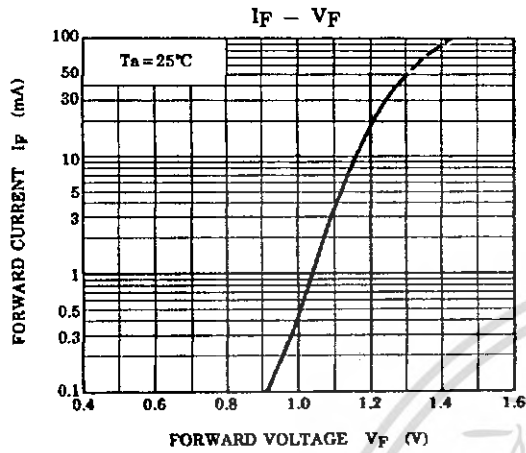
SWITCHING CHARACTERISTICS (Ta = 25°C)

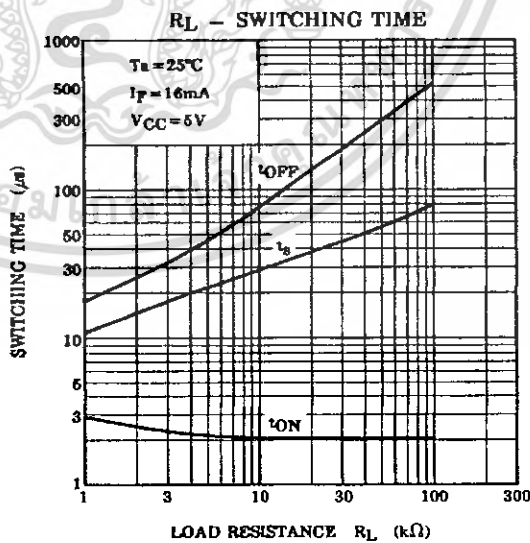
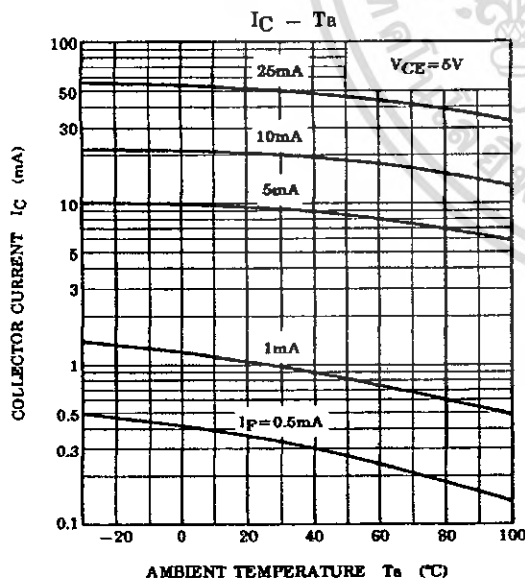
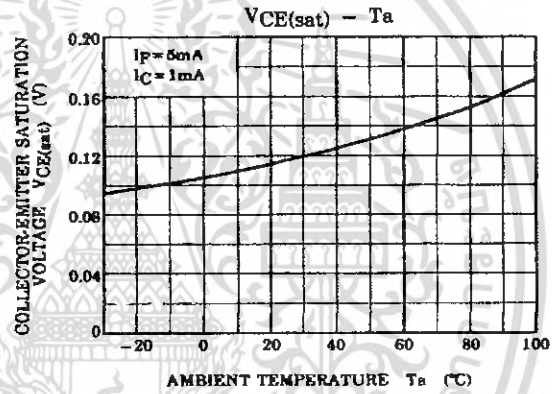
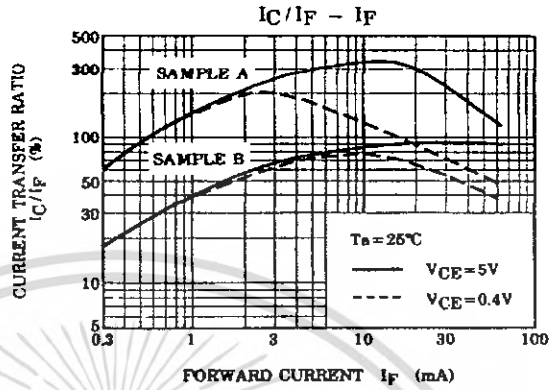
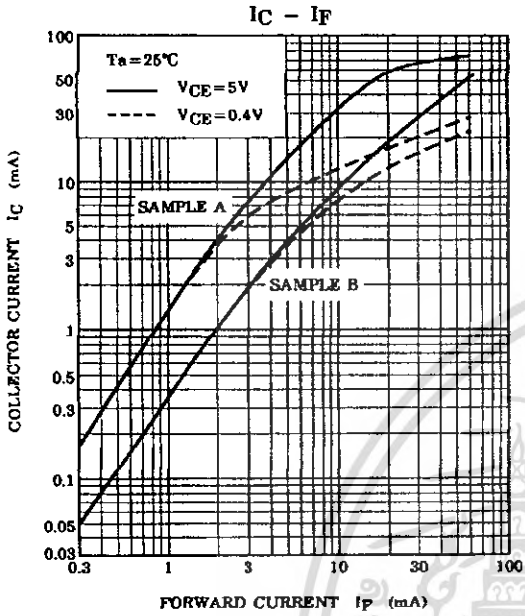
CHARACTERISTIC	SYMBOL	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
Rise Time	$t_r$	$V_{CC} = 10V$ $I_C = 2mA$ $R_L = 100\Omega$	—	2	—	$\mu s$
Fall Time	$t_f$		—	3	—	
Turn-on Time	$t_{on}$		—	3	—	
Turn-off Time	$t_{off}$		—	3	—	
Turn-on Time	$t_{ON}$	$R_L = 1.9k\Omega$ (Fig.1) $V_{CC} = 5V, I_F = 16mA$	—	2	—	$\mu s$
Storage Time	$t_s$		—	15	—	
Turn-off Time	$t_{OFF}$		—	25	—	

Fig.1 SWITCHING TIME TEST CIRCUIT



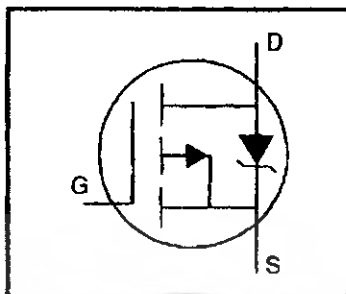






HEXFET® Power MOSFET

Dynamic dv/dt Rating  
 Repetitive Avalanche Rated  
 N-Channel  
 175°C Operating Temperature  
 Fast Switching  
 Ease of Paralleling  
 Simple Drive Requirements



$$V_{DSS} = -100V$$

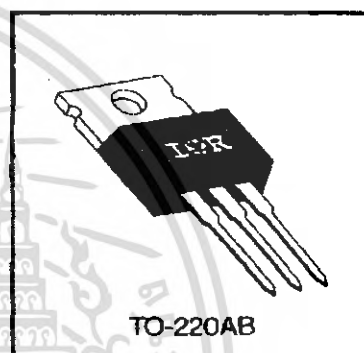
$$R_{DS(on)} = 0.30\Omega$$

$$I_D = -12A$$

### Description

1st Generation HEXFETs from International Rectifier provide the designer the best combination of fast switching, ruggedized device design, low resistance and cost-effectiveness.

TO-220 package is universally preferred for all commercial-industrial applications at power dissipation levels to approximately 50 watts. The low thermal resistance and low package cost of the TO-220 contribute to its wide acceptance throughout the industry.



DATA SHEETS

### Absolute Maximum Ratings

	Parameter	Max.	Units
$T_C = 25^\circ C$	Continuous Drain Current, $V_{GS} @ -10 V$	-12	A
$T_C = 100^\circ C$	Continuous Drain Current, $V_{GS} @ -10 V$	-8.2	
	Pulsed Drain Current ①	-48	
$T_C = 25^\circ C$	Power Dissipation	88	W
	Linear Derating Factor	0.59	W/°C
	Gate-to-Source Voltage	$\pm 20$	V
	Single Pulse Avalanche Energy ②	400	mJ
	Avalanche Current ①	-12	A
	Repetitive Avalanche Energy ①	8.8	mJ
	Peak Diode Recovery dv/dt ③	-5.5	V/ns
	Operating Junction and Storage Temperature Range	-55 to +175	°C
	Soldering Temperature, for 10 seconds	300 (1.6mm from case)	
	Mounting Torque, 6-32 or M3 screw	10 lbf-in (1.1 N-m)	

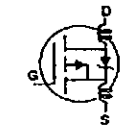
### Thermal Resistance

	Parameter	Min.	Typ.	Max.	Units
	Junction-to-Case	—	—	1.7	°C/W
	Case-to-Sink, Flat, Greased Surface	—	0.50	—	
	Junction-to-Ambient	—	—	62	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ 323 เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าการใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Electrical Characteristics @ $T_J = 25^\circ\text{C}$ (unless otherwise specified)

	Parameter	Min.	Typ.	Max.	Units	Test Conditions
$V_{(BR)DSS}$	Drain-to-Source Breakdown Voltage	-100	—	—	V	$V_{GS}=0V, I_D=-250\mu A$
$\Delta V_{(BR)DSS}/\Delta T_J$	Breakdown Voltage Temp. Coefficient	—	-0.10	—	V/°C	Reference to $25^\circ\text{C}$ , $I_D=-1mA$
$R_{DS(on)}$	Static Drain-to-Source On-Resistance	—	—	0.30	$\Omega$	$V_{GS}=-10V, I_D=-7.2A$ ④
$V_{GS(th)}$	Gate Threshold Voltage	-2.0	—	-4.0	V	$V_{DS}=V_{GS}, I_D=-250\mu A$
$g_{fs}$	Forward Transconductance	3.7	—	—	S	$V_{DS}=-50V, I_D=-7.2A$ ④
$I_{DSS}$	Drain-to-Source Leakage Current	—	—	-100	$\mu A$	$V_{DS}=-100V, V_{GS}=0V$
		—	—	-500		$V_{DS}=-80V, V_{GS}=0V, T_J=150^\circ\text{C}$
$I_{GSS}$	Gate-to-Source Forward Leakage	—	—	-100	nA	$V_{GS}=-20V$
	Gate-to-Source Reverse Leakage	—	—	100		$V_{GS}=20V$
$Q_g$	Total Gate Charge	—	—	38	nC	$I_D=-12A$
$Q_{gs}$	Gate-to-Source Charge	—	—	6.8		$V_{DS}=-80V$
$Q_{gd}$	Gate-to-Drain ("Miller") Charge	—	—	21		$V_{GS}=-10V$ See Fig. 6 and 13 ④
$t_{d(on)}$	Turn-On Delay Time	—	12	—	ns	$V_{DD}=-50V$
$t_r$	Rise Time	—	52	—		$I_D=-12A$
$t_{d(off)}$	Turn-Off Delay Time	—	31	—		$R_G=12\Omega$
$t_f$	Fall Time	—	39	—		$R_D=3.9\Omega$ See Figure 10 ④
$L_D$	Internal Drain Inductance	—	4.5	—	nH	Between lead, 6 mm (0.25in.) from package and center of die contact
$L_S$	Internal Source Inductance	—	7.5	—		
$C_{iss}$	Input Capacitance	—	860	—	pF	$V_{GS}=0V$
$C_{oss}$	Output Capacitance	—	340	—		$V_{DS}=-25V$
$C_{rss}$	Reverse Transfer Capacitance	—	93	—		$f=1.0MHz$ See Figure 5



## Source-Drain Ratings and Characteristics

	Parameter	Min.	Typ.	Max.	Units	Test Conditions
$I_S$	Continuous Source Current (Body Diode)	—	—	-12	A	MOSFET symbol showing the integral reverse p-n junction diode.
$I_{SM}$	Pulsed Source Current (Body Diode) ①	—	—	-48		
$V_{SD}$	Diode Forward Voltage	—	—	-6.3	V	$T_J=25^\circ\text{C}, I_S=-12A, V_{GS}=0V$ ④
$t_{rr}$	Reverse Recovery Time	—	120	240	ns	$T_J=25^\circ\text{C}, I_F=-12A$
$Q_{rr}$	Reverse Recovery Charge	—	0.46	0.92	$\mu C$	$di/dt=100A/\mu s$ ④
$t_{on}$	Forward Turn-On Time	Intrinsic turn-on time is negligible (turn-on is dominated by $L_S+L_D$ )				

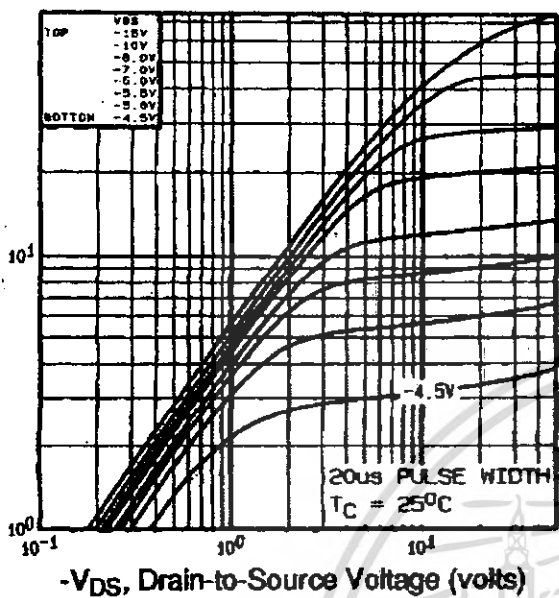
### Notes:

① Repetitive rating; pulse width limited by max. junction temperature (See Figure 11)

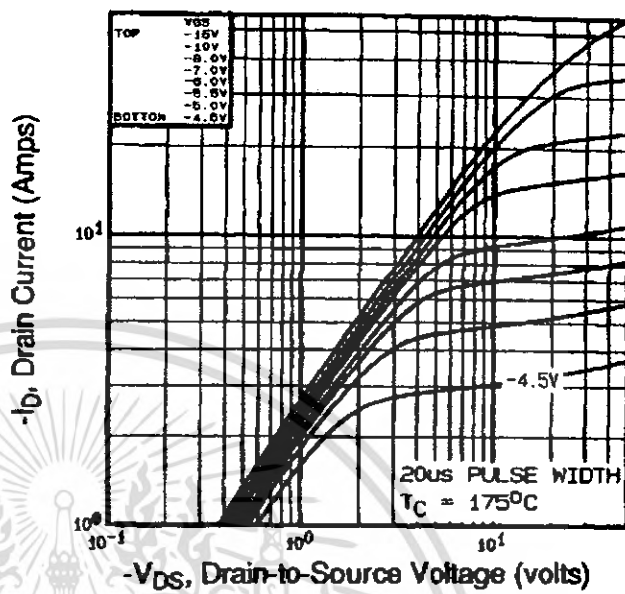
③  $I_{SD} \leq -12A, di/dt \leq 140A/\mu s, V_{DD} \leq V_{(BR)DSS}, T_J \leq 175^\circ\text{C}$

②  $V_{DD}=-25V$ , starting  $T_J=25^\circ\text{C}$ ,  $L=4.2mH$ ,  $R_G=25\Omega, I_{AS}=-12A$  (See Figure 12)

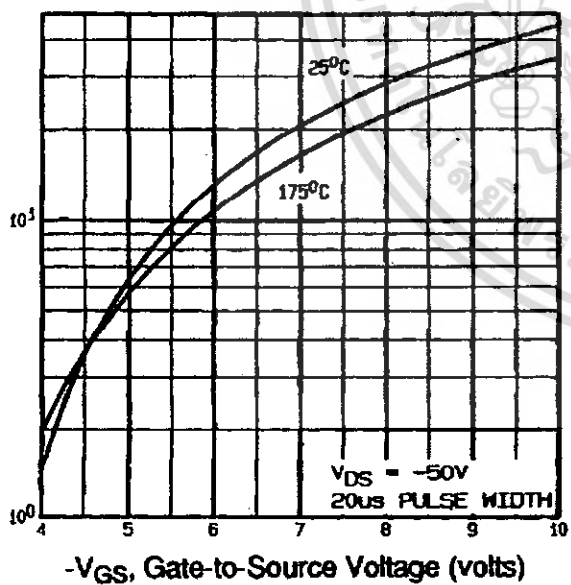
④ Pulse width  $\leq 300 \mu s$ ; duty cycle  $\leq 2\%$ .



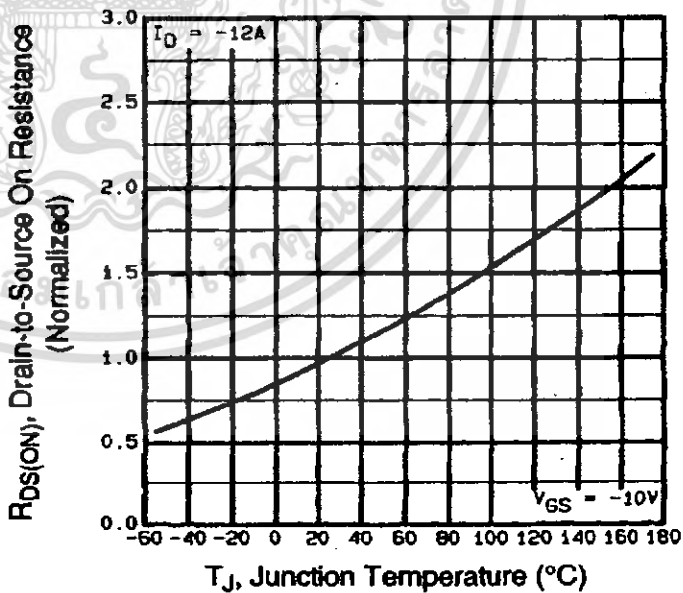
**Fig 1. Typical Output Characteristics,  $T_C=25^\circ\text{C}$**



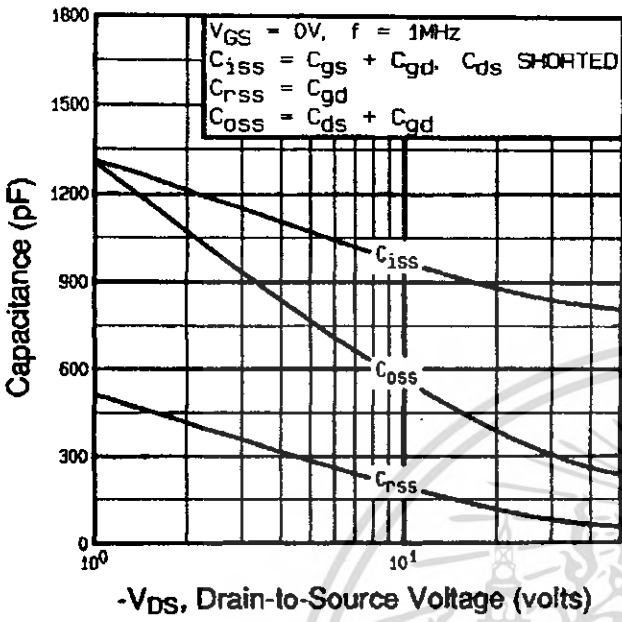
**Fig 2. Typical Output Characteristics,  $T_C=175^\circ\text{C}$**



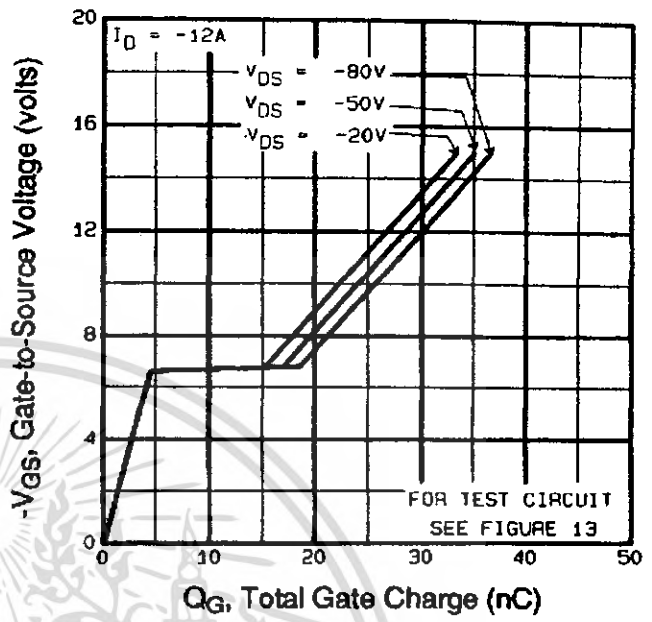
**Fig 3. Typical Transfer Characteristics**



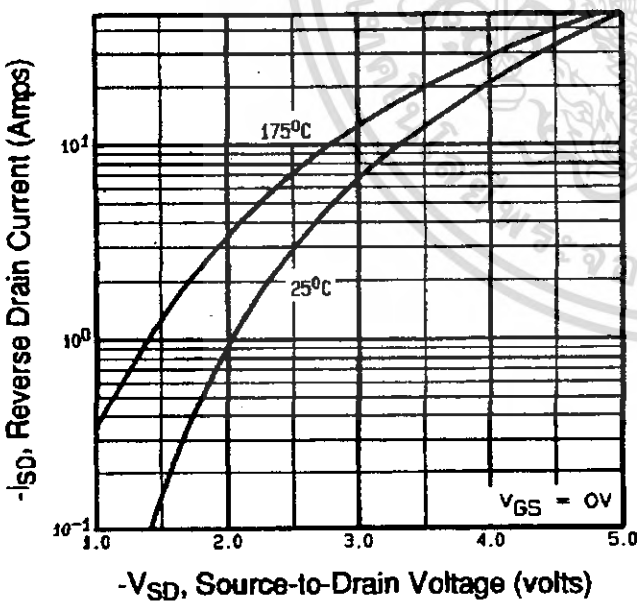
**Fig 4. Normalized On-Resistance Vs. Temperature**



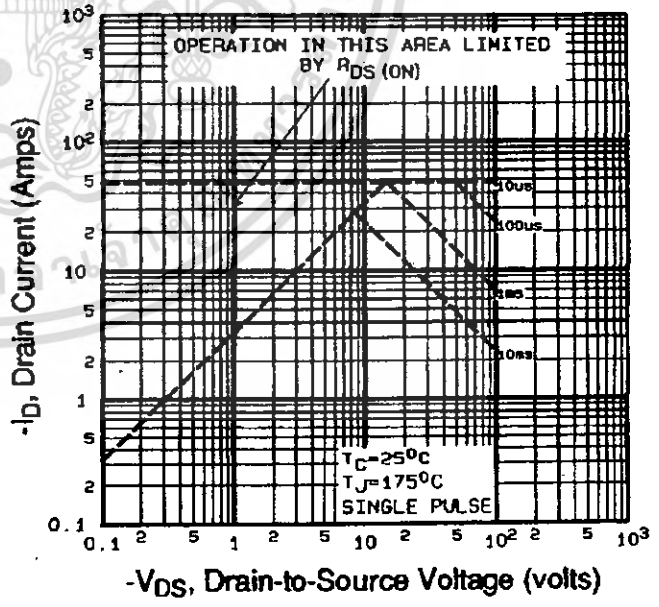
**Fig 5.** Typical Capacitance Vs. Drain-to-Source Voltage



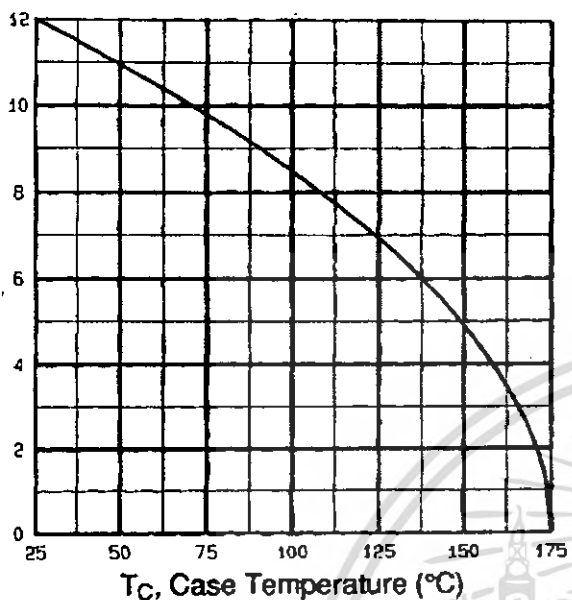
**Fig 6.** Typical Gate Charge Vs. Gate-to-Source Voltage



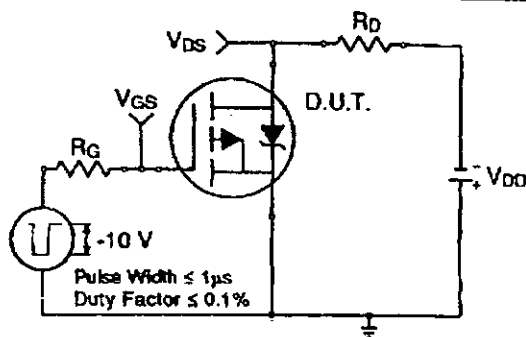
**Fig 7.** Typical Source-Drain Diode Forward Voltage



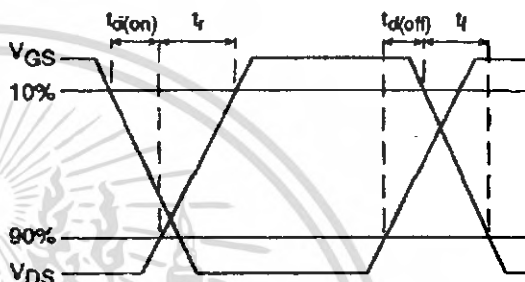
**Fig 8.** Maximum Safe Operating Area



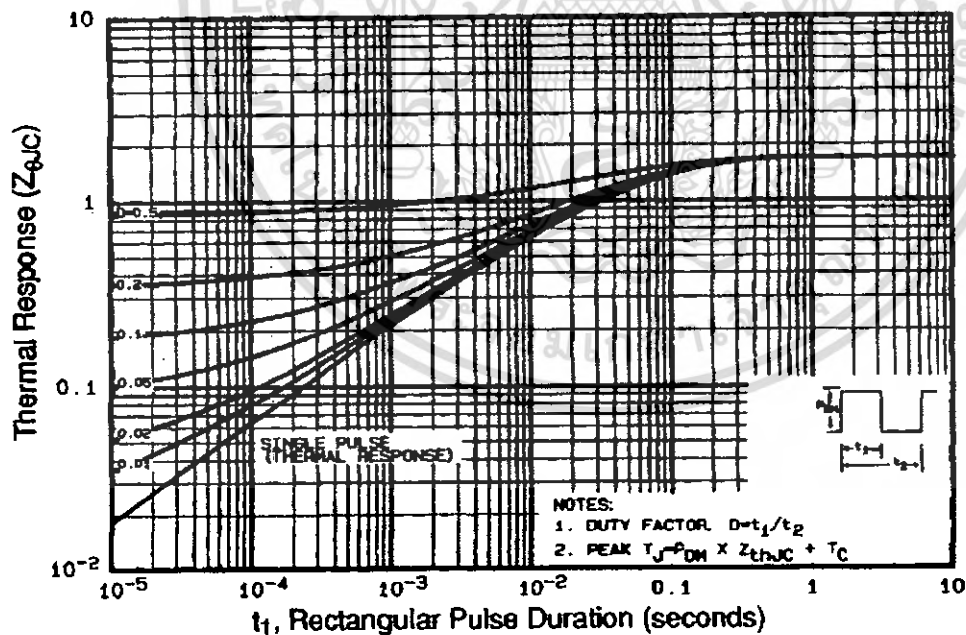
**Fig 9. Maximum Drain Current Vs. Case Temperature**



**Fig 10a. Switching Time Test Circuit**

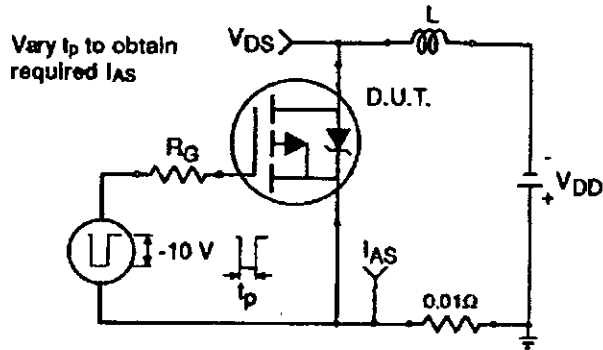


**Fig 10b. Switching Time Waveforms**

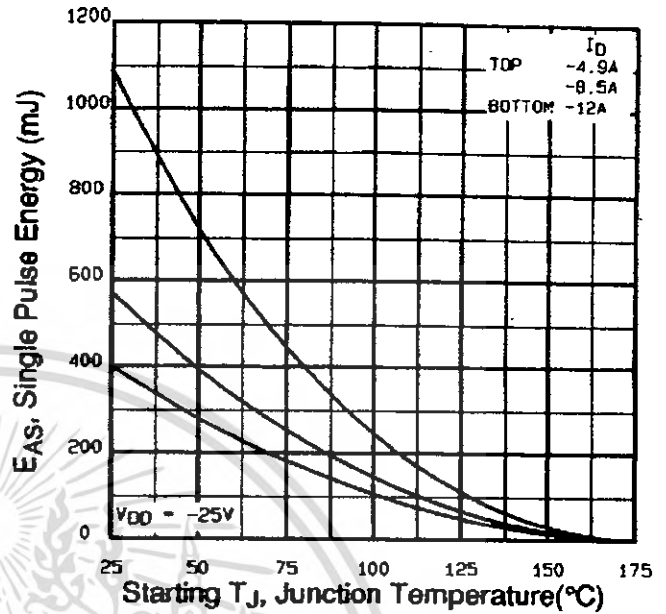


**Fig 11. Maximum Effective Transient Thermal Impedance, Junction-to-Case**

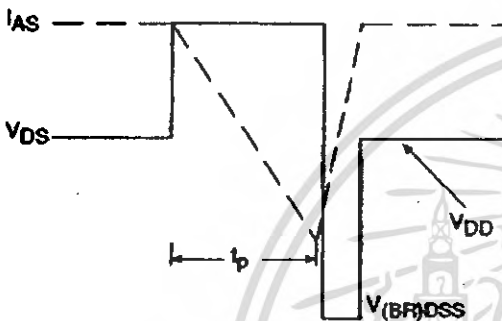
DATA SHEETS



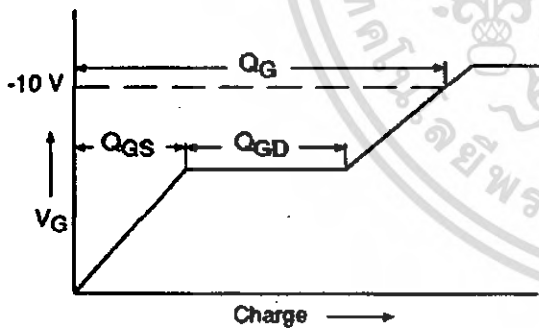
**Fig 12a. Unclamped Inductive Test Circuit**



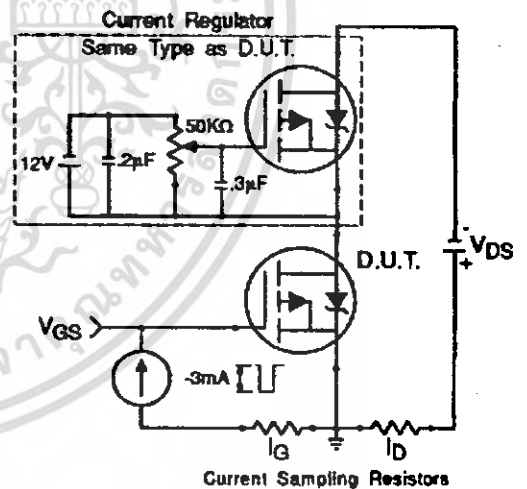
**Fig 12c. Maximum Avalanche Energy Vs. Drain Current**



**Fig 12b. Unclamped Inductive Waveforms**



**Fig 13a. Basic Gate Charge Waveform**



**Fig 13b. Gate Charge Test Circuit**

**Appendix A:** Figure 14, Peak Diode Recovery  $dv/dt$  Test Circuit – See page 1506

**Appendix B:** Package Outline Mechanical Drawing – See page 1509

**Appendix C:** Part Marking Information – See page 1516

**Appendix E:** Optional Leadforms – See page 1525

**International  
IR Rectifier**

This datasheet has been download from:

[www.datasheetarchive.com](http://www.datasheetarchive.com)

Datasheets for electronics components.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**IRF530/531/532/533**  
**IRFP130/131/132/133**

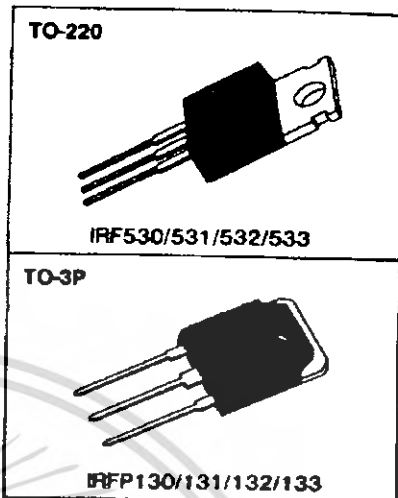
**N-CHANNEL**  
**POWER MOSFETS**

**FEATURES**

- Lower  $R_{DS(on)}$
- Improved inductive ruggedness
- Fast switching times
- Rugged polysilicon gate cell structure
- Lower input capacitance
- Extended safe operating area
- Improved high temperature reliability

**PRODUCT SUMMARY**

Part Number	$V_{DS}$	$R_{DS(on)}$	$I_D$
IRF530/IRFP130	100V	0.16 $\Omega$	14A
IRF531/IRFP131	80V	0.16 $\Omega$	14A
IRF532/IRFP132	100V	0.23 $\Omega$	12A
IRF533/IRFP133	80V	0.23 $\Omega$	12A



**MAXIMUM RATINGS**

Characteristics	Symbol	IRF530 IRFP130	IRF531 IRFP131	IRF532 IRFP132	IRF533 IRFP133	Unit
Drain-Source Voltage (1)	$V_{DS}$	100	80	100	80	Vdc
Drain-Gate Voltage ( $R_{GS}=1.0M\Omega$ )(1)	$V_{DGR}$	100	80	100	80	Vdc
Gate-Source Voltage	$V_{GS}$	$\pm 20$				Vdc
Continuous Drain Current $T_C=25^\circ C$	$I_D$	14	14	12	12	A dc
Continuous Drain Current $T_C=100^\circ C$	$I_D$	10	10	8.3	8.3	A dc
Drain Current—Pulsed (3)	$I_{DM}$	56	56	48	48	A dc
Gate Current—Pulsed	$I_{GM}$	$\pm 1.5$				A dc
Single Pulsed Avalanche Energy(4)	$E_{AS}$	69				mJ
Avalanche Current	$I_{AS}$	14				A
Total Power Dissipation @ $T_C=25^\circ C$ Derate above $25^\circ C$	$P_D$	77				Watts
		0.62				W/ $^\circ C$
Operating and Storage Junction to Case	$T_J, T_{stg}$	-55 to 150				$^\circ C$
Maximum Lead Temp. for Soldering Purposes, 1/8" from case for 5 seconds	$T_L$	300				$^\circ C$

Notes: (1)  $T_J=25^\circ C$  to  $150^\circ C$

(2) Pulse test: Pulse width  $< 300\mu s$ , Duty Cycle  $< 2\%$

(3) Repetitive rating: Pulse width limited by max. junction temperature


(4)  $L=0.53$  mH,  $V_{GS}=26V$ ,  $R_G=25\Omega$ , Starting  $T_J=25^\circ C$



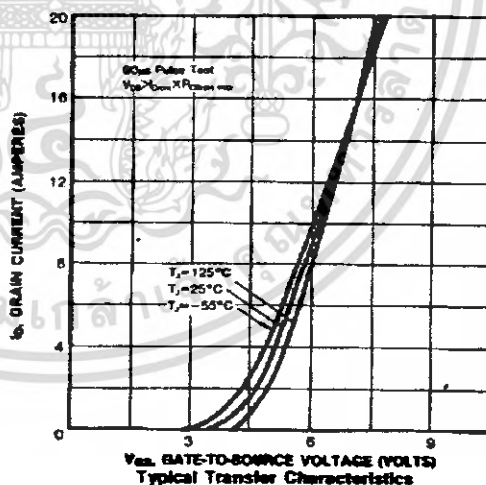
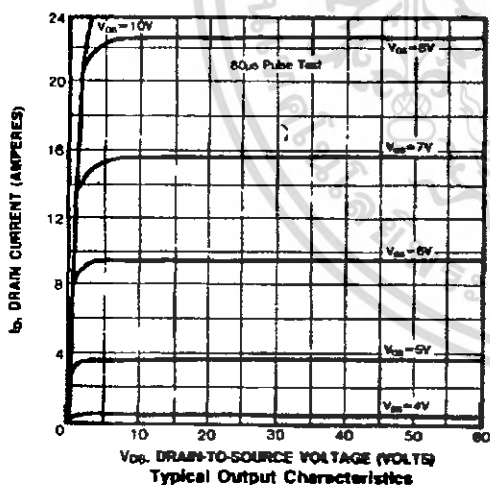
**IRF530/531/532/533**  
**IRFP130/131/132/133**

**N-CHANNEL**  
**POWER MOSFETS**

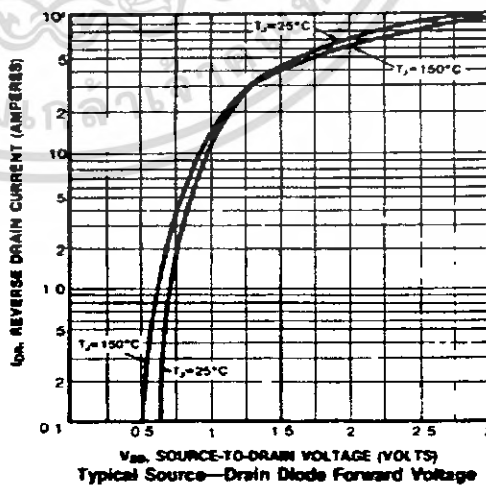
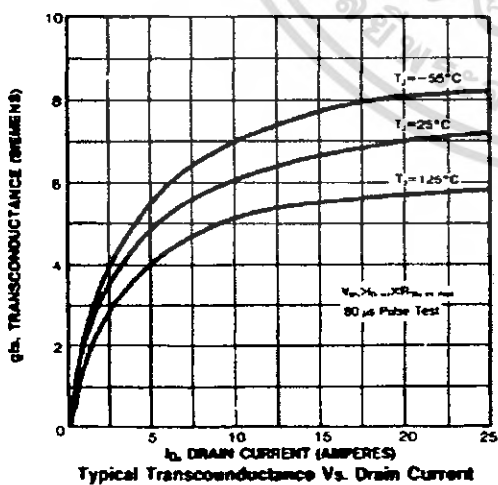
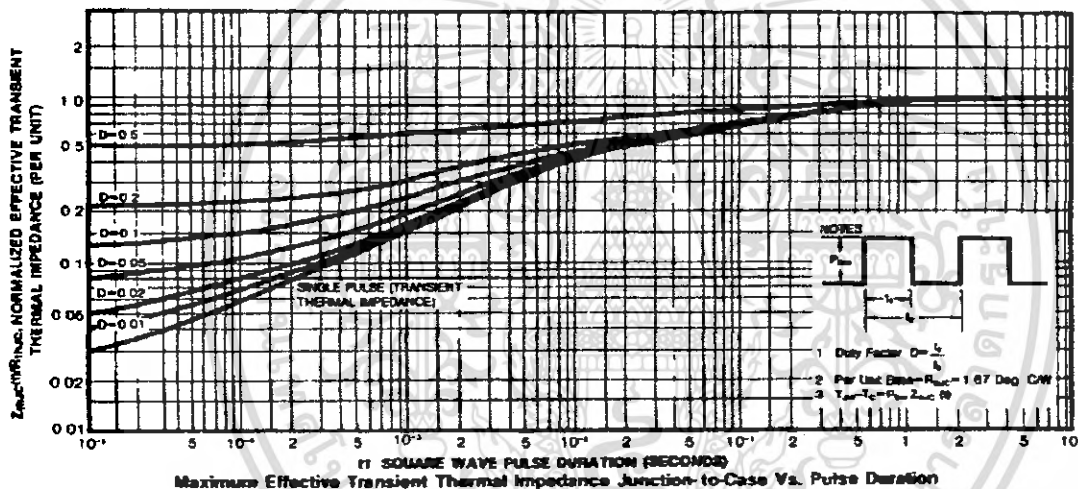
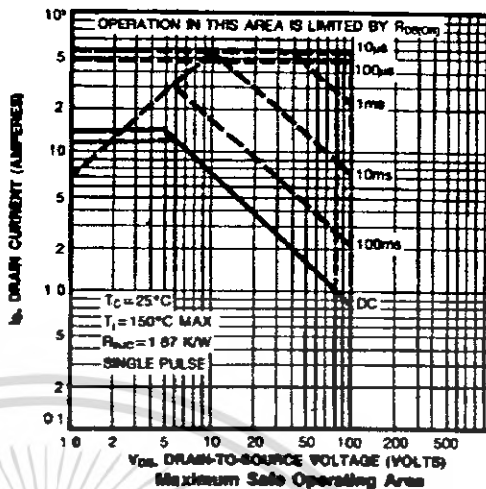
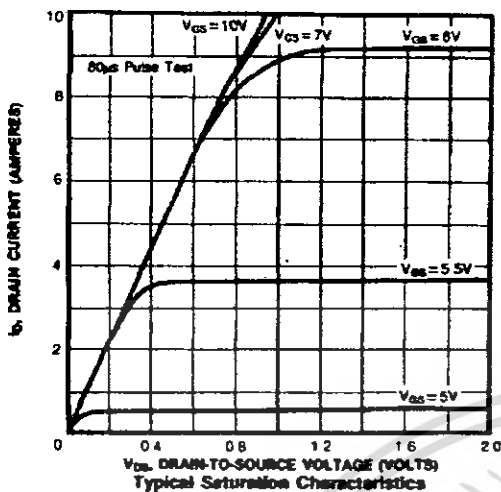
**SOURCE-DRAIN DIODE RATINGS AND CHARACTERISTICS**

Symbol	Characteristic	Min	Typ	Max	Units	Test Conditions
$I_S$	Continuous Source Current (Body Diode) IRF530/IRFP130 IRF531/IRFP131	—	—	14	A	Modified MOSFET symbol showing the integral reverse P-N junction rectifier 
	IRF532/IRFP132 IRF533/IRFP133	—	—	12	A	
$I_{SM}$	Pulse Source Current (Body Diode) (3) IRF530/IRFP130 IRF531/IRFP131	—	—	56	A	
	IRF532/IRFP132 IRF533/IRFP133	—	—	48	A	
$V_{SD}$	Diode Forward Voltage (2) IRF530/IRFP130 IRF531/IRFP131	—	—	2.5	V	$T_C=25^\circ\text{C}$ , $I_S=14\text{A}$ , $V_{GS}=0\text{V}$
	IRF532/IRFP132 IRF533/IRFP133	—	—	2.3	V	$T_C=25^\circ\text{C}$ , $I_S=12\text{A}$ , $V_{GS}=0\text{V}$
$t_{rr}$	Reverse Recovery Time	—	120	250	ns	$T_J=25^\circ\text{C}$ , $I_F=14\text{A}$ , $dI_F/dt=100\text{A}/\mu\text{s}$

Notes: (1)  $T_J=25^\circ\text{C}$  to  $150^\circ\text{C}$  (2) Pulse test: Pulse width < 300 $\mu\text{s}$ , Duty Cycle < 2%  
(3) Repetitive rating: Pulse with limited by max junction temperature



**N-CHANNEL  
 POWER MOSFETS**



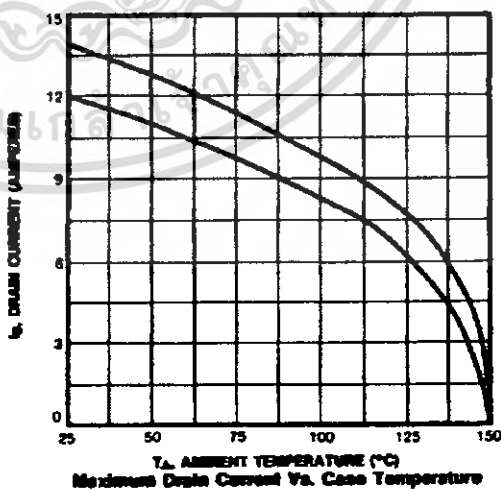
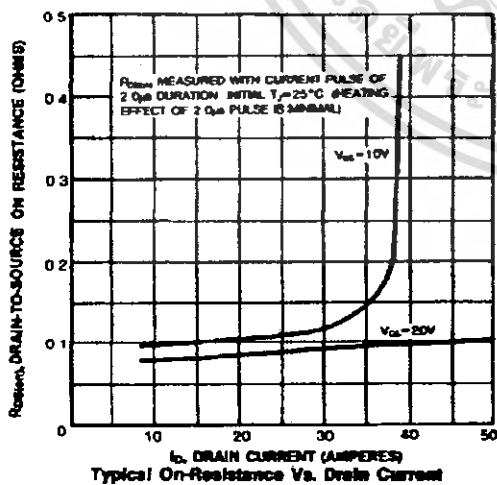
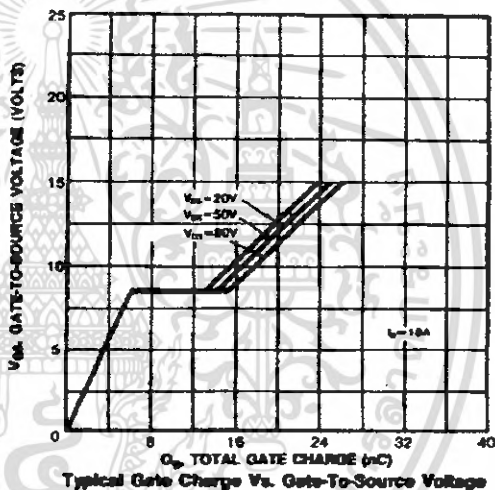
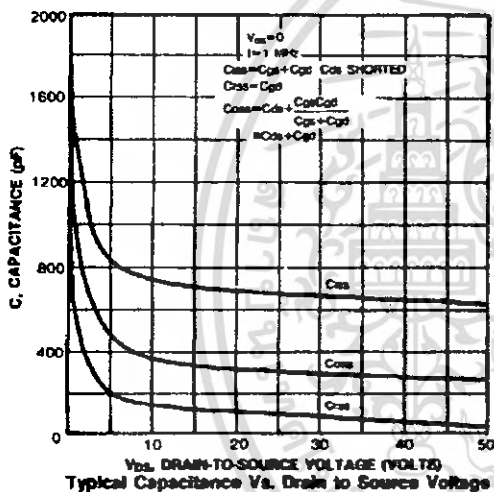
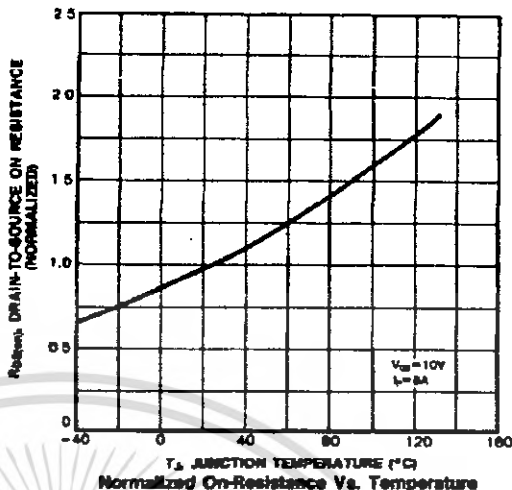
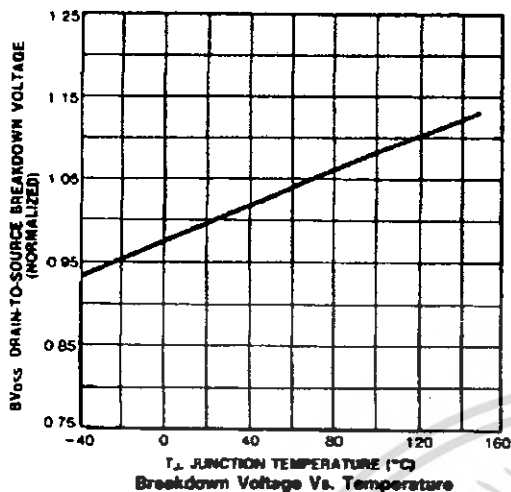
2



ELECTRONICS

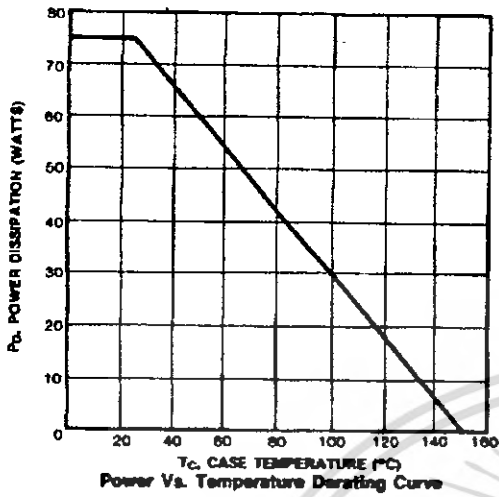
**IRF530/531/532/533**  
**IRFP130/131/132/133**

**N-CHANNEL**  
**POWER MOSFETS**



**IRF530/531/532/533**  
**IRFP130/131/132/133**

**N-CHANNEL**  
**POWER MOSFETS**



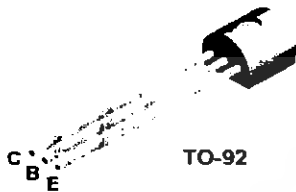
Power Vs. Temperature Derating Curve

2

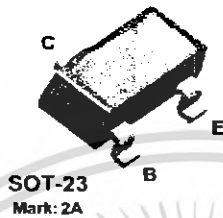




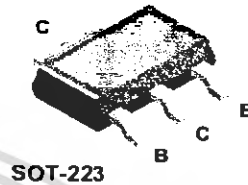
**2N3906**



**MMBT3906**



**PZT3906**



**PNP General Purpose Amplifier**

This device is designed for general purpose amplifier and switching applications at collector currents of 10  $\mu$ A to 100 mA.

**Absolute Maximum Ratings\***  $T_A = 25^\circ\text{C}$  unless otherwise noted

Symbol	Parameter	Value	Units
$V_{CE0}$	Collector-Emitter Voltage	40	V
$V_{CB0}$	Collector-Base Voltage	40	V
$V_{EB0}$	Emitter-Base Voltage	5.0	V
$I_C$	Collector Current - Continuous	200	mA
$T_J, T_{stg}$	Operating and Storage Junction Temperature Range	-55 to +150	$^\circ\text{C}$

\*These ratings are limiting values above which the serviceability of any semiconductor device may be impaired.

**NOTES:**

- 1) These ratings are based on a maximum junction temperature of 150 degrees C.
- 2) These are steady state limits. The factory should be consulted on applications involving pulsed or low duty cycle operations.
- 3) All voltages (V) and currents (A) are negative polarity for PNP transistors.

**Thermal Characteristics**  $T_A = 25^\circ\text{C}$  unless otherwise noted

Symbol	Characteristic	Max			Units
		2N3906	*MMBT3906	**PZT3906	
$P_D$	Total Device Dissipation	625	350	1,000	mW
	Derate above 25 $^\circ\text{C}$	5.0	2.8	8.0	mW/ $^\circ\text{C}$
$R_{\theta JC}$	Thermal Resistance, Junction to Case	83.3			$^\circ\text{C/W}$
$R_{\theta JA}$	Thermal Resistance, Junction to Ambient	200	357	125	$^\circ\text{C/W}$

\* Device mounted on FR-4 PCB 1.6" X 1.6" X 0.06."

\*\* Device mounted on FR-4 PCB 36 mm X 18 mm X 1.5 mm; mounting pad for the collector lead min. 6  $\text{cm}^2$ .

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## PNP General Purpose Amplifier

(continued)

2N3906 / MMBT3906 / PZT3906

### Electrical Characteristics

$T_A = 25^\circ\text{C}$  unless otherwise noted

Symbol	Parameter	Test Conditions	Min	Max	Units
--------	-----------	-----------------	-----	-----	-------

#### OFF CHARACTERISTICS

$V_{(BR)CEO}$	Collector-Emitter Breakdown Voltage*	$I_C = 1.0 \text{ mA}, I_B = 0$	40		V
$V_{(BR)CBO}$	Collector-Base Breakdown Voltage	$I_C = 10 \mu\text{A}, I_E = 0$	40		V
$V_{(BR)EBO}$	Emitter-Base Breakdown Voltage	$I_E = 10 \mu\text{A}, I_C = 0$	5.0		V
$I_{BL}$	Base Cutoff Current	$V_{CE} = 30 \text{ V}, V_{BE} = 3.0 \text{ V}$		50	nA
$I_{CEX}$	Collector Cutoff Current	$V_{CE} = 30 \text{ V}, V_{BE} = 3.0 \text{ V}$		50	nA

#### ON CHARACTERISTICS

$h_{FE}$	DC Current Gain *	$I_C = 0.1 \text{ mA}, V_{CE} = 1.0 \text{ V}$ $I_C = 1.0 \text{ mA}, V_{CE} = 1.0 \text{ V}$ $I_C = 10 \text{ mA}, V_{CE} = 1.0 \text{ V}$ $I_C = 50 \text{ mA}, V_{CE} = 1.0 \text{ V}$ $I_C = 100 \text{ mA}, V_{CE} = 1.0 \text{ V}$	60 80 100 60 30	300	
$V_{CE(sat)}$	Collector-Emitter Saturation Voltage	$I_C = 10 \text{ mA}, I_B = 1.0 \text{ mA}$ $I_C = 50 \text{ mA}, I_B = 5.0 \text{ mA}$		0.25 0.4	V V
$V_{BE(sat)}$	Base-Emitter Saturation Voltage	$I_C = 10 \text{ mA}, I_B = 1.0 \text{ mA}$ $I_C = 50 \text{ mA}, I_B = 5.0 \text{ mA}$	0.65	0.85 0.95	V V

#### SMALL SIGNAL CHARACTERISTICS

$f_T$	Current Gain - Bandwidth Product	$I_C = 10 \text{ mA}, V_{CE} = 20 \text{ V},$ $f = 100 \text{ MHz}$	250		MHz
$C_{obo}$	Output Capacitance	$V_{CB} = 5.0 \text{ V}, I_E = 0,$ $f = 100 \text{ kHz}$		4.5	pF
$C_{ibo}$	Input Capacitance	$V_{EB} = 0.5 \text{ V}, I_C = 0,$ $f = 100 \text{ kHz}$		10.0	pF
NF	Noise Figure	$I_C = 100 \mu\text{A}, V_{CE} = 5.0 \text{ V},$ $R_S = 1.0 \text{ k}\Omega, f = 10 \text{ Hz to } 15.7 \text{ kHz}$		4.0	dB

#### SWITCHING CHARACTERISTICS

$t_d$	Delay Time	$V_{CC} = 3.0 \text{ V}, V_{BE} = 0.5 \text{ V},$		35	ns
$t_r$	Rise Time	$I_C = 10 \text{ mA}, I_{B1} = 1.0 \text{ mA}$		35	ns
$t_s$	Storage Time	$V_{CC} = 3.0 \text{ V}, I_C = 10 \text{ mA}$		225	ns
$t_f$	Fall Time	$I_{B1} = I_{B2} = 1.0 \text{ mA}$		75	ns

\*Pulse Test: Pulse Width  $\leq 300 \mu\text{s}$ , Duty Cycle  $\leq 2.0\%$

NOTE: All voltages (V) and currents (A) are negative polarity for PNP transistors.

### Spice Model

PNP (Is=1.41f Xti=3 Eg=1.11 Vaf=18.7 Bf=180.7 Ne=1.5 Ise=0 Ikf=80m Xtb=1.5 Br=4.977 Nc=2 Isc=0 Ikr=0 Rc=2.5 Cjc=9.728p Mjc=.5776 Vjc=.75 Fc=.5 Cje=8.063p Mje=.3677 Vje=.75 Tr=33.42n Tf=179.3p Iif=.4 Vit=4 Xtf=6 Rb=10)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

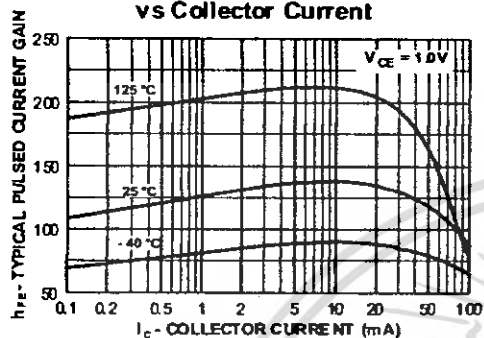
# PNP General Purpose Amplifier

(continued)

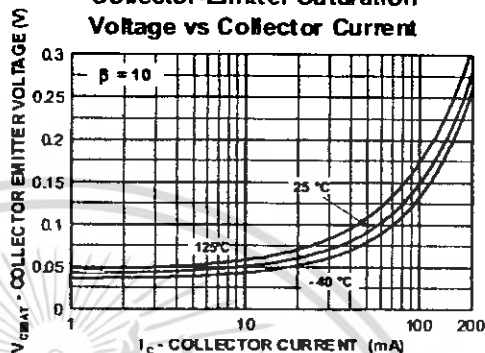
2N3906 / MMBT3906 / PZT3906

## Typical Characteristics

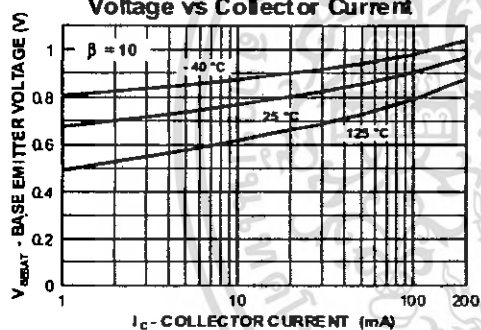
Typical Pulsed Current Gain vs Collector Current



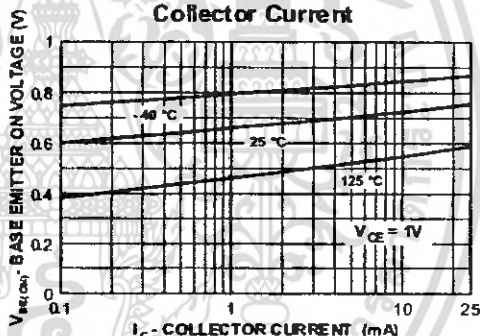
Collector-Emitter Saturation Voltage vs Collector Current



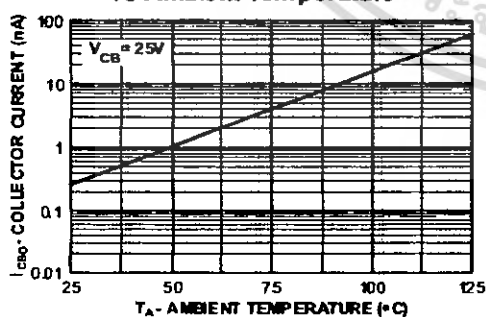
Base-Emitter Saturation Voltage vs Collector Current



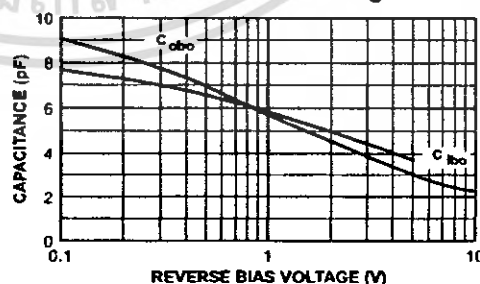
Base Emitter ON Voltage vs Collector Current



Collector-Cutoff Current vs Ambient Temperature



Common-Base Open Circuit Input and Output Capacitance vs Reverse Bias Voltage



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

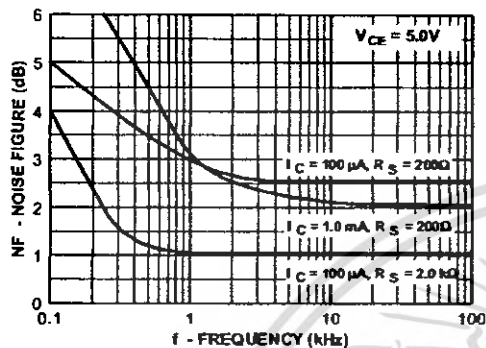
# PNP General Purpose Amplifier

(continued)

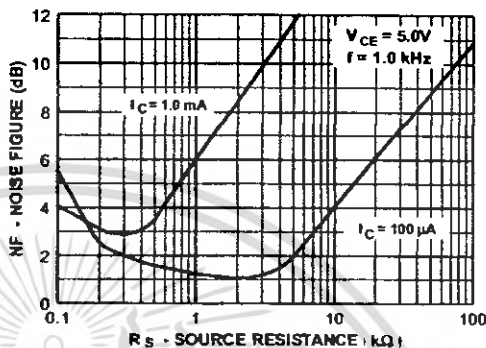
2N3906 / MMBT3906 / PZT3906

## Typical Characteristics (continued)

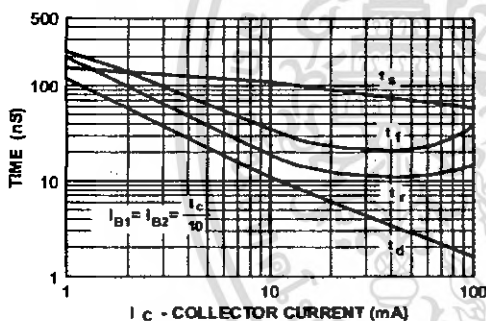
Noise Figure vs Frequency



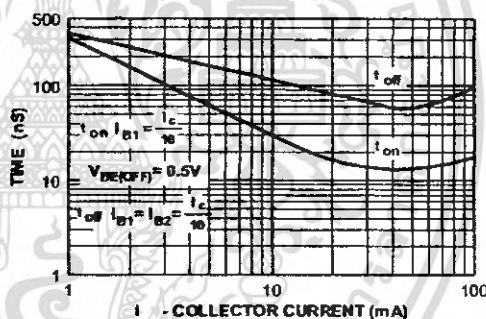
Noise Figure vs Source Resistance



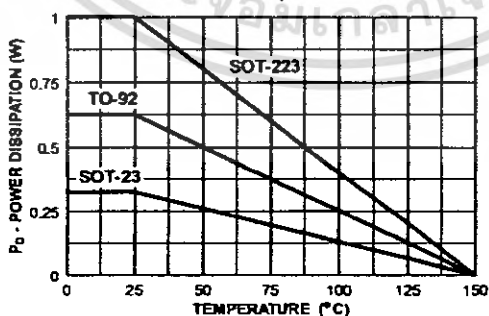
Switching Times vs Collector Current



Turn On and Turn Off Times vs Collector Current



Power Dissipation vs Ambient Temperature

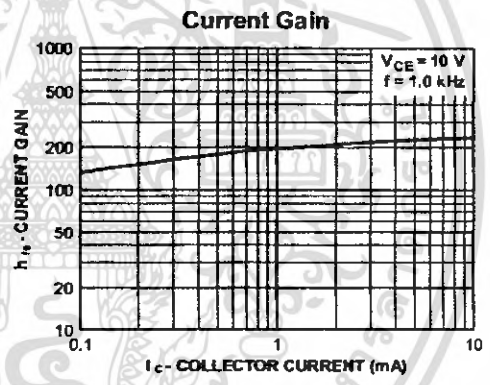
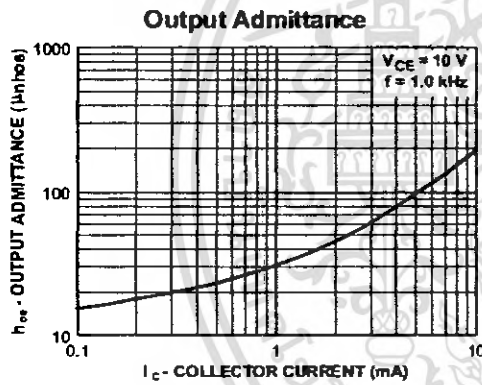
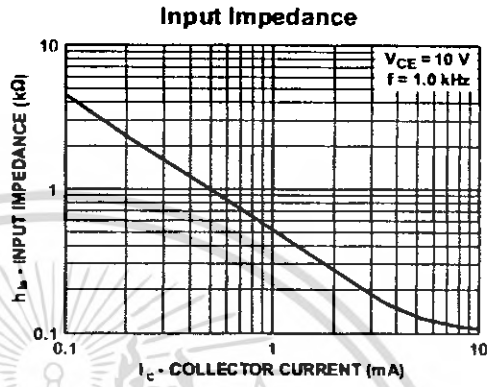
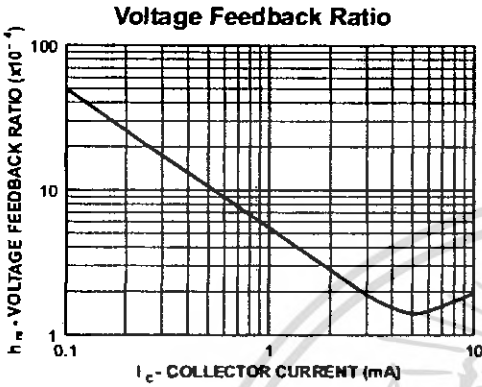


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PNP General Purpose Amplifier  
(continued)

2N3906 / MMBT3906 / PZT3906

Typical Characteristics (continued)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## TRADEMARKS

The following are registered and unregistered trademarks Fairchild Semiconductor owns or is authorized to use and is not intended to be an exhaustive list of all such trademarks.

ACE <sup>x</sup> ™	FAST <sup>r</sup> ™	PowerTrench®	SyncFET™
Bottomless™	GlobalOptoisolator™	QFET™	TinyLogic™
CoolFET™	GTO™	QS™	UHC™
CROSSVOLT™	HiSeC™	QT Optoelectronics™	VCX™
DOME™	ISOPLANAR™	Quiet Series™	
E <sup>2</sup> CMOS™	MICROWIRE™	SILENT SWITCHER®	
EnSigna™	OPTOLOGIC™	SMART START™	
FACT™	OPTOPLANAR™	SuperSOT™-3	
FACT Quiet Series™	PACMAN™	SuperSOT™-6	
FAST®	POP™	SuperSOT™-8	

## DISCLAIMER

FAIRCHILD SEMICONDUCTOR RESERVES THE RIGHT TO MAKE CHANGES WITHOUT FURTHER NOTICE TO ANY PRODUCTS HEREIN TO IMPROVE RELIABILITY, FUNCTION OR DESIGN. FAIRCHILD DOES NOT ASSUME ANY LIABILITY ARISING OUT OF THE APPLICATION OR USE OF ANY PRODUCT OR CIRCUIT DESCRIBED HEREIN; NEITHER DOES IT CONVEY ANY LICENSE UNDER ITS PATENT RIGHTS, NOR THE RIGHTS OF OTHERS.

## LIFE SUPPORT POLICY

FAIRCHILD'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF FAIRCHILD SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, or (c) whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in significant injury to the user.

2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

## PRODUCT STATUS DEFINITIONS

### Definition of Terms

Datasheet Identification	Product Status	Definition
Advance Information	Formative or In Design	This datasheet contains the design specifications for product development. Specifications may change in any manner without notice.
Preliminary	First Production	This datasheet contains preliminary data, and supplementary data will be published at a later date. Fairchild Semiconductor reserves the right to make changes at any time without notice in order to improve design.
No Identification Needed	Full Production	This datasheet contains final specifications. Fairchild Semiconductor reserves the right to make changes at any time without notice in order to improve design.
Obsolete	Not In Production	This datasheet contains specifications on a product that has been discontinued by Fairchild semiconductor. The datasheet is printed for reference information only.

Rev. G

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้