

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

หุ่นยนต์ลาดตระเวนระยะไกล

Far Away Reconnoiter Robot



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ไปเผยแพร่ใช้ในการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผ่านการตรวจรับงานแล้ว

(ลงชื่อ).....ผู้ตรวจ

ผ่านการตรวจรูปเล่มแล้ว

(ลงชื่อ).....ผู้ตรวจ

หุ่นยนต์ลาดตระเวนระยะไกล

Far Away Reconnoiter Robot

โดย

นายภพสุข สุขสำราญ 46015064

นายนวัตกรณ์ ไก่แก้ว 46015057

นายธานิน วัฒนประดิษฐ์ 46015053

อาจารย์ที่ปรึกษา

รศ.สมยศ จุณณะปิยะ

อ.ศรวัฒน์ ชิวปรีชา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2548

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง หุ่นยนต์ลาดตระเวนระยะไกล


Far Away Reconnoiter Robot : FARR

ผู้จัดทำ

1. นายธานิน วัฒนประดิษฐ์ 46015053
2. นายณวัตรณ์ ไก่แก้ว 46015057
3. นายภพสุข สุขสำราญ 46015064

  
.....  
( รศ.สมยศ จุณณะปิยะ )

อาจารย์ที่ปรึกษา

  
.....  
(อาจารย์ศรวัฒน์ ชิวปรีชา)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## หุ่นยนต์ลาดตระเวนระยะไกล

### Far Away Reconnoiter Robot : FARR

โดย นายธานี วัฒนประดิษฐ์	46015053
นายนวัตรณ ใก่อแก้ว	46015057
นายภพสุข สุขสำราญ	46015064

อาจารย์ที่ปรึกษา รศ.สมยศ จุณณะปิยะ

อาจารย์ศรีวัฒน์ ชิวปรีชา

### บทคัดย่อ

ในทางทหาร (A military) การลาดตระเวนเป็นภารกิจที่สำคัญอย่างยิ่ง และเป็นส่วนหนึ่งที่สำคัญในการที่จะพาไปซึ่งความสูญเสียน้อยที่สุดของกองทัพ และเราไม่สามารถที่จะหลีกเลี่ยงได้เลยในพื้นที่ที่เต็มไปด้วยอันตราย เช่น กับระเบิด หรือทหารลาดตระเวนฝ่ายตรงข้ามที่พร้อมเสมอที่จะกำจัดเรา ด้วยเหตุนี้เอง เราจึงมีการสร้างระบบที่ใช้ในการลาดตระเวน ใช้เป็นหูเป็นตาแทนเหล่าทหารหาญ เพื่อปกป้องชีวิตของเขาให้รอดพ้นจากอันตรายต่างๆ ในระหว่างการลาดตระเวน

Far Away Reconnoiter Robot (FARR) หรือ “หุ่นยนต์ลาดตระเวนระยะไกล” มีคุณสมบัติในการตรวจสอบพื้นที่ในสมรภูมิ ตรวจสอบข่าวศึก แล้วส่งข้อมูลเหล่านั้นมายังฐานปฏิบัติการในเขตพื้นที่นั้นๆ ด้วยข้อมูลภาพ ข้อมูลเสียง พื้นที่สนามรบ เราสามารถนำข้อมูลต่างๆ เหล่านั้นมาวางแผนในการจัดวางกองกำลังของเราได้อย่างถูกต้องแม่นยำมากขึ้น โดยไม่ต้องเสี่ยงกับชีวิตของหน่วยลาดตระเวนนอกจากนี้ FARR ยังสามารถปีนป่ายในพื้นที่ขรุขระได้ หรือแม้แต่ขึ้นหรือลงบันไดก็สามารถทำได้ ด้วยความสามารถของระบบขับเคลื่อนแบบตีนตะขาบ นั้นหมายถึงไม่เพียงแต่ในสนามรบเท่านั้นแต่ FARR ยังสามารถปฏิบัติภารกิจช่วยงานหน่วยรบพิเศษต่างๆ ในการช่วยเหลือตัวประกันตามสถานที่ต่างๆ ได้ด้วย

### Abstract

For military, surveillances are one of the important duties which need to be performing to reduce forces damage. Especially the insecure area such as bomb filed or in the opposite area which cannot be ignore in the defense. Therefore the system which can be use in the military surveillance was developed to help our soldier save their life and reduce injury.

Far Away Reconnoiter Robot (FARR) or surveillance robotic have the feature which can be use in surveillances and observe enemies movement and send information to the closest center in that area in the form of image or sound. We can then use such information for more effective defense plan with out risking our soldier life, because of the ability to proceed even in rough surface of the endless belt of a caterpillar tractor allow the FARR to proceed even up or down steps. These qualities make FARR very useful and adaptable to the special combat unit mission to rescue hostage life.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ

<b>บทที่ 1 บทนำ</b>	<b>1</b>
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์ของโครงการ	1
1.3 ประโยชน์อื่นๆ ที่ได้จากโครงการ	1
<b>บทที่ 2 ทฤษฎีและหลักการ</b>	<b>2</b>
2.1 ส่วนของตัวหุ่นยนต์	2
2.1.1 การทำงานในส่วนของตัวหุ่นยนต์	2
2.2 ส่วนของสถานีควบคุม	3
2.2.1 การทำงานในส่วนของสถานีควบคุม	3
2.3 ทฤษฎีที่เกี่ยวข้องและอุปกรณ์ที่นำมาใช้งาน	4
2.3.1 BIM2-433-64 Radiometrix RF Module	4
2.3.1.1 คุณสมบัติทั่วไปของ BIM2-433-64	5
2.3.1.2 รายละเอียดหน้าที่ของขาต่างๆ	5
2.3.2 TA7279P/AP DC Motor Rotation Control	6
2.3.2.1 คุณสมบัติทั่วไปของ TA7279P/AP	7
2.3.3 การควบคุมทิศทางการหมุนของมอเตอร์กระแสตรงด้วยทรานซิสเตอร์	8
2.3.4 การควบคุมทิศทางการหมุนของมอเตอร์กระแสตรงด้วยรีเลย์	10
2.3.5 ไมโครคอนโทรลเลอร์ MCS-51	12
2.3.5.1 8051 เป็นไมโครคอนโทรลเลอร์ขนาด 40 ขา ซึ่งมีขาต่าง ๆ ดังนี้	12
2.3.5.2 วงจรคัลลอคของ MCS-51	15
2.3.5.3 พอร์ทสื่อสารอนุกรม	16
2.3.5.4 SM0, SM1 บิตเลือกโหมดการทำงาน	17
<b>บทที่ 3 การคำนวณและการสร้าง</b>	<b>19</b>
3.1 การคำนวณออกแบบ	19
3.1.1 แหล่งจ่ายแรงดัน	19
3.1.1.1 การทำงานของวงจร	20
3.1.2 วงจรควบคุม	20
3.1.2.1 การทำงานของวงจร	20
3.1.2.1.1 Main CPU	21
3.1.2.1.2 Motor Control	21
3.1.2.1.3 Arm Control	21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2.1.4 Light Control	22
3.1.2.1.5 RF Module	22
3.1.3 Software	22
3.1.3.1 Flow Chart การทำงานของโปรแกรม	23
3.1.3.2 อธิบายการทำงานของโปรแกรมจาก Flow Chart	24
3.1.4 RF Module ในส่วนของสถานีควบคุม	24
3.1.4.1 การทำงานของวงจร	24
3.1.5 Software ที่ใช้ในการควบคุมการส่งข้อมูลที่ตัวของสถานีส่ง	25
<b>บทที่ 4 การทดลองและผลการทดลอง</b>	26
4.1 การทดลองการส่งข้อมูล	26
4.2 การทดลองส่งข้อมูลภาพและเสียง	35
4.3 การทดลองวัดค่ากระแสในขณะที่ทำการขับเคลื่อนหุ่นยนต์	38
<b>บทที่ 5 บทวิจารณ์และบทสรุป</b>	40
5.1 ปัญหาที่พบ	40
5.1.1 โครงสร้างของหุ่นยนต์	40
5.1.2 ความร้อนที่เกิดขึ้นกับแหล่งจ่ายไฟ	40
5.1.3 ความร้อนที่ตัวไอซีขับเคลื่อนมอเตอร์	40
5.1.4 สัญญาณจากคอมพิวเตอร์	40
5.1.5 การเชื่อมต่อกับ Pocket PC	41
5.1.6 สัญญาณรบกวนที่ไม่ทราบสาเหตุ	41
5.2 สรุปผล	41
<b>ภาคผนวก</b>	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูปภาพ

รูปที่ 2.1 ส่วนประกอบหลักภายในตัวหุ่นยนต์	2
รูปที่ 2.2 ส่วนประกอบหลักภายในสถานีควบคุม	3
รูปที่ 2.3 รูปร่างหน้าตาของ BIM2-433-64 และเสาอากาศที่ใช้งาน	4
รูปที่ 2.4 การต่อใช้งานของ BIM2-433-64	6
รูปที่ 2.5 รายละเอียดขาใช้งานต่างๆ และโครงสร้างภายใน	6
รูปที่ 2.6 รูปร่างหน้าตาของไอซีขับมอเตอร์กระแสตรงเบอร์ TA7279P/AP	7
รูปที่ 2.7 การต่อใช้งานของ TA7279P/AP	8
รูปที่ 2.8 การควบคุมทิศทางการหมุนของมอเตอร์โดยใช้ทรานซิสเตอร์ 2 ตัว	9
รูปที่ 2.9 การควบคุมทิศทางการหมุนของมอเตอร์โดยใช้ทรานซิสเตอร์แบบ NPN เป็นรูปตัว “H”	9
รูปที่ 2.10 การใช้รีเลย์ควบคุมการปิด-เปิดมอเตอร์โดยใช้สัญญาณดิจิทัล	10
รูปที่ 2.11 การใช้รีเลย์ควบคุมทิศทางการหมุนมอเตอร์โดยใช้สัญญาณดิจิทัล	11
รูปที่ 2.12 การจัดวางขาของ 8051	12
รูปที่ 2.13 แสดงโครงสร้าง พอร์ต 0 (บิต)	13
รูปที่ 2.14 โครงสร้างของพอร์ต 1 (บิต)	13
รูปที่ 2.15 โครงสร้างของ พอร์ต 2 (บิต)	14
รูปที่ 2.16 การต่อขารีเลย์ให้กับ 8051	14
รูปที่ 2.17 วงจรสร้างคล็อกของ 8051	15
รูปที่ 3.1 วงจรแหล่งจ่ายแรงดัน	19
รูปที่ 3.2 วงจรสมบรูณ์ของตัวหุ่นยนต์	20
รูปที่ 3.3 Flow Chart การทำงานของโปรแกรมภายในตัวไมโครคอนโทรลเลอร์ MCS-51	23
รูปที่ 3.4 วงจรในส่วนของ RF Module ที่สถานีควบคุม	24
รูปที่ 3.5 โปรแกรมที่ใช้ interface กับคอมพิวเตอร์	25
รูปที่ 4.1 สัญญาณที่วัดได้จากขา 9 ของ MAX232 เมื่อกดตัวอักษร “ w ” บนแป้นคีย์บอร์ด	26
รูปที่ 4.2 สัญญาณที่วัดได้จากขา 9 ของ MAX232 เมื่อกดตัวอักษร “ s ” บนแป้นคีย์บอร์ด	27
รูปที่ 4.3 สัญญาณที่วัดได้จากขา 9 ของ MAX232 เมื่อกดตัวอักษร “ a ” บนแป้นคีย์บอร์ด	27
รูปที่ 4.4 สัญญาณที่วัดได้จากขา 9 ของ MAX232 เมื่อกดตัวอักษร “ d ” บนแป้นคีย์บอร์ด	28
รูปที่ 4.5 สัญญาณที่วัดได้จากขา 9 ของ MAX232 เมื่อกดตัวอักษร “ i ” บนแป้นคีย์บอร์ด	28
รูปที่ 4.6 สัญญาณที่วัดได้จากขา 9 ของ MAX232 เมื่อกดตัวอักษร “ k ” บนแป้นคีย์บอร์ด	29
รูปที่ 4.7 สัญญาณที่วัดได้จากขา 9 ของ MAX232 เมื่อกดตัวอักษร “ l ” บนแป้นคีย์บอร์ด	29
รูปที่ 4.8 สัญญาณที่วัดได้ที่ตัวหุ่นยนต์จากขา 10 ของ MCS-51 เมื่อกดตัวอักษร “ w ” บนแป้นคีย์บอร์ด	30

รูปที่ 4.9 สัญญาณที่วัดได้ที่ตัวหุ่นยนต์จากขา 10 ของ MCS-51 เมื่อกดตัวอักษร “ s ” บนแป้นคีย์บอร์ด	31
รูปที่ 4.10 สัญญาณที่วัดได้ที่ตัวหุ่นยนต์จากขา 10 ของ MCS-51 เมื่อกดตัวอักษร “ a ” บนแป้นคีย์บอร์ด	31
รูปที่ 4.11 สัญญาณที่วัดได้ที่ตัวหุ่นยนต์จากขา 10 ของ MCS-51 เมื่อกดตัวอักษร “ d ” บนแป้นคีย์บอร์ด	32
รูปที่ 4.12 สัญญาณที่วัดได้ที่ตัวหุ่นยนต์จากขา 10 ของ MCS-51 เมื่อกดตัวอักษร “ i ” บนแป้นคีย์บอร์ด	32
รูปที่ 4.13 สัญญาณที่วัดได้ที่ตัวหุ่นยนต์จากขา 10 ของ MCS-51 เมื่อกดตัวอักษร “ k ” บนแป้นคีย์บอร์ด	33
รูปที่ 4.14 สัญญาณที่วัดได้ที่ตัวหุ่นยนต์จากขา 10 ของ MCS-51 เมื่อกดตัวอักษร “ l ” บนแป้นคีย์บอร์ด	33
รูปที่ 4.15 การวัดสัญญาณที่รับได้ที่ตัวหุ่นยนต์ผ่าน RF Module	34
รูปที่ 4.16 Spectrum ที่วัดได้โดย Signal Analysis จากการส่งสัญญาณที่ตัว RF Module	34
รูปที่ 4.17 การวัด Spectrum ในรูปที่ 4.16	35
รูปที่ 4.18 Spectrum ที่จับได้จากชุดส่งสัญญาณภาพและเสียง โดยเครื่อง Signal Analysis	36
รูปที่ 4.19 สัญญาณภาพที่รับได้ที่หน้าจอคอมพิวเตอร์	36
รูปที่ 4.20 ชุดอุปกรณ์ Video Sender	37
รูปที่ 4.21 อุปกรณ์ที่ใช้ในการทดลองวัดสัญญาณ	37
รูปที่ 4.22 แสดงการต่อวงจรวัดค่ากระแสที่ตัวมอเตอร์ขับเคลื่อนขณะใช้งานจริง	38
รูปที่ 4.23 ผลการทดลองจากวงจรรูปที่ 4.8	39
รูปที่ 5.1 รูปร่างหน้าตาของเจ้า Pack Bot	41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญตาราง

ตารางที่ 2.1 ตารางแสดงสถานะของขา TX และ RX	5
ตารางที่ 2.2 รายละเอียดหน้าที่ของขาต่างๆ	7
ตารางที่ 2.3 ฟังก์ชันการทำงานของสัญญาณควบคุม	8
ตารางที่ 2.4 ตารางการใช้ ไทม์เมอร์ 1 กำหนด บอดเรท	18
ตารางที่ 4.1 ค่าของตัวอักษรในรูปแบบของเลขฐานต่างๆ	30



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1

## บทนำ

### 1.1 ความสำคัญและที่มา

ในปัจจุบันเทคโนโลยีได้ก้าวเข้ามาแทนที่ในเกือบจะทุกๆ บทบาทหน้าที่ที่เกี่ยวข้องกับชีวิตประจำวันของเรา หรือแม้แต่ในทางทหาร(A Military) ซึ่งได้มีการนำเอาเทคโนโลยีต่างๆ ที่มีความทันสมัย เข้ามาช่วยในภารกิจที่เกี่ยวข้อง และสำหรับเหล่าทหารในหน่วยลาดตระเวนซึ่งจัดได้ว่าเป็นหน่วยที่มีขีดความสามารถมากที่สุด แต่ในทางกลับกัน ก็มีอันตรายมากที่สุดเช่นกัน

Far Away Reconnoiter Robot หรือชื่อย่อที่เราเรียกว่า “FARR” หรือภาษาไทยเราเรียกว่า “หุ่นยนต์ลาดตระเวนระยะไกล” มีคุณสมบัติในการตรวจสอบพื้นที่ในสมรภูมิตตรวจสอบข้าศึก แล้วส่งข้อมูลเหล่านั้นมายังฐานปฏิบัติการในเขตพื้นที่นั้นๆ ด้วยข้อมูลภาพ ข้อมูลเสียง พื้นที่สนามรบ เราสามารถนำข้อมูลต่างๆ เหล่านี้มาวางแผนในการจัดวางกองกำลังของเราได้อย่างถูกต้องแม่นยำมากขึ้น โดยไม่ต้องเสี่ยงกับชีวิตของหน่วยลาดตระเวน นอกจากนี้ FARR ยังสามารถปีนป่ายในพื้นที่ขรุขระได้ หรือแม้แต่ขึ้นหรือลงบันไดก็สามารถทำได้ ด้วยความสามารถของระบบขับเคลื่อนแบบตีนตะขาบ นั้นหมายถึงไม่เพียงแค่นั้นแต่ FARR ยังสามารถปฏิบัติภารกิจช่วยงานหน่วยรบพิเศษต่างๆ ในการช่วยเหลือตัวประกันตามสถานที่ต่างๆ ได้อีกด้วย

### 1.2 วัตถุประสงค์ของโครงการ

- 1.2.1 เพื่อใช้เป็นหุ่นยนต์ต้นแบบในการพัฒนาไปสู่ระบบช่วยในการลาดตระเวนในกิจการทหาร
- 1.2.2 เพื่อศึกษาการทำงานของระบบไมโครคอนโทรลเลอร์
- 1.2.3 เพื่อศึกษาการนำไมโครคอนโทรลเลอร์มาใช้ในการอินเตอร์เฟสกับระบบต่างๆ
- 1.2.4 เพื่อศึกษาการนำเอา Application ต่างๆ ของระบบ Wireless มา ประยุกต์ ใช้งาน
- 1.2.5 ฝึกทักษะในเรื่องของการออกแบบวงจรต่างๆ

### 1.3 ประโยชน์อื่นๆ ที่ได้จากโครงการ

- นำไปพัฒนาต่อยอดในเรื่องของเทคโนโลยีการระบุตำแหน่งของตัวหุ่นยนต์
- ติดตั้งอุปกรณ์ต่างๆ เพิ่มเติมได้ง่าย อาทิเช่น อุปกรณ์ตรวจจับระเบิด
- ศึกษาการใช้งานฟังก์ชันต่างๆ ของเครื่องมือวัดและทดสอบ
- เพิ่มทักษะในเรื่องของการแก้ไขปัญหาเฉพาะหน้า ในกรณีที่วงจรไม่ทำงาน หรือเพื่อเพิ่มประสิทธิภาพของวงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

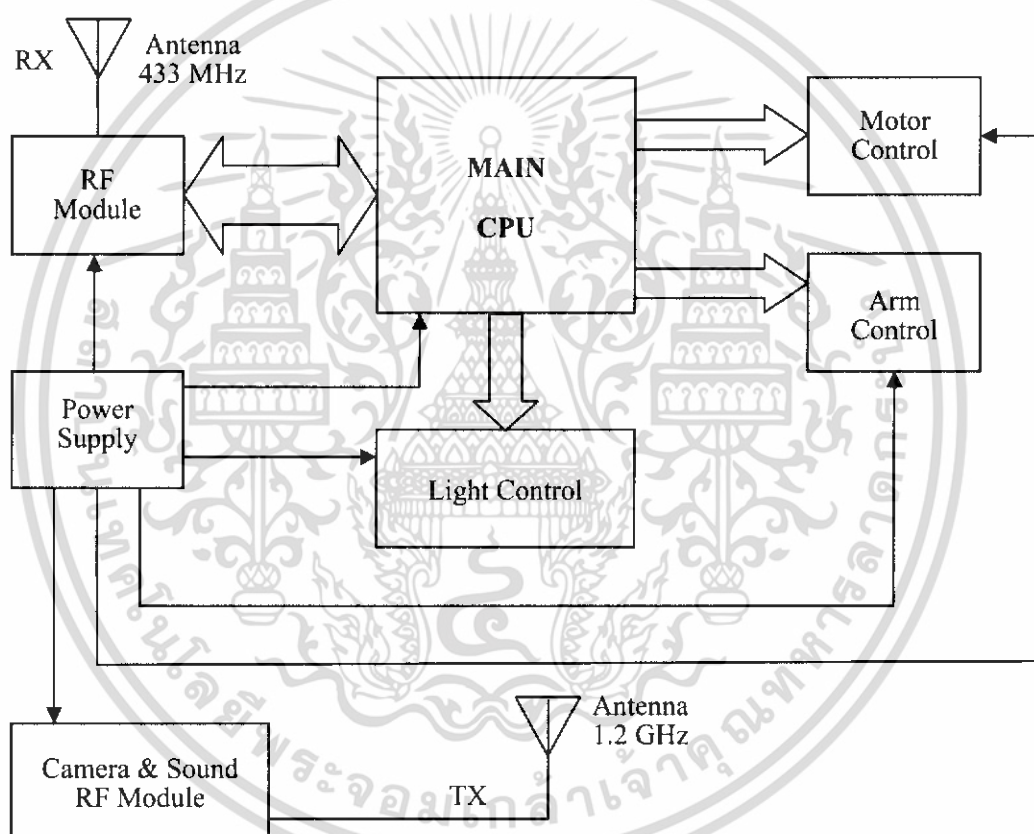
## บทที่ 2

### ทฤษฎีและหลักการ

เราทำการแบ่งการทำงานออกเป็น 2 ส่วนใหญ่ๆ ด้วยกัน คือ

1. ส่วนของตัวหุ่นยนต์
2. ส่วนของสถานีควบคุม

#### 2.1 ส่วนของตัวหุ่นยนต์



รูปที่ 2.1 ส่วนประกอบหลักภายในตัวหุ่นยนต์

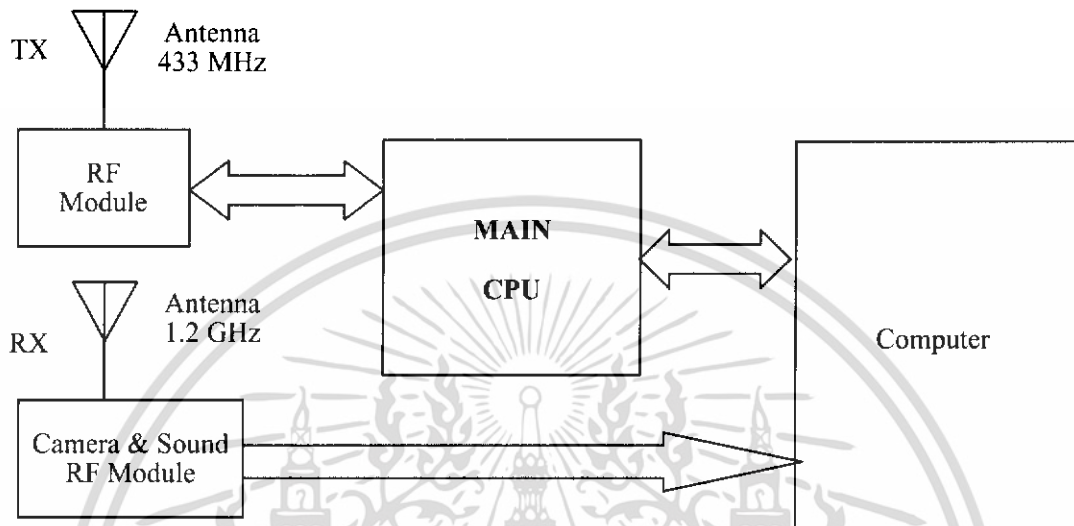
##### 2.1.1 การทำงานในส่วนของตัวหุ่นยนต์

จากรูปที่ 2.1 เราอธิบายได้ว่าตัว RF Module จะรับสัญญาณเข้ามาในรูปแบบของข้อมูลอนุกรม โดยจะทำการส่งข้อมูลที่ได้รับเข้ามาได้ไปยัง MAIN CPU ซึ่งเราใช้ไมโครคอนโทรลเลอร์ตระกูล MCS-51 เป็นตัวประมวลผลหลัก โดย MCS-51 จะทำการถอดรหัสออกมาว่าให้ไปควบคุมวงจรส่วนใด เช่น ควบคุมการขับเคลื่อน ควบคุมการปิดเปิดไฟ หรือควบคุมตัวแขนเพื่อช่วยในการปีนป่าย เป็นต้น โดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่คำสั่งที่รับเข้ามาจะอยู่ในรูปแบบของ ASCII Code นอกจากนี้ยังสามารถส่งได้ทั้งภาพ และเสียงอีกด้วย โดยที่ข้อมูลทั้งหมดส่งผ่านระบบไร้สาย

## 2.2 ส่วนของสถานีควบคุม



รูปที่ 2.2 ส่วนประกอบหลักภายในสถานีควบคุม

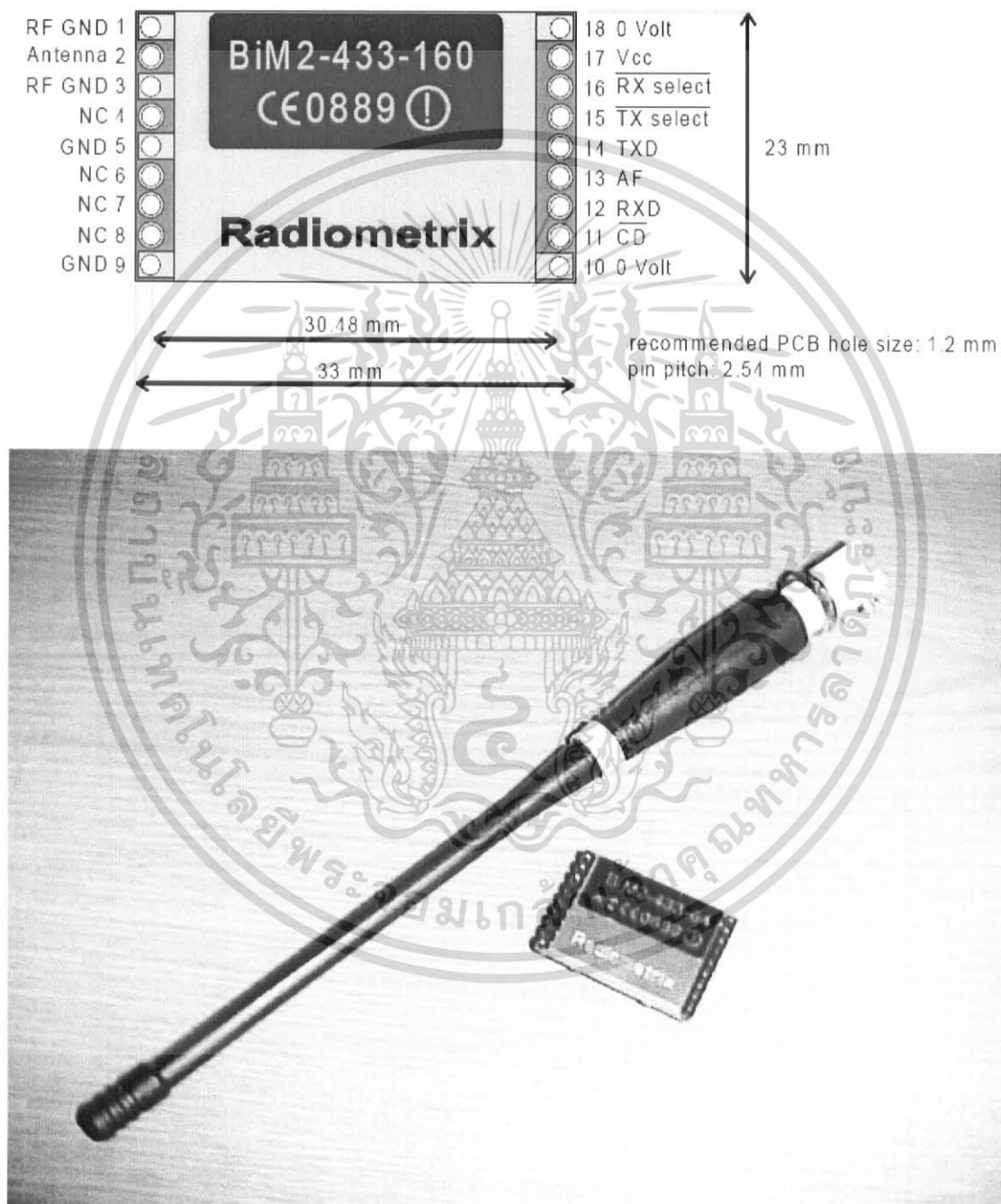
### 2.2.1 การทำงานในส่วนของสถานีควบคุม

ในส่วนนี้เราจะใช้คอมพิวเตอร์เป็นตัวบังคับสั่งงานหุ่นยนต์จากระยะไกลโดยผ่าน RF Module และมี MAIN CPU เป็นตัวปรับอัตราการส่งข้อมูลให้มีความเร็วสัมพันธ์กับการทำงานของเครื่องคอมพิวเตอร์ โดยที่เราสามารถรับทั้งภาพ และเสียงได้จาก Camera & Sound RF Module โดยภาพและเสียงนี้จะถูกส่งไปยังคอมพิวเตอร์ เพื่อแสดงผลบนหน้าจอเพื่อให้สามารถมองเห็นสภาพแวดล้อมต่างๆ ตามที่หุ่นยนต์สามารถเคลื่อนที่ไปยังบริเวณนั้นๆ ได้

## 2.3 ทฤษฎีที่เกี่ยวข้องและอุปกรณ์ที่นำมาใช้งาน

### 2.3.1 BIM2-433-64 Radiometrix RF Module

เป็นหัวใจหลักในการรับและส่งข้อมูลระหว่างสถานีควบคุม กับตัวหุ่นยนต์ โดยข้อมูลที่ส่งออกเป็นลักษณะของข้อมูลอนุกรม



รูปที่ 2.3 รูปร่างหน้าตาของ BIM2-433-64 และเสาอากาศที่ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.3.1.1 คุณสมบัติทั่วไปของ BIM2-433-64

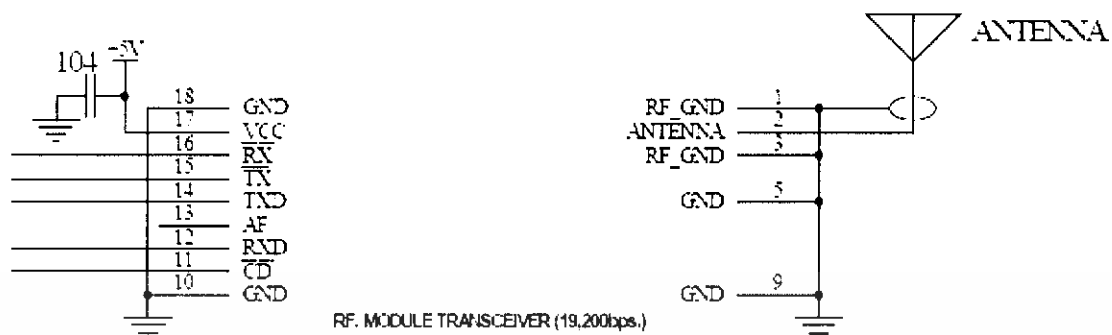
- สามารถรับและส่งได้ในตัวเดียวกัน
- ระยะในการส่ง 200 เมตร ในที่โล่ง และ 50 เมตร ในตัวอาคาร
- ความเร็วในการรับส่งข้อมูล 64 kbps
- กำลังส่ง 10 mw.
- ใช้ไฟเลี้ยงที่ 3 – 5 โวลต์ และกินกระแสต่ำกว่า 20 mA.

### 2.3.1.2 รายละเอียดหน้าที่ของขาต่างๆ

RF GND	เป็นขาสำหรับต่อกราวด์ของภาค RF ซึ่งต่อขา 1 และ 3 เข้าด้วยกัน
Antenna	เป็นขาสำหรับต่อกับเสาอากาศ
GND	เป็นขาสำหรับต่อกราวด์ ได้แก่ขา 5,9,10,18 เข้าด้วยกัน
CD	เป็นขาสำหรับ Check carrier detect โดยถ้ามีสถานะเป็นลอจิก 0 ก็คือเริ่มมีการเข้ามาของข้อมูลนั่นเอง
RXD	เป็นขาสำหรับรับข้อมูล
TXD	เป็นขาสำหรับส่งข้อมูล
AF	เป็นขาสำหรับ Filter ด้านรับข้อมูลจากวงจร Demodulation ขานี้ควรมีค่า Impedance ที่มากกว่า 2 k $\Omega$ และน้อยกว่า 100 pF
Vcc	เป็นขาไฟเลี้ยง 3-5 โวลต์
TX,RX	เป็นขาสำหรับเลือกรูปแบบการควบคุมการส่งข้อมูล โดยมีตารางดังนี้

ตารางที่ 2.1 ตารางแสดงสถานะของขา TX และ RX

TX	RX	สถานะ
0	0	ทดสอบแบบ Close Loop
0	1	ส่งอย่างเดียว
1	0	รับอย่างเดียว
1	1	Power down ( <1 $\mu$ A )

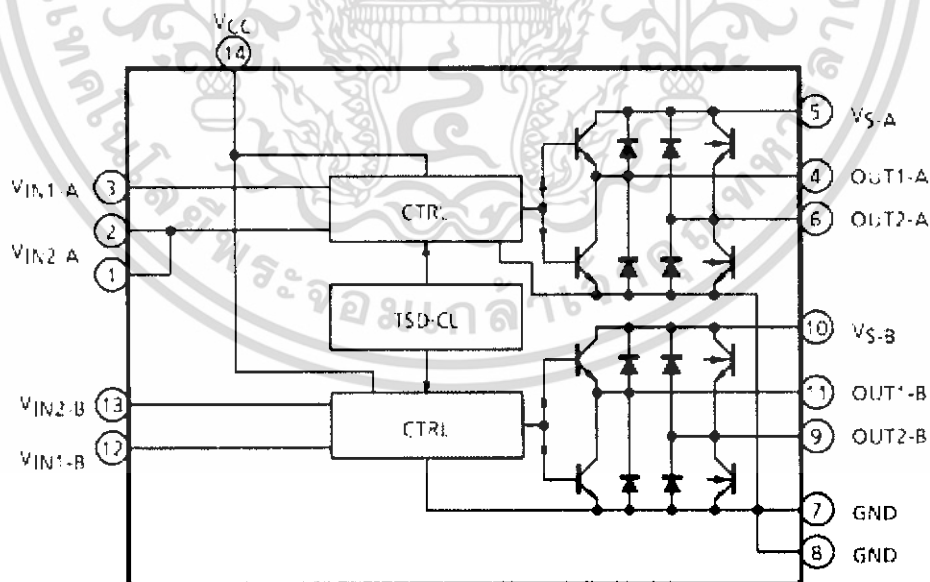


BIM2-433-64

รูปที่ 2.4 การต่อใช้งานของ BIM2-433-64 อย่างง่ายที่สุด

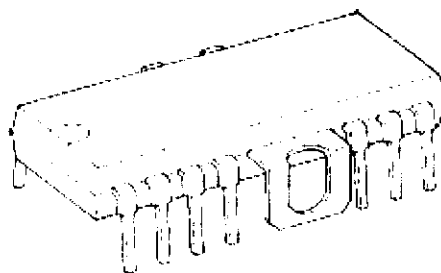
### 2.3.2 TA7279P/AP DC Motor Rotation Control

ไอซีตัวนี้ทำหน้าที่ควบคุมการเคลื่อนที่ของหุ่นยนต์ โดยที่สามารถควบคุมทิศทางการหมุนของมอเตอร์ได้ 2 ตัว โดยเราสามารถเลือกค่าแรงดันที่ขั้วมอเตอร์ได้โดยไม่ผลต่อแรงดันควบคุมทางด้านอินพุท โดยที่อินพุทเราต้องการเพียงสถานะลอจิก 0 หรือ 1 เท่านั้นที่เข้ามาควบคุม ซึ่งเราสามารถควบคุมได้โดยตรงจากไมโครคอนโทรลเลอร์



รูปที่ 2.5 รายละเอียดการใช้งานต่างๆ และ โครงสร้างภายใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 รูปร่างหน้าตาของไอซีขับมอเตอร์กระแสตรงเบอร์ TA7279P/AP

### 2.3.2.1 คุณสมบัติทั่วไปของ TA7279P/AP

- Output Current ที่ 1A.(AVE) และ 3A.(PEAK)
- ช่วงแรงดันที่ใช้งาน
  - $V_{cc(opr.)} = 6 - 18V$  (P,AP)
  - $V_s(opr.) = 0 - 16V$  (P) ,  $0 - 18V$  (AP)
- สามารถควบคุมมอเตอร์กระแสตรงได้ 2 ตัว

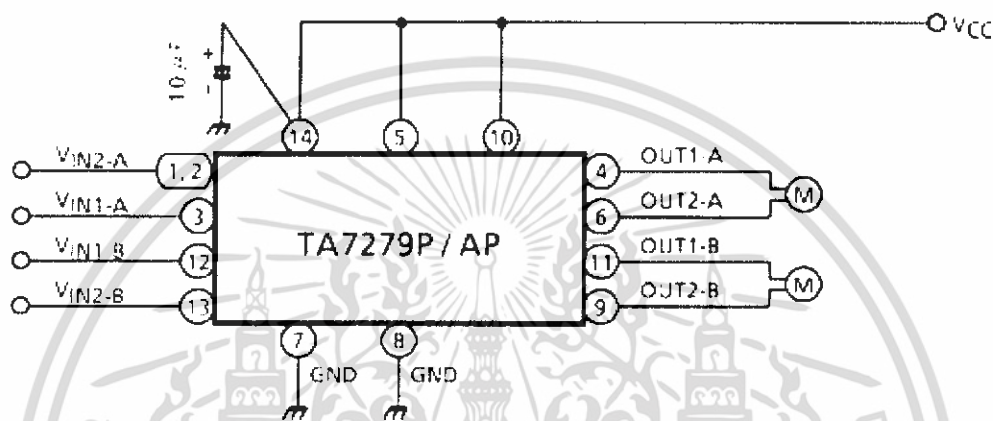
ตารางที่ 2.2 รายละเอียดหน้าที่ของขาต่างๆ

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 ฟังก์ชันการทำงานของสัญญาณควบคุม

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



รูปที่ 2.7 การต่อใช้งานของ TA7279P/AP

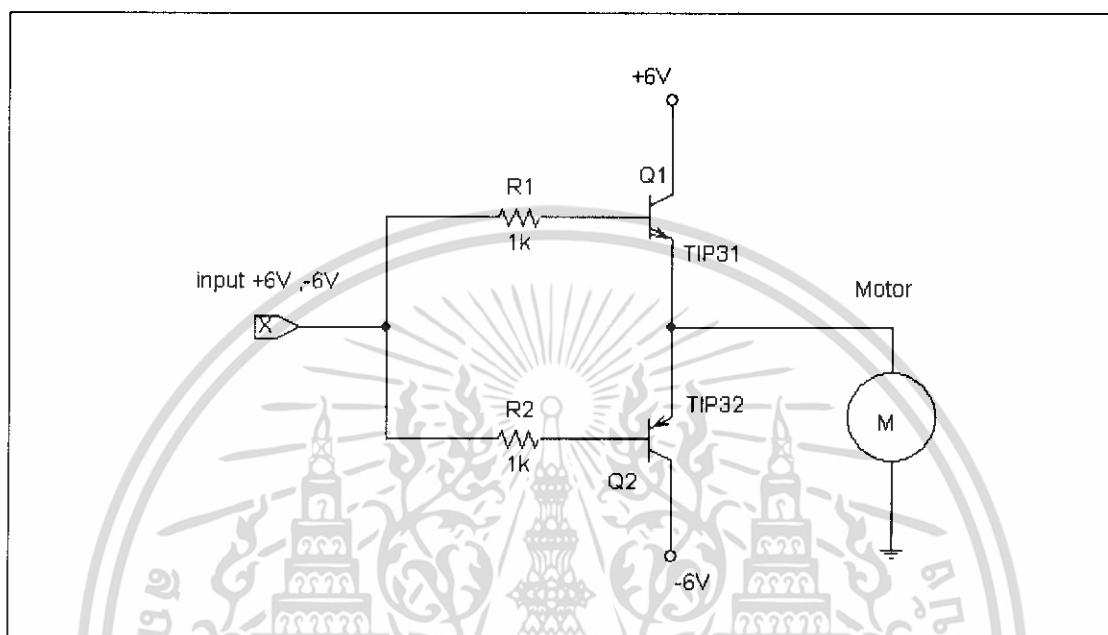
### 2.3.3 การควบคุมทิศทางการหมุนของมอเตอร์กระแสตรงด้วยทรานซิสเตอร์

เนื่องจากการควบคุมการเคลื่อนที่ของแขนหุ่นยนต์ที่ใช้ในการช่วยเป็น پایลิ่งต่างๆ นั้นเราใช้มอเตอร์กระแสตรงเพียงตัวเดียว ด้วยสาเหตุนี้เอง เราจึงไม่จำเป็นที่จะต้องใช้โซลีนอยด์อย่างเจ้า TA7279 เพราะว่ามันเปลืองโดยเปล่าประโยชน์ เพราะฉะนั้นเราจึงใช้ทรานซิสเตอร์เป็นตัวขับเคลื่อน

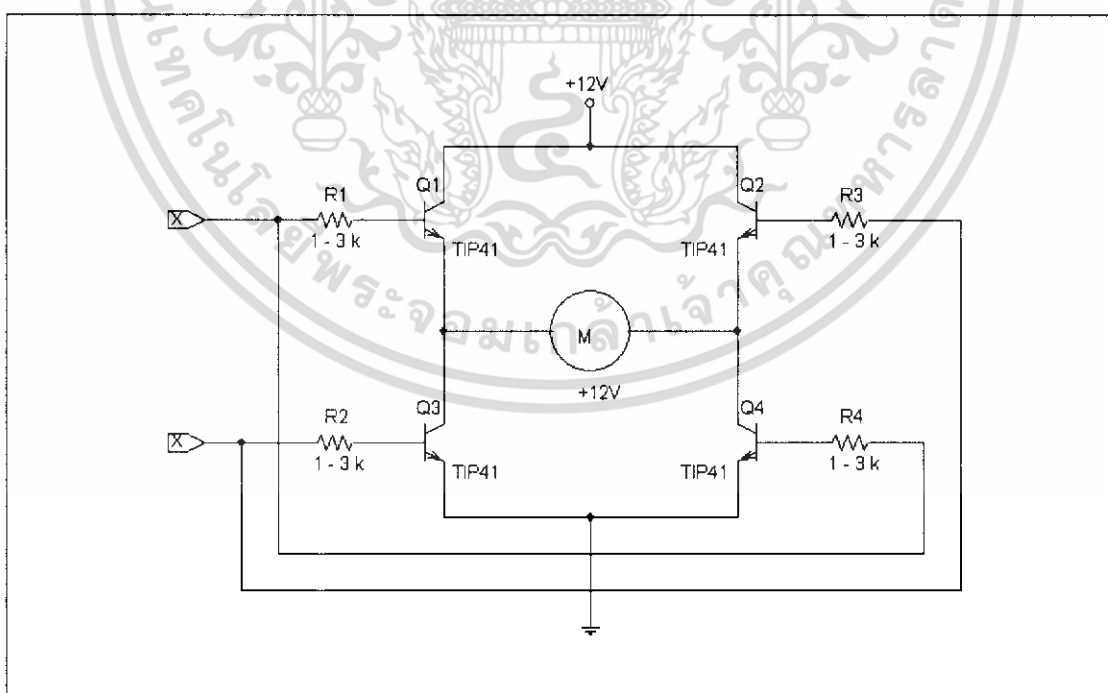
ทรานซิสเตอร์ที่ทำหน้าที่เป็นสวิตช์ควบคุมการทำงานของมอเตอร์ โดยทั่วไปจะมีการใช้งานอยู่ 2 แบบคือ แบบแรกจะมีลักษณะดังแสดงในรูปที่ 2.8 ซึ่งจะเห็นว่าเราใช้ทรานซิสเตอร์ในการควบคุมการทำงานทั้งหมด โดยถ้าจ่ายกระแสไฟฟ้าให้กับทรานซิสเตอร์ตัวแรก มอเตอร์ก็จะหมุนตามเข็มนาฬิกา และถ้าไม่จ่ายกระแสไฟฟ้า มอเตอร์ก็จะไม่หมุน แต่ถ้าจ่ายกระแสไฟฟ้าให้กับทรานซิสเตอร์อีกตัวหนึ่ง มอเตอร์ก็จะหมุนในทิศทางทวนเข็มนาฬิกา จะเห็นได้ว่าการออกแบบในลักษณะนี้จะต้องใช้แหล่งจ่ายไฟฟ้าแบบ 2 ขั้ว จากรูปจะเป็นการจ่ายกระแสไฟฟ้าขนาด +6 และ -6 โวลต์ให้กับมอเตอร์ขนาด 6 โวลต์ 1 ตัว จะสังเกตได้ว่าในวงจรจะมีตัวต้านทานที่ใช้ในการไบอัสที่ขาเบสของทรานซิสเตอร์ เพื่อป้องกันกระแสไฟฟ้ากระชากที่มาจากพอร์ตของคอมพิวเตอร์หรือลอจิกเกต ถ้าไม่มีตัวต้านทานอาจจะทำให้ทรานซิสเตอร์ไหม้ได้

ค่าของตัวต้านทานที่ใช้นั้นจะขึ้นอยู่กับค่าของแรงดันไฟฟ้า ค่ากระแสไฟฟ้าที่มอเตอร์สามารถรับได้ และประเภทของทรานซิสเตอร์ที่ใช้ ปกติแล้วจะใช้ตัวต้านทานขนาด 1-3 กิโลโอห์ม ซึ่งสามารถใช้กฎของโอห์มคำนวณหาค่าความต้านทานได้ โดยสังเกตจากอัตราขยายและปริมาณกระแสไฟฟ้าจากเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทรานซิสเตอร์ ซึ่งจะต้องทำการทดลองไปจนกว่าจะหาค่าความต้านทานที่เหมาะสมได้ มีข้อสังเกตอยู่ประการหนึ่งคือ วงจรอิเล็กทรอนิกส์ที่ใช้ในการควบคุมจะร้อนมาก ไม่ควรใช้ความต้านทานที่มีค่าต่ำกว่า 1 กิโลโอห์ม และควรจะอ่านคู่มืออุปกรณ์อิเล็กทรอนิกส์ประกอบในการใช้งานด้วย ซึ่งเราได้แนบมาด้วยในส่วนของภาคผนวก



รูปที่ 2.8 การควบคุมทิศทางการหมุนของมอเตอร์โดยใช้ทรานซิสเตอร์ 2 ตัว



รูปที่ 2.9 การควบคุมทิศทางการหมุนของมอเตอร์โดยใช้ทรานซิสเตอร์แบบ NPN เป็นรูปตัว “H”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

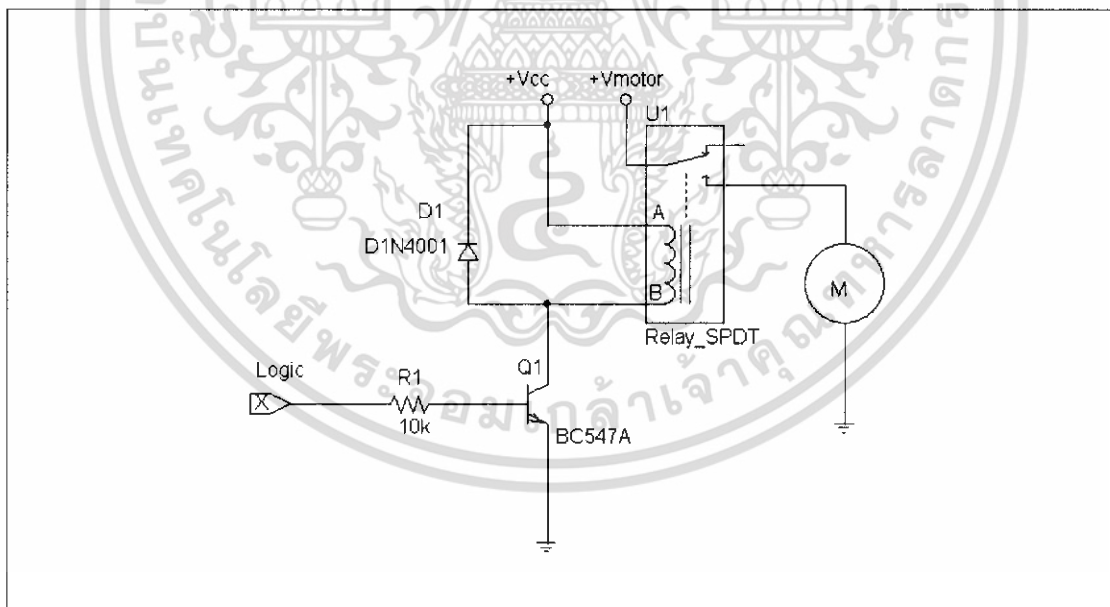
วงจรควบคุมที่ใช้ทรานซิสเตอร์แบบอื่นๆ ได้แก่ การต่อทรานซิสเตอร์แบบตัวเอช(H) ดังแสดงในรูปที่ 2.9 จะเห็นได้ว่าเมื่อ Q1 และ Q4 ทำงาน จะทำให้มอเตอร์หมุน และถ้า Q2 และ Q3 ทำงาน มอเตอร์จะหมุนในทิศทางตรงกันข้าม การเลือกใช้ตัวต้านทานก็ควรเลือกใช้ตามความเหมาะสมกับวงจร เพื่อป้องกันความเสียหายที่จะเกิดขึ้นกับทรานซิสเตอร์

### 2.3.4 การควบคุมทิศทางหมุนของมอเตอร์กระแสตรงด้วยรีเลย์

การควบคุมทิศทางหมุนของมอเตอร์โดยใช้รีเลย์นั้น เป็นการควบคุมการหมุนแบบทางเดียว ซึ่งเป็นวิธีการที่เก่าและค่อนข้างยุ่งยาก เราควรที่จะหาวิธีการอื่นมากกว่าที่จะใช้วิธีการนี้ เพราะอาจจะพบปัญหาในขณะใช้งาน เช่น รีเลย์เกิดการชำรุดในขณะใช้งาน แต่วิธีการนี้มีข้อดีคือ เป็นวิธีการที่ประหยัดกว่าวิธีอื่นๆ ใช้งานง่าย และใช้พื้นที่ในขณะติดตั้งน้อย

การควบคุมรีเลย์ประเภท Single-Pole เป็นการควบคุมแบบเปิด-ปิด คือ รีเลย์จะยกขึ้นเมื่อไม่มีกระแสไฟฟ้าไหลผ่านและรีเลย์จะไม่ทำงาน แต่ถ้าอยู่ในสถานะปิด กระแสไฟฟ้าจะสามารถไหลผ่านรีเลย์ได้ ทำให้มอเตอร์สามารถหมุนได้

การควบคุมรีเลย์ให้ทำงานได้นั้นสามารถทำได้หลายรูปแบบ เช่น ทำเป็นปุ่มกด ซึ่งเป็นวิธีการควบคุมแบบธรรมดา หรือการใช้สัญญาณดิจิทัลดังรูปที่ 2.10 หรืออาจใช้เกิดแบบต่างๆ และอาจรวมไปถึงการใช้คอมพิวเตอร์ในการควบคุมผ่านทางพอร์ตของคอมพิวเตอร์ หรือ ไมโครคอนโทรลเลอร์ก็สามารถทำได้โดยไม่ยากนัก

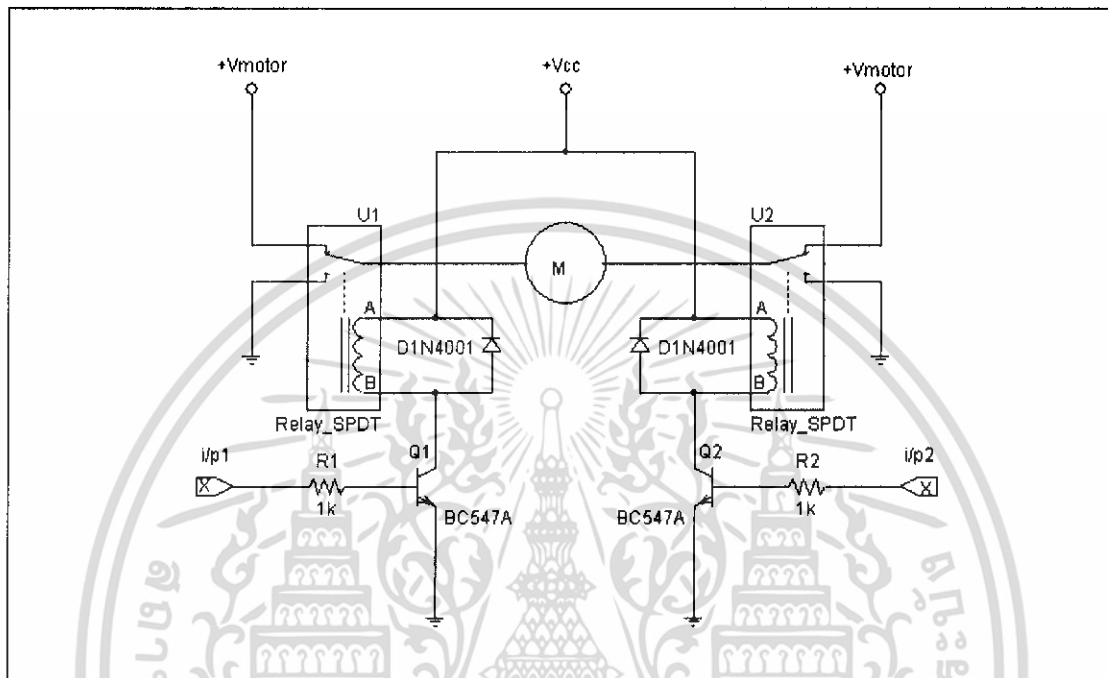


รูปที่ 2.10 การใช้รีเลย์ควบคุมการปิด-เปิดมอเตอร์โดยใช้สัญญาณดิจิทัล

จากรูปที่ 2.10 เราสามารถอธิบายได้ว่าเมื่อมีลอจิก 1 มาไปอัสทางขาเบสของทรานซิสเตอร์ ทรานซิสเตอร์จะนำกระแส โดยไปอัสผ่าน R1 ซึ่งทำหน้าที่ในการจัดไบอัสที่เหมาะสมให้กับทรานซิสเตอร์ โดยทั่วไปแล้วรีเลย์ต้องการกระแสประมาณ 50mA ถึง 100mA โดยประมาณในการกระตุ้นให้เกิดการเหนี่ยวนำขึ้นที่ขดลวดภายในตัวของรีเลย์ ซึ่งจะทำให้เกิดการเปิด-ปิดหน้าสัมผัสภายใน โดยที่ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไดโอดที่ต่ออยู่นั้นทำหน้าที่ป้องกันกระแสไหลย้อนกลับ อันเนื่องมาจากการขยับตัวของสนามแม่เหล็กที่เกิดขึ้นในขณะที่ทรานซิสเตอร์หยุดทำงานโดยฉับพลัน ซึ่งทำให้เกิดกระแสไหลที่สูงมากๆ เพียงเสี้ยววินาที ซึ่งอาจทำให้ทรานซิสเตอร์ของเราพังได้

แต่วงจรในรูปที่ 2.10 นั้นเป็นการควบคุมมอเตอร์กระแสตรงแบบหมุนทางเดียว ดังนั้นในการที่จะทำให้มอเตอร์สามารถหมุนไปมาได้สองทิศทาง เราสามารถทำได้ดังรูปที่ 2.11



รูปที่ 2.11 การใช้รีเลย์ควบคุมทิศทางการหมุนมอเตอร์โดยใช้สัญญาณดิจิทัล

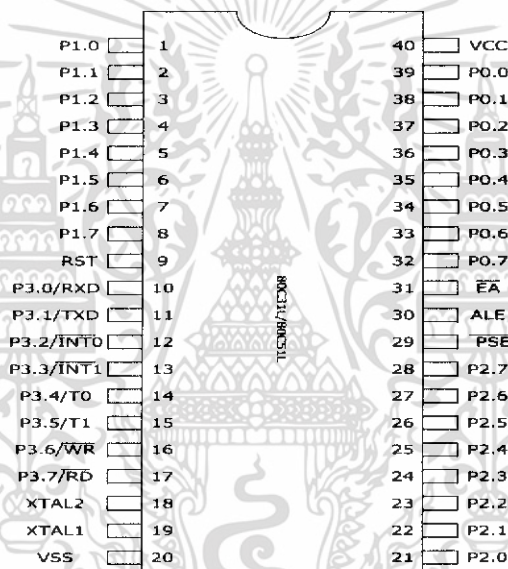
จากรูปที่ 2.11 เราสามารถอธิบายได้ว่าเมื่อให้ลอจิก 1 ที่ขาเบสของ Q1 จะทำให้ Q1 นำกระแส ซึ่งจะส่งผลให้รีเลย์ทำงาน หน้าสัมผัสภายในจะสลับมาที่กราวด์ ส่วน Q2 เราให้ลอจิกเป็น 0 ซึ่งจะทำให้ Q2 อยู่ในสถานะคัทออฟ คือทรานซิสเตอร์ไม่ทำงาน ส่งผลให้รีเลย์ไม่ทำงานด้วย หน้าสัมผัสภายในยังคงอยู่ที่ +Vmotor ทำให้กระแสไฟฟ้าไหลจาก +Vmotor ของ Q2 ไหลผ่านมอเตอร์มายังกราวด์ทางด้านของ Q1 ทำให้มอเตอร์หมุน

ในทางกลับกัน หากให้ลอจิก 1 ที่ขาเบสของ Q2 และลอจิก 0 ที่ขาเบสของ Q1 จะทำให้มอเตอร์หมุนในทิศทางตรงกันข้ามกับเมื่อซักรุ่นนี้ ด้วยหลักการเดียวกันนั่นเอง

สำหรับในเรื่องของปริมาณโหลดที่วงจรนี้สามารถรับได้นั้น ขึ้นอยู่กับขนาดของรีเลย์ทางด้านคุณสมบัติทางเอาท์พุทของรีเลย์เองว่า สามารถทนแรงดันและกระแสได้สูงสุดเท่าไร ซึ่งไม่เกี่ยวกับทางด้านขดลวดซึ่งทำหน้าที่เหนี่ยวนำให้ทำงาน เพราะในส่วนนี้เป็นส่วนที่เกี่ยวกับการควบคุมให้รีเลย์ทำงานหรือไม่ทำงานเท่านั้นเอง ซึ่งโดยส่วนใหญ่จะขึ้นอยู่กับค่าแรงดันที่ใช้ในการเหนี่ยวนำขดลวดภายใน โดยทั่วไปจะมีค่าไม่สูงมากนัก เราสามารถใช้ทรานซิสเตอร์มาควบคุมได้อย่างง่ายดาย ในวงงบประมาณที่ต่ำมากๆ สำหรับงานควบคุมระดับนี้

### 2.3.5 ไมโครคอนโทรลเลอร์ MCS-51

- ต้องการแหล่งจ่ายไฟ +5V ชุดเดียว
- มีหน่วยความจำโปรแกรม (Program Memory) ขนาด 4 กิโลไบต์สำหรับเบอร์ 8051 และ 8031 สำหรับเบอร์ 8052 มีหน่วยความจำถึง 8 กิโลไบต์
- มีหน่วยความจำสำหรับเก็บข้อมูล (Data Memory) ขนาด 128 ไบต์ สำหรับเบอร์ 8052 ขึ้นไป มีถึง 256 ไบต์
- หน่วยความจำสำหรับเก็บโปรแกรมและข้อมูลแยกจากกันอย่างละ 64 กิโลไบต์
- มีไทม์เมอร์ และเคาน์เตอร์ ขนาด 16 บิต 2 ชุด (สำหรับ 8052 มี 3 ชุด) ทำงานได้ 4 โหมด
- รับอินเทอร์รัพท์ได้ 6 แหล่ง 5 เวกเตอร์ สำหรับเบอร์ 8052 ขึ้นไปมี 8 แหล่ง 6 เวกเตอร์
- มีพอร์ตรับส่งข้อมูลอนุกรม (UART) 2 พอร์ตแบบ Full Duplex เลือกรูปได้ 4 โหมด
- มีคำสั่งในการทำ AND, OR หรือ Complement ได้ทั้งแบบ 8 บิต และ 1 บิต

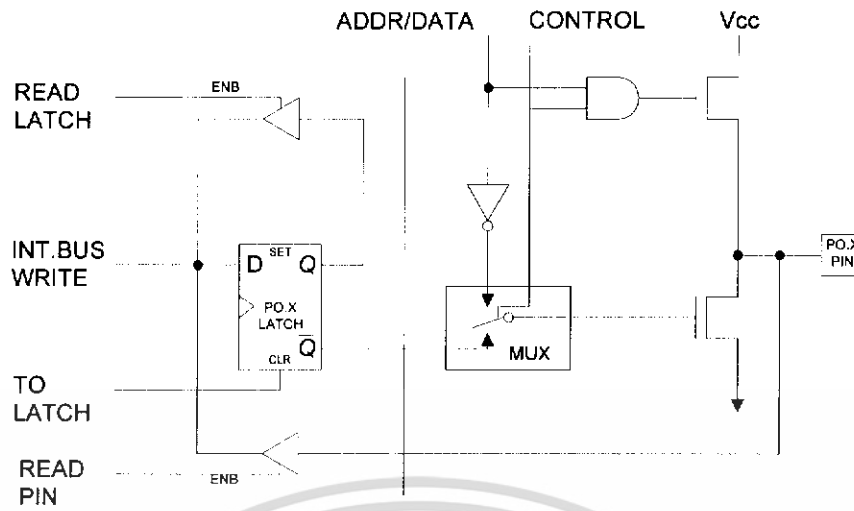


รูปที่ 2.12 การจัดวางขาของ 8051

#### 2.3.5.1 8051 เป็นไมโครคอนโทรลเลอร์ขนาด 40 ขา ซึ่งมีขาต่าง ๆ ดังนี้

- Vcc (ขา 40) ต่อกับ + 5V
- Vss (ขา 20) เป็นขา GND
- พอร์ต 0 (ขา 32-39) มีทั้งหมด 8 บิต คือ (P0.7-P0.0) มีโครงสร้างแบบ Open-Drain Bi-directional ดังแสดงในรูปที่ 2.12

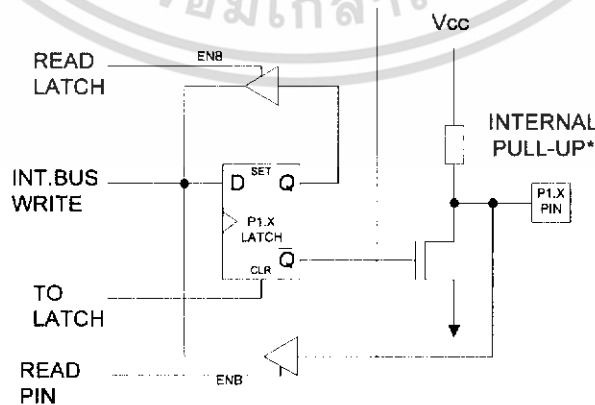
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



**A. Port 0 Bit**

รูปที่ 2.13 แสดงโครงสร้าง พอร์ต 0 (บิต)

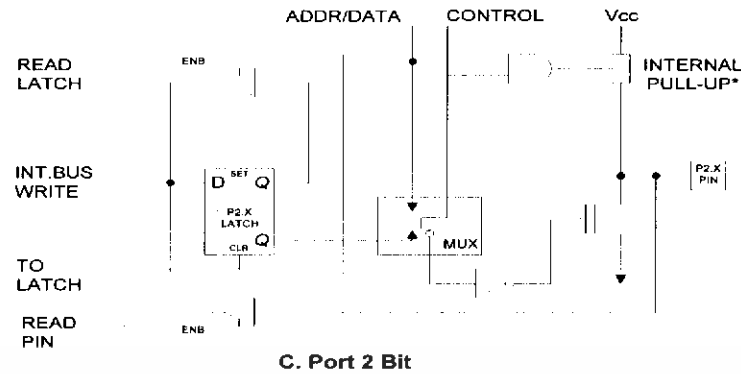
- พอร์ต 0 (ขา 32-39) มีทั้งหมด 8 บิต คือ (P0.7-P0.0) ใช้งานได้ 2 หน้าที่ คือเป็นทางผ่านของแอดเดรสบัสและค่าตัวบิตเมื่อต้องการติดต่อกับหน่วยความจำภายนอกหรือเป็นไอโอพอร์ต ถ้าต้องการ ให้ทำงานเป็นอินพุทพอร์ทต้องส่งลอจิก "1" ไปยังพอร์ทนี้ จะมีผลให้ Q ของ D-FF เป็น "0" ทำให้ FET ตัวล่างมีสถานะ OFF สัญญาณที่ใช้อ่านอินพุทพอร์ทแลตช์โดยส่งสัญญาณ READ LATCH ไปกระตุ้นที่ Tri-State Buffer ตัวบนและการอ่าน Port (pin) จะใช้สัญญาณ Read (pin)
- พอร์ต 1 (ขา 1-8) มีทั้งหมด 8 บิต คือ (P1.0-P1.7) มีโครงสร้างคล้าย พอร์ต 0 แต่จะใช้ความต้านทานภายในพูลอัพแทน Internal Pull up Register มีโครงสร้างดังรูป 2.13
- พอร์ต 2 (ขา 21-28) มีทั้งหมด 8 บิต คือขา (P2.7-P2.0) มีโครงสร้างคล้าย พอร์ต 0 โดยมี FET ตัวล่างตัวเดียวส่วนด้านบนใช้ความต้านทานพูลอัพแทน (Internal pull up) พอร์ทนี้ ใช้งาน 2 หน้าที่ คือสามารถใช้เป็นแอดเดรสบัสขนาด 8 บิต (A15-A8) และเป็นไอโอพอร์ทใช้งานทั่วไปเมื่อจะใช้งานเป็นอินพุทพอร์ทต้องส่งลอจิก "1" มาที่พอร์ทนี้ก่อน เพื่อบังคับให้ FET อยู่ในสภาวะ off ดังแสดงในรูปที่ 2.14



**B. Port 1 Bit**

รูปที่ 2.14 โครงสร้างของพอร์ท 1 (บิต)

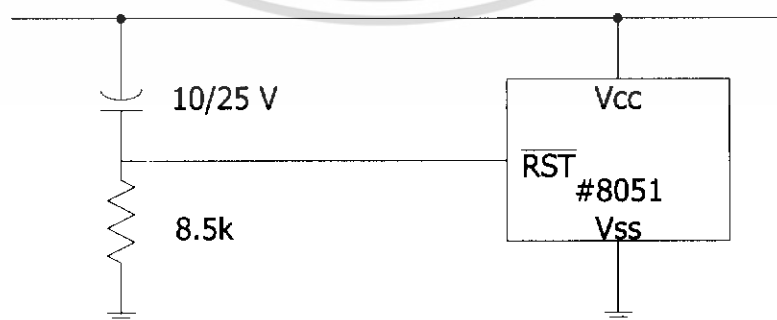
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 โครงสร้างของ พอร์ต 2 (บิต)

- พอร์ต 3 (ขา 10-17) มีทั้งหมด 8 บิต คือ ขา (P3.7-P3.0) มีโครงสร้างคล้ายพอร์ต 1 ทำงานได้ 2 หน้าที่คือเป็นไอโอพอร์ตถ้าจะโปรแกรมให้เป็นอินพุทพอร์ตต้องส่งลอจิก "1" มาที่พอร์ตนี้อีก และอีกหน้าที่หนึ่งก็คือใช้ส่งสัญญาณควบคุมออกมา และรับสัญญาณเข้าไปสัญญาณต่าง ๆ มีดังนี้

P3.0/RXD (Serial Input Port)	เป็นขาที่ใช้รับข้อมูลแบบอนุกรม (UART)
P3.1/TXD (Serial Output Port)	เป็นขาที่ใช้ส่งข้อมูลแบบอนุกรม (UART)
P3.2/INT0 (External Interrupt 0)	ใช้รับสัญญาณการขัดจังหวะจากภายนอกเบอร์ 0
P3.3/INT1 (External Interrupt 1)	ใช้รับสัญญาณการขัดจังหวะจากภายนอกเบอร์ 1
P3.4/T0 (Counter 0 External Input)	ขารับสัญญาณพัลส์อินพุทเข้าไปยังวงจร Counter0 (เป็นอินพุทโหมดเคาน์เตอร์)
P3.5/T1 (Counter 1 External Input)	ขารับสัญญาณพัลส์อินพุทเข้าไปยังวงจร Counter1 (เป็นอินพุทโหมดเคาน์เตอร์)
P3.6/WR (External Data Memory Write Strobe)	ขาสัญญาณควบคุมการเขียนข้อมูลลงหน่วยความจำข้อมูลภายนอก
P3.7/RD (External Data Memory Read Strobe)	ขาสัญญาณควบคุมการอ่านข้อมูลจากหน่วยความจำข้อมูลภายนอก

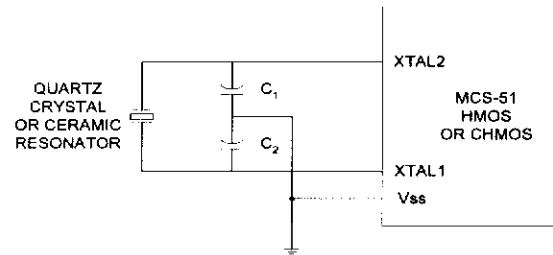


รูปที่ 2.16 การต่อขารีสตาร์ทให้กับ 8051

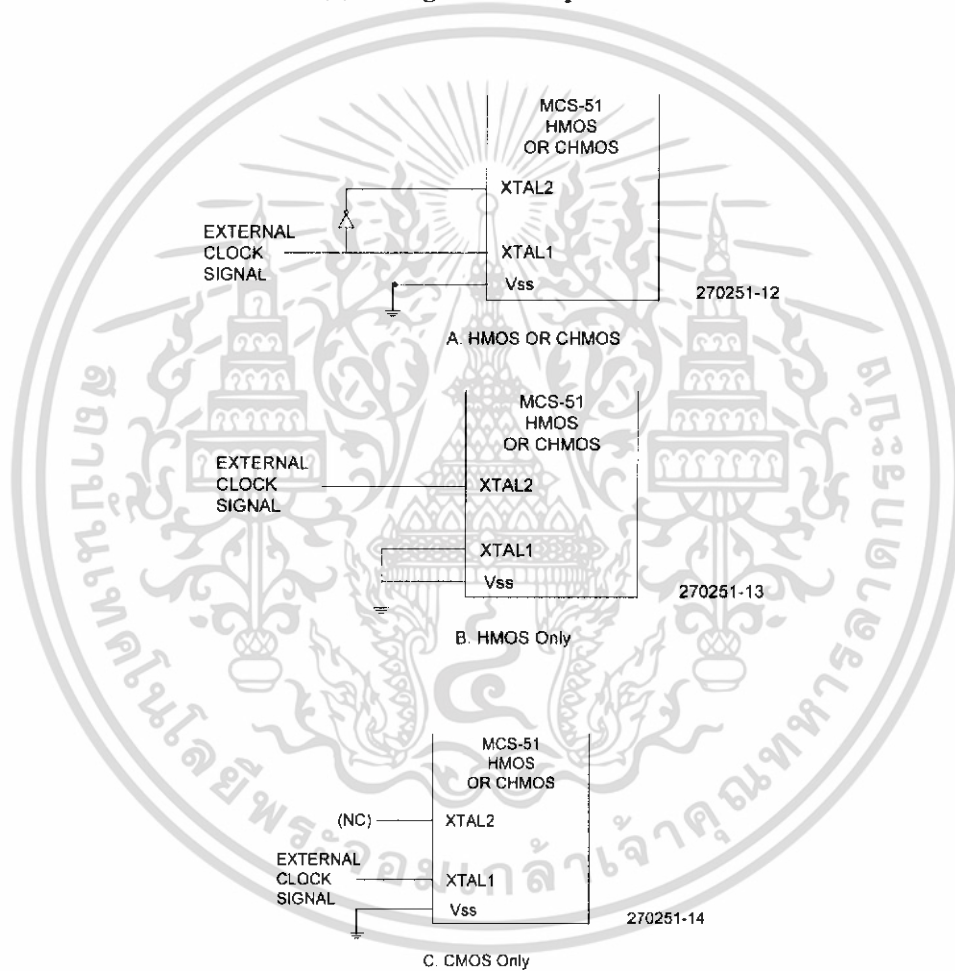
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.3.5.2 วงจรคล็อกของ MCS-51

การต่อมียู่ด้วยกัน 2 รูปแบบคือแบบใช้คล็อกภายในและคล็อกจากภายนอกมีรูปแบบการต่อดังรูปที่ 2.17



(a) Using the on-chip Oscillator



(b) Using the External Clock

รูปที่ 2.17 วงจรสร้างคล็อกของ 8051

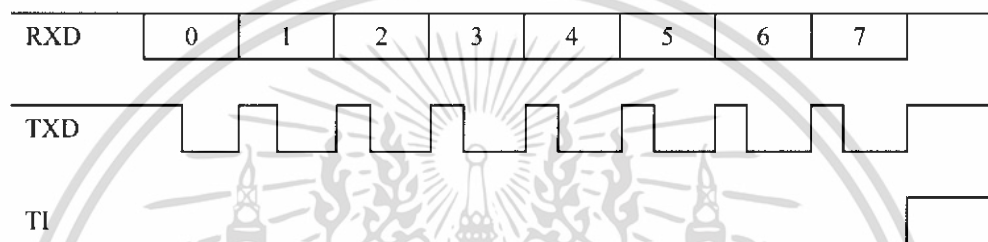
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.3.5.3 พอร์ตสื่อสารอนุกรม

พอร์ตสื่อสารอนุกรมมีโครงสร้างการทำงานในแบบที่เรียกว่าฟูลดูเพล็กซ์ (Full Duplex) ในการรับและส่งข้อมูลอนุกรมได้ในเวลาเดียวกัน โดยทางด้านส่งใช้ขา TxD (พอร์ต 3.1) ทางด้านรับใช้ขา RxD (พอร์ต 3.0) SBUF ใช้เป็นบัฟเฟอร์สำหรับรับและส่งข้อมูลอนุกรม

พอร์ตสื่อสารอนุกรมสามารถโปรแกรมการทำงานได้หลายโหมดด้วยกันโดยเลือกที่บิต SM1 และ SM0 ซึ่งอยู่ในรีจิสเตอร์ควบคุม SCON การทำงานทั้ง 4 โหมด ของพอร์ตสื่อสารอนุกรม มีดังนี้

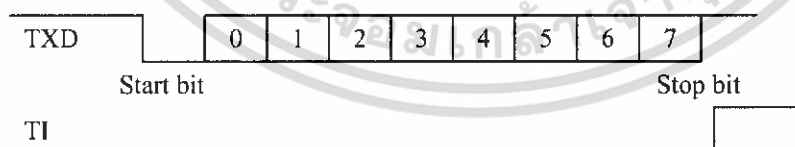
**โหมด 0** : พอร์ตสื่อสารอนุกรม 8 บิต โดยการส่งจะเลื่อนออกทีละบิตโดยส่งบิต D0 ออกไปก่อนทางขา RxD และไม่มีการส่ง start bit แต่จะส่ง shift clock ทางขา TxD (ความเร็ว 1/12 เท่าของ CPU Clock)



**โหมด 1** : พอร์ตสื่อสารอนุกรม 10 บิต ข้อมูล 8 บิต 1 start bit และ 1 stop bit และสามารถเปลี่ยนแปลง (ความเร็วในการส่งข้อมูลได้ โดยขึ้นกับบิต SMOD ใน PCON และอัตราโอเวอร์โพล์ของ Timer 1)

$$\text{Baud Rate Mode 1,3} = \frac{2^{\text{SMOD}} \times \text{Oscillator Freq.}}{32 \times 12 \times [256 - (\text{TH1})]} \quad \text{โดยใช้ Timer 1}$$

$$\text{Baud Rate Mode 1,3} = \frac{\text{Oscillator Freq.}}{32 \times [65536 - (\text{RCAP2H, RCAP2L})]} \quad \text{โดยใช้ Timer 2}$$

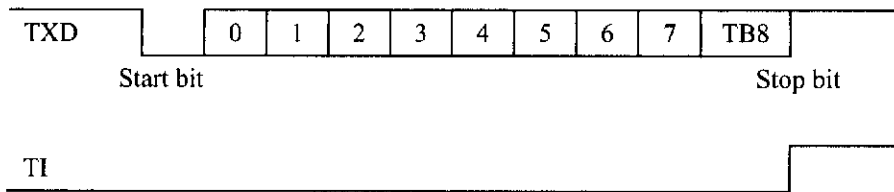


**โหมด 2** : พอร์ตสื่อสารอนุกรม 11 บิต ข้อมูล 9 บิต 1 start bit และ 1 stop bit (TB8 นิยมนำมาใช้ส่ง Parity bit) ความเร็วในการรับส่งข้อมูลเท่ากับ 1/32 และ 1/64 ของ CPU Clock โดยขึ้นกับบิต SMOD ใน PCON

$$\text{-Baud Rate Mode 2} = (1/32 \text{ Osc Freq}) \quad \text{เมื่อ SMOD} = 1$$

$$\text{-Baud Rate Mode 2} = (1/64 \text{ Osc Freq}) \quad \text{เมื่อ SMOD} = 0$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



**โหมด 3** : พอร์ตสื่อสารแบบ 11 bit UART โดยส่งข้อมูล 9 บิต 1 start bit และ 1 stop bit เหมือนโหมด 2 ยกเว้นอัตราความเร็วจะขึ้นกับบิต SMOD ใน PCON และอัตราโอเวอร์โพล์ของ Timer 1, 2 สำหรับ 8051 หรือ อัตราโอเวอร์โพล์ของ Timer 2 (สำหรับ 80C154D)

$$\text{Baud Rate Mode 3} = \frac{2^{\text{SMOD}} \times \text{Oscillator Freq.}}{32 \times 12 \times [256 - (\text{TH1})]} \quad \text{โดยใช้ Timer 1}$$

$$\text{Baud Rate Mode 3} = \frac{\text{Oscillator Freq.}}{32 \times [65536 - (\text{RCAP2H}, \text{RCAP2L})]} \quad \text{โดยใช้ Timer 2}$$



**2.3.5.4 SM0, SM1 บิตเลือกโหมดการทำงาน**

SM0	SM1	โหมด	การทำงาน
0	0	0	shift register อัตราเร็วในการรับหรือส่งข้อมูลเท่ากับ 1/12 ของ CPU osc
0	1	1	8 bit UART อัตราเร็วในการรับหรือส่งข้อมูลกำหนดได้จาก Timer 1,2
1	0	2	9 bit UART อัตราเร็วในการรับหรือส่งข้อมูล = 1/32 หรือ 1/64 ของความถี่ออสซิลเลเตอร์ ขึ้นกับบิต SMOD ใน PCON
1	1	3	9 bit UART อัตราเร็วในการรับหรือส่งข้อมูลกำหนดที่ Timer 1, 2

SM2 บิตเลือกการทำงานแบบ Single Processores Mode หรือ Multiprocessors Mode

- 1 : เลือก Multiprocessors Mode ใช้ได้กับโหมด 2,3
- 0 : เลือก Single Processores Mode ใช้ได้กับทุกโหมด

เมื่อเลือกการทำงานรับข้อมูลแบบ Multiprocessors Mode แล้ว

- ถ้าข้อมูลบิตที่ 9 ที่รับได้มีค่าเป็น 1 RI จะเซ็ท
- ถ้าข้อมูลบิตที่ 9 ที่รับได้มีค่าเป็น 0 RI จะไม่เซ็ท

REN (Receive Enable) บิตควบคุมให้รับหรือไม่รับข้อมูล

- 1 : ให้รับข้อมูลได้
- 0 : ห้ามรับข้อมูล

TB8 (Transmit bit D8) ข้อมูลบิตที่ 9 ที่จะส่งออกไปในโหมด 2, 3 ให้ใส่ในบิตนี้ได้เลย

RB8 (Receive bit D8) ข้อมูลบิตที่ 9 ที่รับเข้ามาจะมากับในบิตนี้

เอกสารนี้เป็นเอกสารของหอสมุดกลางพระจอมเกล้าลาดกระบัง ขอสงวนสิทธิ์ในอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงแก้ไข หรือเผยแพร่ข้อมูลใดๆ อย่างอสังถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(ข้อมูลบิตที่ 9 ก็คือค่าใน TB8 ทางด้านส่งนั่นเอง)

TI แฟล็กซ์ TI จะเป็น 1 เมื่อสิ้นสุดการส่งข้อมูล 1 ไบต์

RI แฟล็กซ์ RI จะเป็น 1 เมื่อรับข้อมูลเสร็จ 1 ไบต์ (บิต RI, TI ผู้เขียนโปรแกรมจะต้องเคลียร์เอง)

Baud Rate	Fosc	SMOD ใน PCON	TIMER 1		
			C/T	MODE	Reload Value
(MODE 0) Max : 1 MHz	12 MHz	X	X	X	X
(MODE 2) Max : 375 KHz	12 MHz	1	X	X	X
(MODE2) Min : 187.5 KHz	12 MHz	0	X	X	X
<b>MODE 1, 3</b> : 62.5 K	12 MHz	1	0	2	FFH
19.2 K	11.059 MHz	1	0	2	FDH
9.6 K	11.059 MHz	0	0	2	FDH
4.8 K	11.059 MHz	0	0	2	FAH
2.4 K	11.059 MHz	0	0	2	F4H
1.2 K	11.059 MHz	0	0	2	E8H
137.5	11.059 MHz	0	0	2	1DH
110	6 MHz	0	0	2	72H
110	12 MHz	0	0	1	FEEBH

ตารางที่ 2.4 ตารางการใช้ ไทม์เมอร์ 1 กำหนด บอดเรท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 3

### การคำนวณและการสร้าง

สำหรับการคำนวณและการสร้างนั้น เราสามารถแบ่งออกเป็นส่วนต่างๆ ได้ดังนี้

#### 1. ที่ตัวหุ่นยนต์

- แหล่งจ่ายแรงดัน
- วงจรควบคุม
- Software

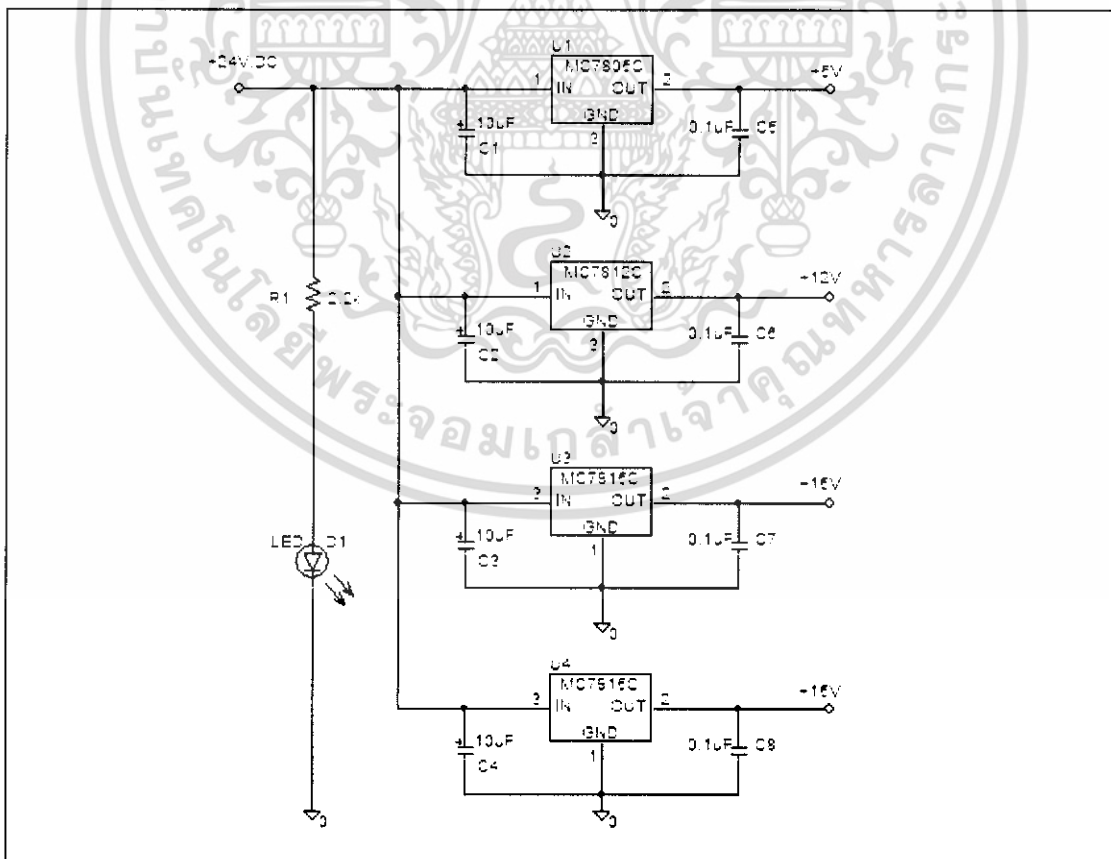
#### 2. สถานีควบคุม

- RF Module

#### 3. Software ที่ใช้ในการควบคุมการส่งข้อมูลที่ตัวของสถานีส่ง

### 3.1 การคำนวณออกแบบ

#### 3.1.1 แหล่งจ่ายแรงดัน



รูปที่ 3.1 วงจรแหล่งจ่ายแรงดัน

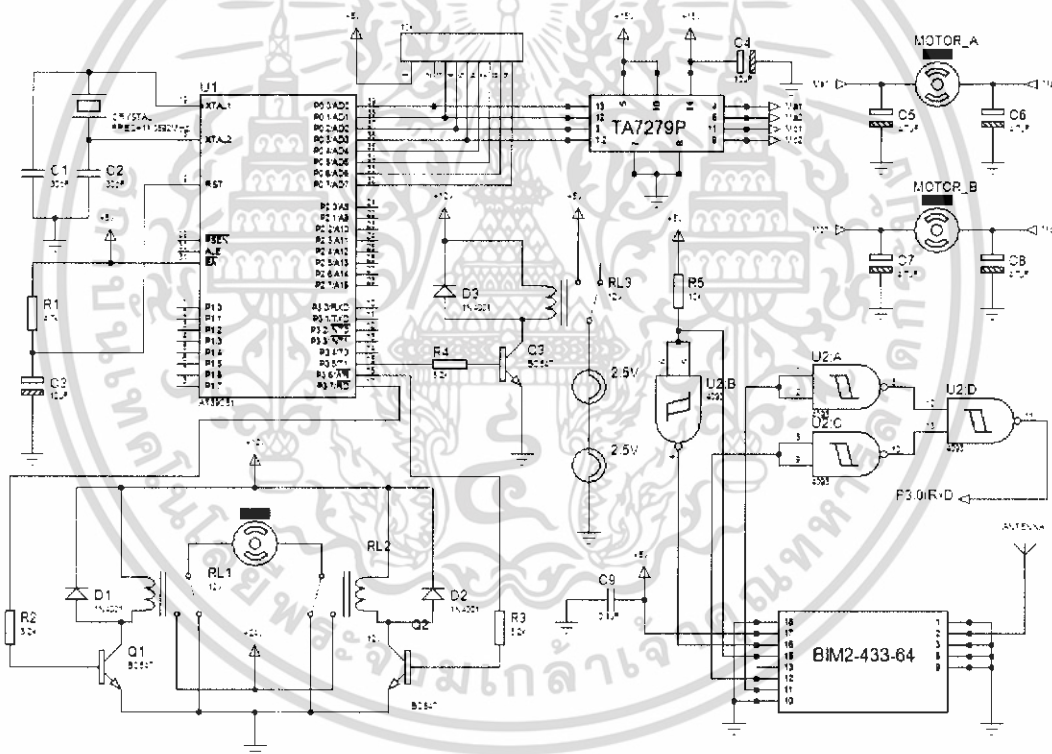
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.1.1 การทำงานของวงจร

ในส่วนของวงจรแหล่งจ่ายแรงดันนั้น ง่ายมากสำหรับการออกแบบเพราะเนื่องจากว่าเรามีไอซีสำเร็จรูปที่สามารถจำกัดแรงดันได้ตามที่เราต้องการโดยต่ออุปกรณ์ภายนอกเพิ่มอีกไม่กี่ตัว ก็สามารถใช้งานได้อย่างมีประสิทธิภาพ

จากวงจรจะเห็นได้ว่าแรงดันที่เราต้องการนั้นคือ 5V ,12V ,15V สังเกตว่าที่แรงดัน 15V นั้นเราใช้ไอซีถึง 2 ตัวด้วยกัน ที่เป็นแบบนี้ก็เพราะว่าเรานำเอาแรงดัน 15V นี้ไปขับมอเตอร์ที่ทำหน้าที่ในการขับเคลื่อนตัวของหุ่นยนต์ ซึ่งต้องการกระแสที่สูง ฉะนั้นเราจึงแบ่งจ่ายออกเป็น 2 ตัว ตัวหนึ่งจ่ายให้กับมอเตอร์ทางด้านซ้าย อีกตัวหนึ่งจ่ายให้มอเตอร์ทางด้านขวา โดยที่คาปาซิเตอร์ 10 $\mu$ F ทำหน้าที่กรองแรงดันให้เรียบอีกครั้งหนึ่ง ส่วนคาปาซิเตอร์ 0.1 $\mu$ F ทำหน้าที่กรองสัญญาณรบกวนที่อาจจะเกิดขึ้นได้ในวงจร โดยมี R1 ทำหน้าที่จำกัดกระแสและแรงดันให้กับ LED ซึ่งเป็นตัวแสดงผลการทำงานของของวงจร

### 3.1.2 วงจรควบคุม



รูปที่ 3.2 วงจรสมบูรณของตัวหุ่นยนต์

#### 3.1.2.1 การทำงานของวงจร

ในวงจรรูปที่ 3.2 เป็นวงจรทั้งหมดในส่วนของตัวหุ่นยนต์ (ไม่รวมแหล่งจ่ายไฟ) โดยมีไมโครคอนโทรลเลอร์เป็นตัวจัดการประมวลผลข้อมูล รวมทั้งควบคุมอุปกรณ์ต่างๆ ภายในตัวหุ่นยนต์ให้ทำงานประสานกันอย่างลงตัว โดยเราแบ่งหน้าที่ต่างๆ ของวงจรถูกออกเป็นส่วนย่อยคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Main CPU
- Motor Control
- Arm Control
- Light Control
- RF Module

หมายเหตุ : ดูรูปวงจรถนาคใหญ่ได้จากในส่วนของภาคผนวก

### 3.1.2.1.1 Main CPU

ส่วนนี้ทำหน้าที่ในการควบคุมและประมวลผลทั้งหมดของหุ่นยนต์โดยใช้ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ซึ่งเป็นไอซีประเภทที่เราสามารถเขียนโปรแกรมควบคุมสั่งงานได้ตามที่เราต้องการ R1 และ C3 ทำหน้าที่ในการ Reset การทำงานของ MCS-51 ในครั้งแรกเพื่อให้ MCS-51 ทำงานเริ่มต้นตั้งแต่บรรทัดแรกของโปรแกรม Crystal 11.0592 MHz ,C1 และ C2 ทำหน้าที่ในการกำเนิดความถี่เพื่อกำหนดจังหวะการทำงานให้กับ MCS-51 โดยเราใช้ Port 0 ทำหน้าที่ควบคุมการหมุนของมอเตอร์ที่ใช้ขับเคลื่อนตัวหุ่นยนต์ผ่านไอซี TA7972P ซึ่งเป็นไอซีควบคุม DC Motor โดยเฉพาะ โดยที่ตารางการควบคุมสามารถดูได้จากบทที่ 2 ในหัวข้อที่ 2.3.2 ส่วน Port 3 ใช้ควบคุมการขับเคลื่อนในส่วนของ Arm Control ,Light Control ,RF Module

### 3.1.2.1.2 Motor Control

ในส่วนนี้จะประกอบไปด้วยไอซี TA7279P ซึ่งเป็นไอซีที่ใช้ควบคุม DC Motor โดยเฉพาะ โดยการป้อนลอจิกทางด้าน input เพื่อใช้ควบคุม โดยที่สามารถควบคุมได้ถึง 2 ตัวด้วยกัน สังเกตว่า C5 ,C6 ,C7 ,C8 นั้นต่อไว้เพื่อป้องกันสัญญาณรบกวนที่อาจจะเกิดขึ้นจากมอเตอร์

### 3.1.2.1.3 Arm Control

ในส่วนนี้เราสามารถอธิบายได้ว่าเมื่อให้ลอจิก 1 ที่ขาเบสของ Q1 จะทำให้ Q1 นำกระแส ซึ่งจะส่งผลให้รีเลย์ทำงาน หน้าสัมผัสภายในจะสลับมาที่กราวด์ ส่วน Q2 เราให้ลอจิกเป็น 0 ซึ่งจะทำให้ Q2 อยู่ในสถานะคัทออฟ คือทรานซิสเตอร์ไม่ทำงาน ส่งผลให้รีเลย์ไม่ทำงานด้วย หน้าสัมผัสภายในยังคงอยู่ที่ +24V ทำให้กระแสไฟฟ้าไหลจาก +24V ของ Q2 ไหลผ่านมอเตอร์มายังกราวด์ทางด้านของ Q1 ทำให้มอเตอร์หมุน

ในทางกลับกัน หากให้ลอจิก 1 ที่ขาเบสของ Q2 และลอจิก 0 ที่ขาเบสของ Q1 จะทำให้มอเตอร์หมุนในทิศทางตรงกันข้ามกับเมื่อซักรุ่นนี้ ด้วยหลักการเดียวกันนั่นเอง

สำหรับในเรื่องของปริมาณโหลดที่วงจรนี้สามารถรับได้นั้น ขึ้นอยู่กับขนาดของรีเลย์ทางด้านคุณสมบัติทางเอาท์พุทของรีเลย์เองว่า สามารถทนแรงดันและกระแสได้สูงสุดเท่าไร ซึ่งไม่เกี่ยวกับทางด้านขดลวดซึ่งทำหน้าที่เหนี่ยวนำให้ทำงาน เพราะว่าในส่วนนี้เป็นส่วนที่เกี่ยวกับการควบคุมให้รีเลย์ทำงานหรือไม่ทำงานเท่านั้นเอง ซึ่งโดยส่วนใหญ่จะขึ้นอยู่กับค่าแรงดันที่ใช้ในการเหนี่ยวนำขดลวด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้ผู้ใดเห็นประโยชน์ประการใด กรุณาแจ้งกลับมาที่ [www.kit.com](mailto:info@www.kit.com) ไม่ว่ากรรมใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภายในโดยทั่วไปจะมีค่าไม่สูงมากนัก เราสามารถใช้ทรานซิสเตอร์มาควบคุมได้อย่างง่ายดาย ในวงจรมีขนาดที่ต่ำมาก ๆ สำหรับงานควบคุมระดับนี้ โดยที่ไดโอดที่ต่ออยู่นั้นทำหน้าที่ป้องกันกระแสไหลย้อนกลับ อันเนื่องมาจากการยุบตัวของสนามแม่เหล็กที่เกิดขึ้นในขณะที่ทรานซิสเตอร์หยุดทำงานโดยฉับพลัน ซึ่งทำให้เกิดกระแสไหลที่สูงมาก ๆ เพียงเสี้ยววินาที ซึ่งอาจทำให้ทรานซิสเตอร์ของเราพังได้

สำหรับการคำนวณหาค่า R2 และ R3 นั้นคำนวณได้จากสูตร

$$Rb = \frac{(Vb - Vbe)\beta}{Ic}$$

เมื่อ Rb คือ ความต้านทานที่ขาเบสที่ต้องการ

Vb คือ แรงดันที่ขาเบส = 5V

Vbe คือ ค่าแรงดันระหว่างขาเบสกับขาคอนโทรล = 0.6V

$\beta$  คือ อัตราการขยายกระแสของทรานซิสเตอร์ = 130

Ic คือ กระแสที่ขาคอนโทรล = 70 mA

เราได้  $Rb = \frac{(5 - 0.6)130}{0.07}$

$$Rb = 8.171k\Omega$$

เราเลือกใช้ค่าที่ใกล้เคียงที่สุดคือ 8.2 k $\Omega$

#### 3.1.2.1.4 Light Control

สำหรับในส่วนของการควบคุมไฟนั้น ถ้าสังเกตจะพบว่ารูปแบบการต่อวงจรนี้มีลักษณะคล้ายคลึงกับส่วนของการขับเคลื่อนแขนหุ่นยนต์ เพราะฉะนั้นการทำงานและการคำนวณจึงเหมือนกัน เราจึงขอไม่อธิบายในส่วนนี้

#### 3.1.2.1.5 RF Module

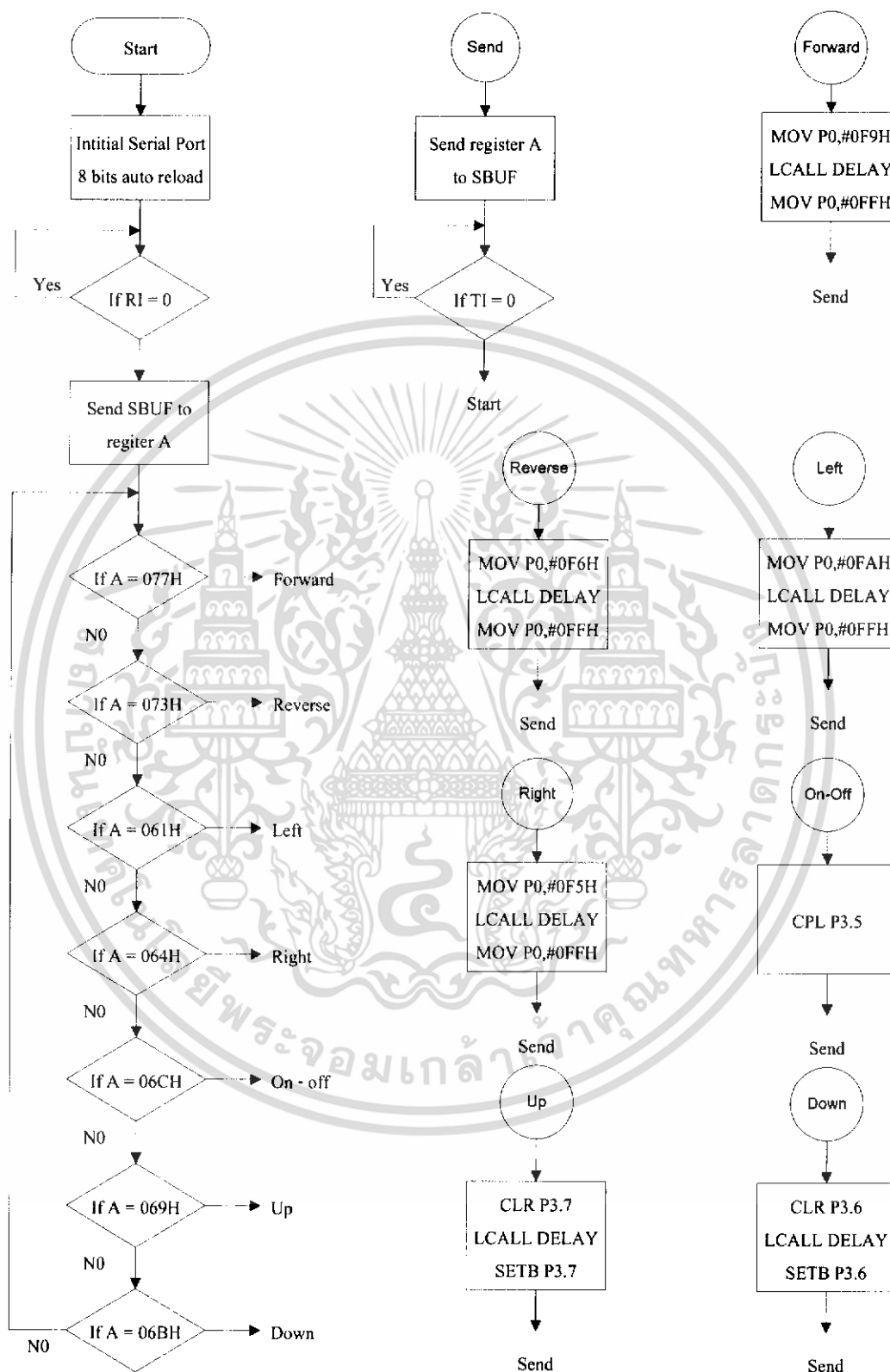
สำหรับในส่วนนี้จะเรียกว่าเป็นพระเอกของงานเลยก็ว่าได้ เนื่องจากว่าโมดูลตัวนี้ทำหน้าที่ได้ทั้งรับและส่งข้อมูลได้ในตัวเดียวกัน โดยข้อมูลสามารถส่งได้ความเร็วสูงโดยเบอร์ที่นำมาใช้งานนี้สามารถส่งได้ถึง 64 kbps โดยจากวงจรเราสามารถตั้งให้โมดูลทำหน้าที่รับหรือส่งได้ สำหรับที่ตัวหุ่นยนต์นั้นเราทำหน้าที่เป็นภาครับ โดยรับข้อมูลคำสั่งที่ออกมาจากโมดูลตัวส่งที่คอมพิวเตอร์ โดยมี 4093 CMOS ที่ทำหน้าที่กำหนดการทำงานให้เป็นโหมดการรับเท่านั้น เงื่อนไขการรับหรือส่งนั้นสามารถดูได้จาก Datasheet ด้านหลัง

#### 3.1.3 Software

ในส่วนของ Software นี้เราเขียนเพื่อในการสร้างให้ MCS-51 กลายเป็นสมองกลย่อยๆ ในการรับคำสั่งและประมวลผลส่งงานให้หุ่นยนต์ทำงานตามต้องการ

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของโรงเรียนพระอภัยมณีวิทยานานาชาติสงขลา ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3.1 Flow Chart การทำงานของโปรแกรม



รูปที่ 3.3 Flow Chart การทำงานของ โปรแกรมภายในตัวไมโครคอนโทรลเลอร์ MCS-51

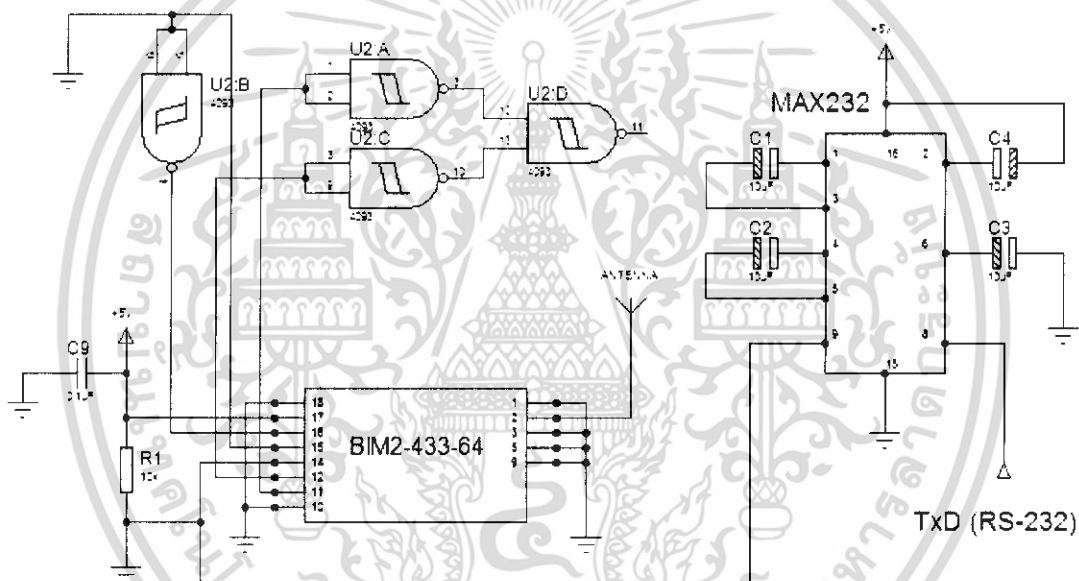
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.3.2 อธิบายการทำงานของโปรแกรมจาก Flow Chart

เริ่มจากอันดับแรก จะทำการกำหนดค่าเริ่มต้นต่างๆ ให้กับ Serial Port โดยกำหนดให้เป็นการส่งแบบ 8 bits Auto Reload ความเร็วการรับส่งที่ 9600 bps จากนั้นทำการตรวจสอบว่ามีข้อมูลเข้ามาแล้วหรือยัง ถ้ามีก็จะทำการเปรียบเทียบค่าที่ได้ กับค่าคงที่ที่ต้องการ หากตรงกันก็ให้กระโดดไปยังตำแหน่งนั้นเพื่อทำการประมวลผล หรือปฏิบัติตามที่ต้องการต่อไป โดยเมื่อทำเสร็จแล้ว ข้อมูลที่ส่งไปจะถูกส่งกลับเข้ามายังคอมพิวเตอร์ที่ทำการส่ง เพื่อเป็นการยืนยันว่าได้ทำการประมวลผลกับคำสั่งนั้นๆ แล้ว

หมายเหตุ : สามารถดูรายละเอียดของ โปรแกรมได้จากภาคผนวก

### 3.1.4 RF Module ในส่วนของสถานีควบคุม

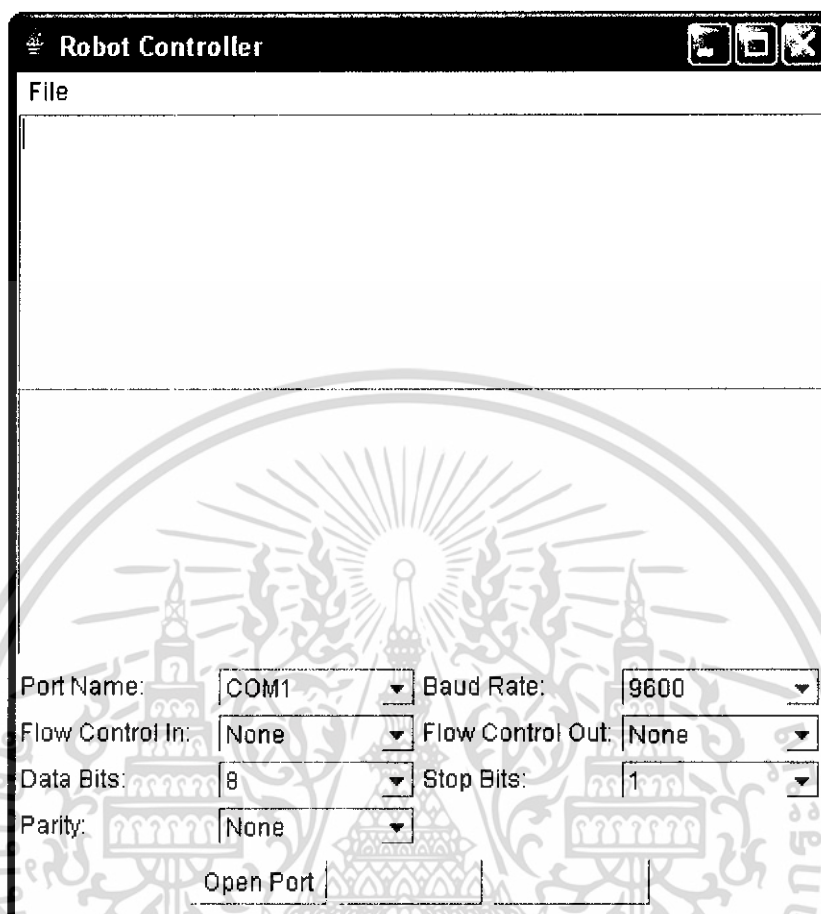


รูปที่ 3.4 วงจรในส่วนของ RF Module ที่สถานีควบคุม

#### 3.1.4.1 การทำงานของวงจร

จากวงจรในรูปที่ 3.4 สังเกตดีๆ จะพบว่าในส่วนของ 4093 CMOS ที่ขา 5,6 ต่อลงกราวด์ เพื่อให้เป็นไปตามเงื่อนไขการตั้งค่าให้เป็นโหมดส่งอย่างเดียว โดยมี MAX232 ทำหน้าที่แปลงระดับสัญญาณจากคอมพิวเตอร์มาเป็นสัญญาณ TTL เพื่อป้อนให้กับ RF Module

### 3.1.5 Software ที่ใช้ในการควบคุมการส่งข้อมูลที่ตัวของสถานีส่ง



รูปที่ 3.5 โปรแกรมที่ใช้ interface กับคอมพิวเตอร์

ใช้สำหรับควบคุมหุ่นยนต์ซึ่งการทำงานจะเหมือนกับโปรแกรม Hyper Terminal ใช้ในการส่งข้อมูลออก Serial Port เพื่อควบคุมหุ่นยนต์ โดยตัวโปรแกรมสามารถเปลี่ยนค่า Port Name, Flow Control In, Flow Control Out, Data Bits, Parity, Baud Rate, และ Stop Bits ที่หน้าจอ Interface โดยตรง และยังสามารถที่จะเก็บค่าต่างได้และเรียกมาใช้งานได้โดยไม่ต้องเปลี่ยนค่าเหล่านั้นใหม่ ทำให้สะดวกเรียกการใช้งานซึ่งการควบคุมจะใช้ตัวอักษรบนคีย์บอร์ดเป็นตัวควบคุมหุ่นยนต์ดังนี้

ตัวอักษร	w	Forward
ตัวอักษร	s	Reverse
ตัวอักษร	a	Turn Left
ตัวอักษร	d	Turn Right
ตัวอักษร	l	Turn On-Off Light
ตัวอักษร	i	Move up Arm
ตัวอักษร	k	Move Down Ann

แนวทางในการพัฒนาขั้นต่อไป สามารถพัฒนาการควบคุมโดยใช้ปุ่มคอนโทรลและ สามารถแสดงภาพผ่านหน้าจออินเทอร์เน็ตเฟสได้ และสามารถที่จะพัฒนาควบคุมหุ่นยนต์ผ่าน Pocket PC ได้ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### การทดลองและผลการทดลอง

สำหรับการทดลองนั้น เราทำการแบ่งการทดลองออกเป็นส่วนต่างๆ ดังนี้

1. ส่วนของการส่งข้อมูล
2. ส่วนของการส่งข้อมูลภาพและเสียง
3. กระแสที่มอเตอร์ใช้ ในขณะที่ขับเคลื่อน

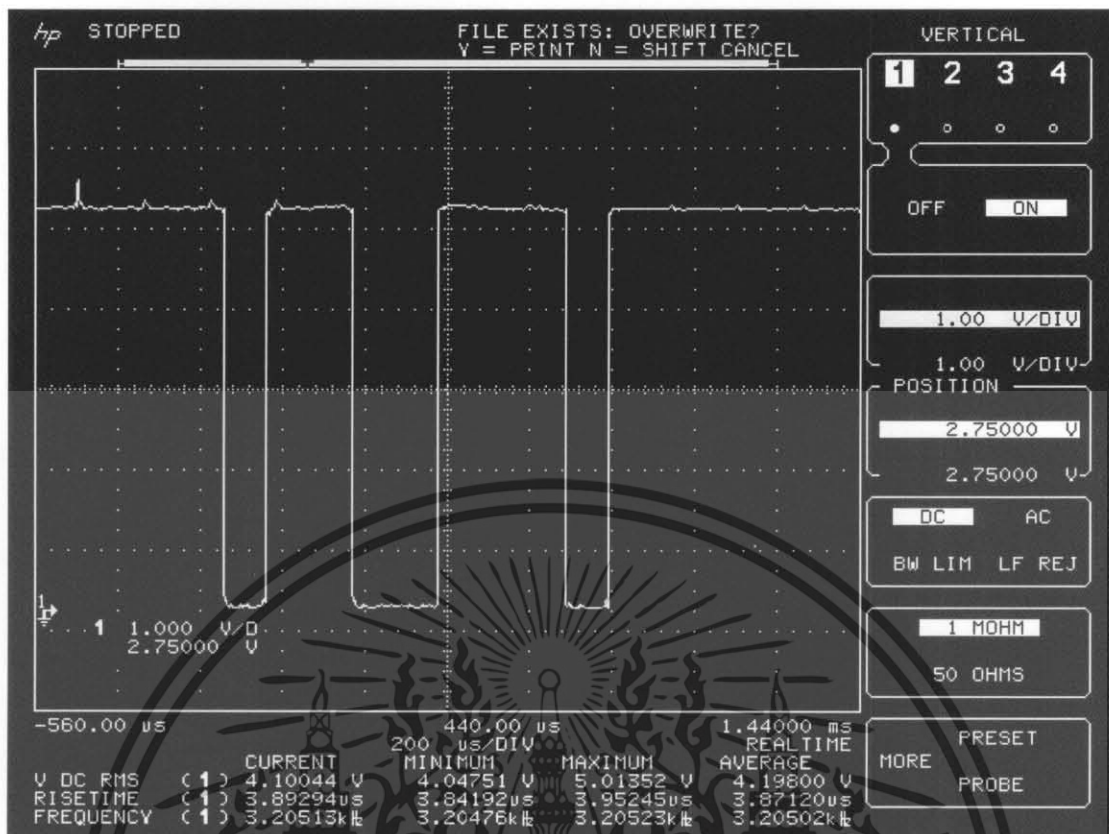
#### 4.1 การทดลองการส่งข้อมูล

เราใช้วิธีการส่งข้อมูล โดยส่งรหัส ASCII ออกมาจากคอมพิวเตอร์แล้วส่งไปยัง MAX232 ซึ่งทำหน้าที่แปลงระดับสัญญาณจากคอมพิวเตอร์ ไปเป็นสัญญาณ TTL เพื่อใช้ส่งให้กับ MSC-51 ในส่วนของตัวหุ่นยนต์ โดยในการทดลองนี้ เราวัดสัญญาณในรูปแบบของสัญญาณ TTL ออกมาจากขา 9 ของ MAX232 ได้รูปสัญญาณดังนี้ โดยเราสามารถเปรียบเทียบค่าต่างๆ ได้จากตารางที่ 4.1



รูปที่ 4.1 สัญญาณที่วัดได้จากขา 9 ของ MAX232 เมื่อกดตัวอักษร “w” บนแป้นคีย์บอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

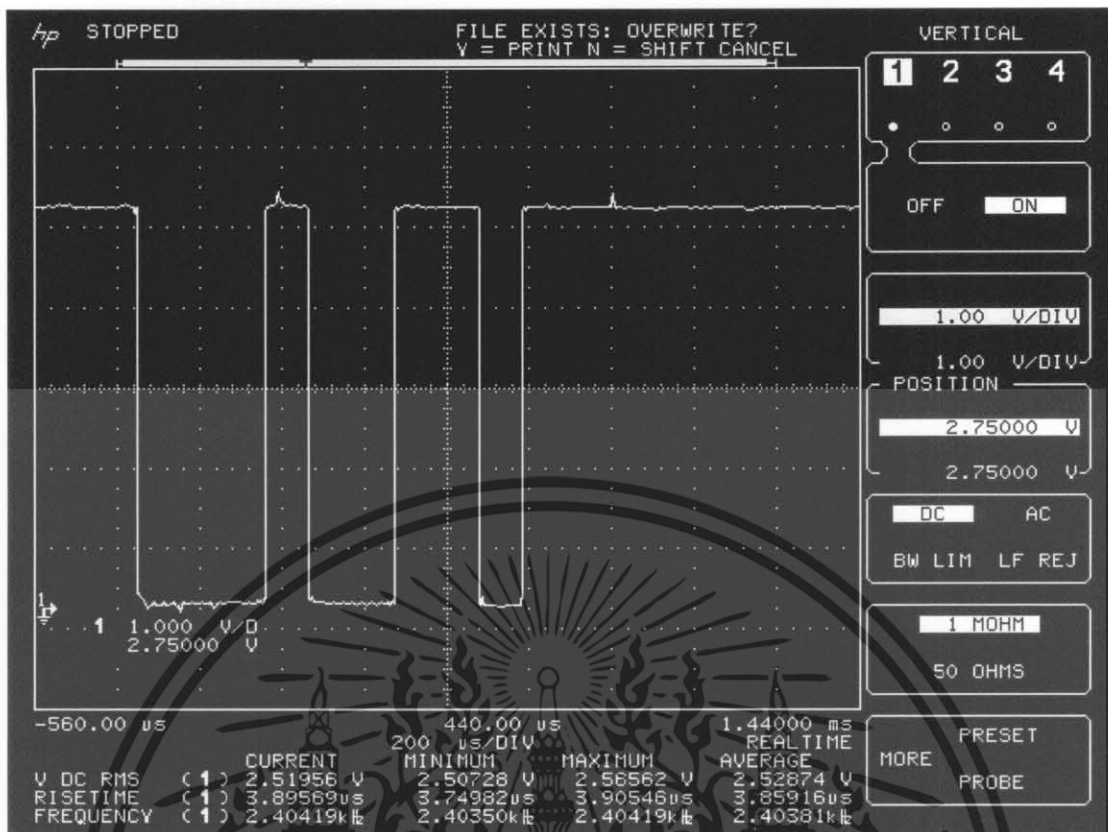


รูปที่ 4.2 สัญญาณที่วัดได้จากขา 9 ของ MAX232 เมื่อกดตัวอักษร “s” บนแป้นคีย์บอร์ด



รูปที่ 4.3 สัญญาณที่วัดได้จากขา 9 ของ MAX232 เมื่อกดตัวอักษร “a” บนแป้นคีย์บอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

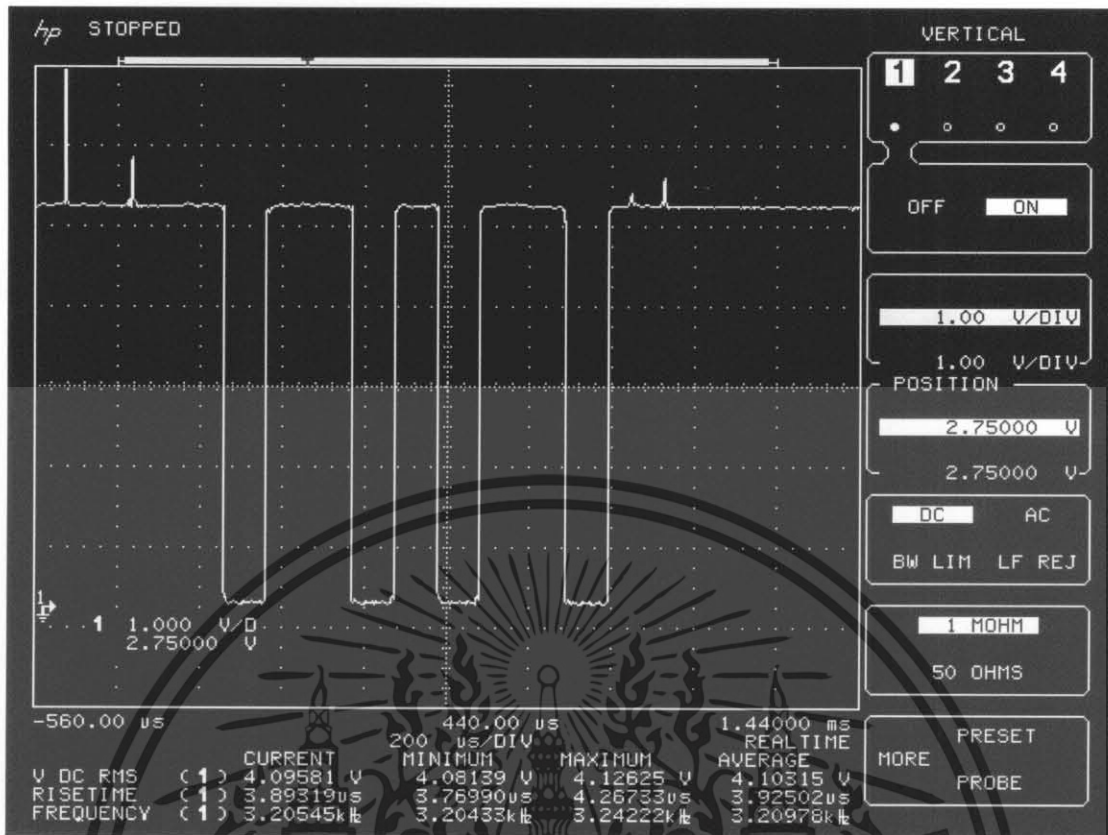


รูปที่ 4.4 สัญญาณที่วัดได้จากขา 9 ของ MAX232 เมื่อกดตัวอักษร “d” บนแป้นคีย์บอร์ด



รูปที่ 4.5 สัญญาณที่วัดได้จากขา 9 ของ MAX232 เมื่อกดตัวอักษร “i” บนแป้นคีย์บอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 สัญญาณที่วัดได้จากขา 9 ของ MAX232 เมื่อกดตัวอักษร “k” บนแป้นคีย์บอร์ด



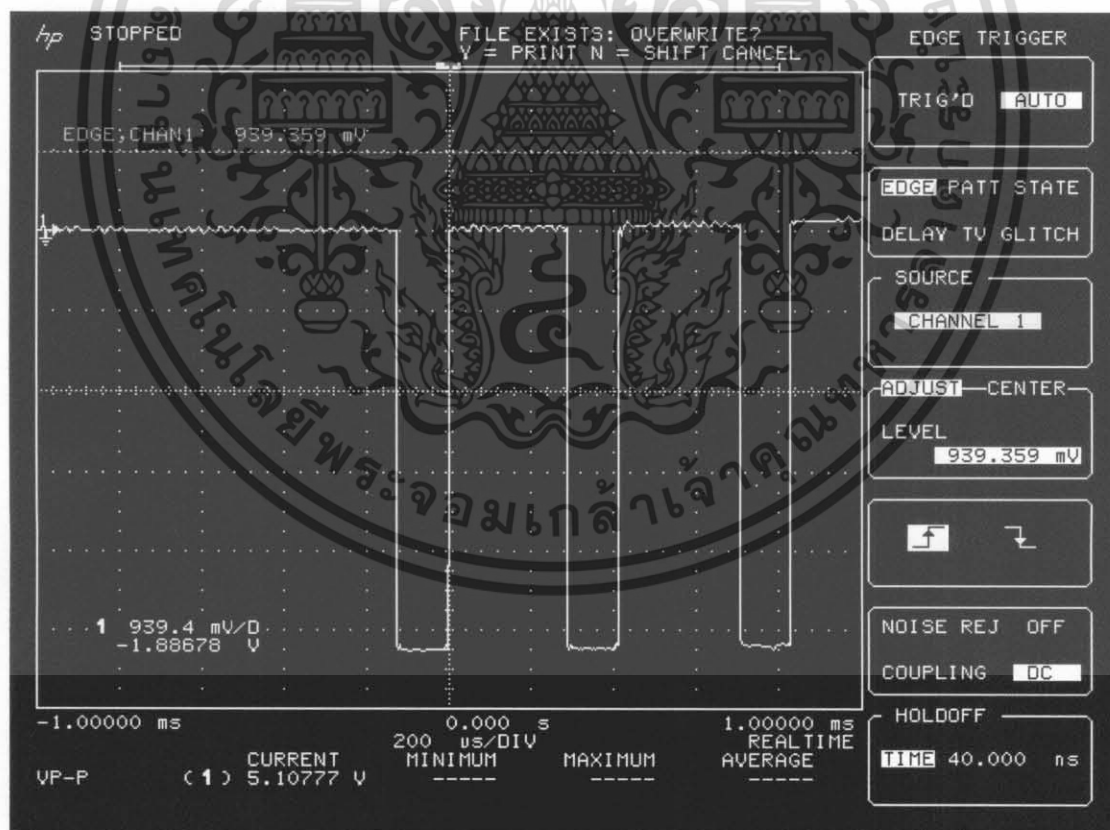
รูปที่ 4.7 สัญญาณที่วัดได้จากขา 9 ของ MAX232 เมื่อกดตัวอักษร “l” บนแป้นคีย์บอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 ค่าของตัวอักษรในรูปแบบของเลขฐานต่างๆ

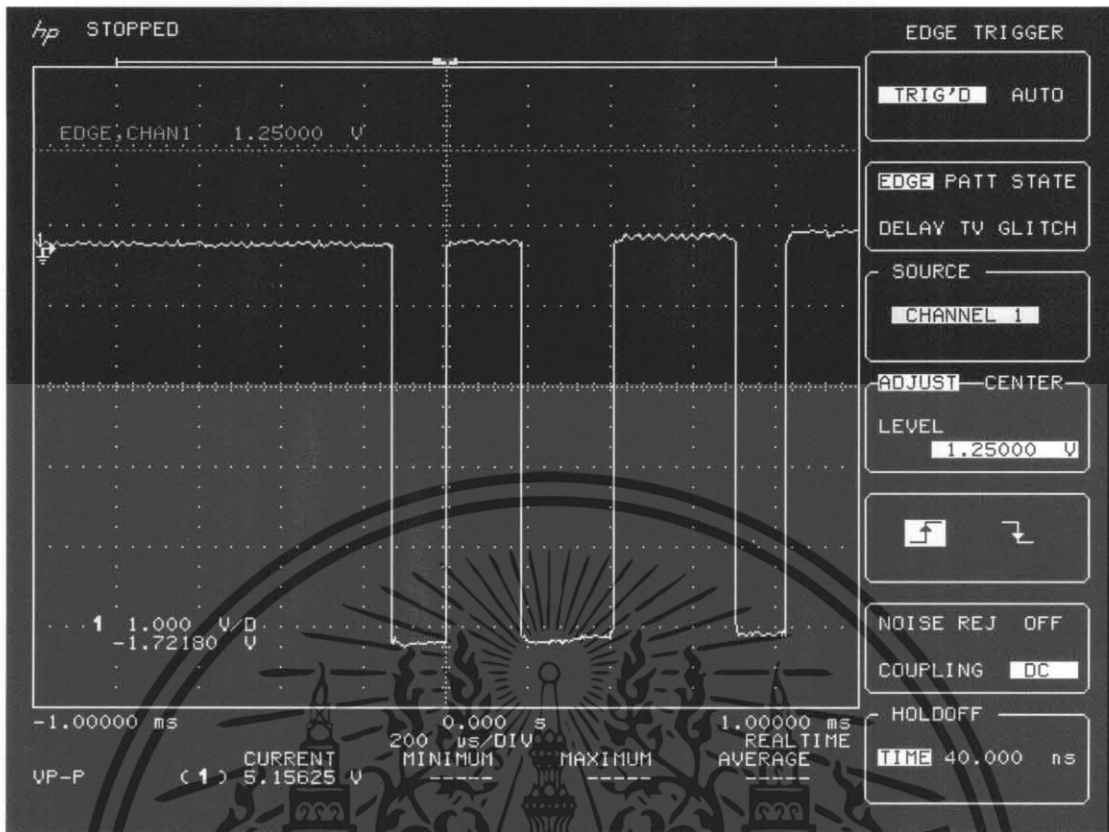
ตัวอักษรบนแป้นคีย์บอร์ด	รูปแบบของเลขฐาน 16	รูปแบบของสัญญาณที่ส่งออกมา
w	77H	0111 0111
s	73H	0111 0011
a	61H	0110 0001
d	64H	0110 0100
i	69H	0110 1001
k	6BH	0110 1011
l	6CH	0110 1100

โดยเราสามารถจับสัญญาณที่ RF Module ตัวรับที่ตัวหุ่นยนต์ได้ โดยพบว่าแทบจะไม่มีการผิดเพี้ยนของสัญญาณเลยแม้แต่น้อย อันเนื่องมาจากคุณภาพของ RF Module ทั้งสองตัวที่ทำการส่งข้อมูลที่มีความถี่ประมาณ 433MHz ความเร็วการส่งที่ 9600 bps ส่งสัญญาณแบบ FSK ตาม Datasheet ที่ทางบริษัทได้กำหนดไว้ และนี่ก็เป็นอีกเหตุผลหนึ่งที่เราเลือกใช้โมดูลตัวนี้ โดยเราสามารถจับสัญญาณได้ดังรูป

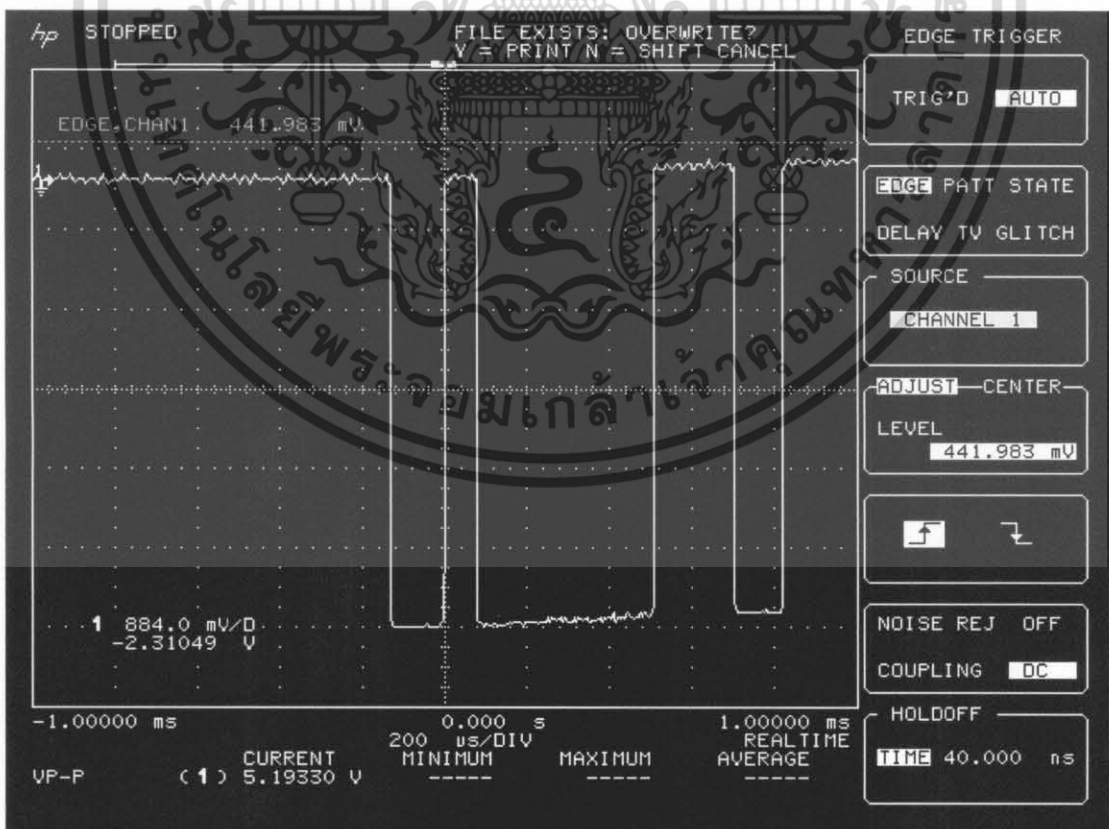


รูปที่ 4.8 สัญญาณที่วัดได้ที่ตัวหุ่นยนต์จากขา 10 ของ MCS-51 เมื่อกดตัวอักษร “w” บนแป้นคีย์บอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

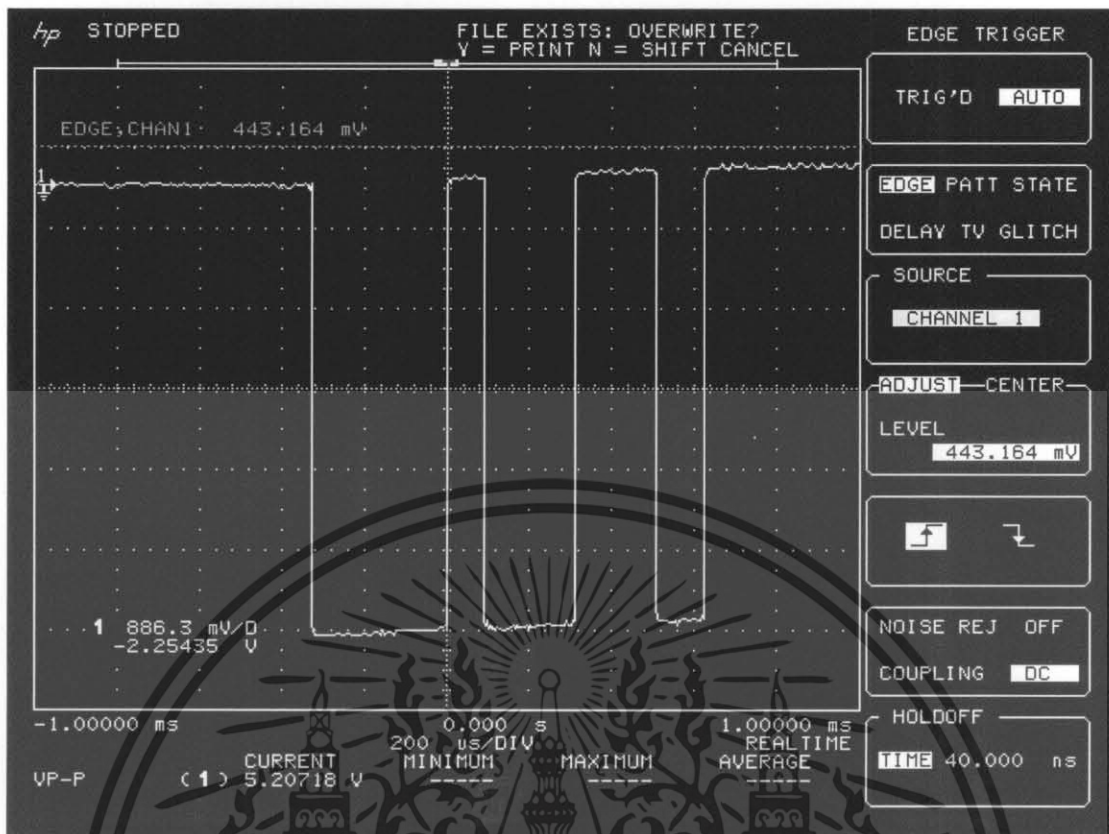


รูปที่ 4.9 สัญญาณที่วัดได้ที่ตัวหุ่นยนต์จากขา 10 ของ MCS-51 เมื่อกดตัวอักษร "s" บนแป้นคีย์บอร์ด

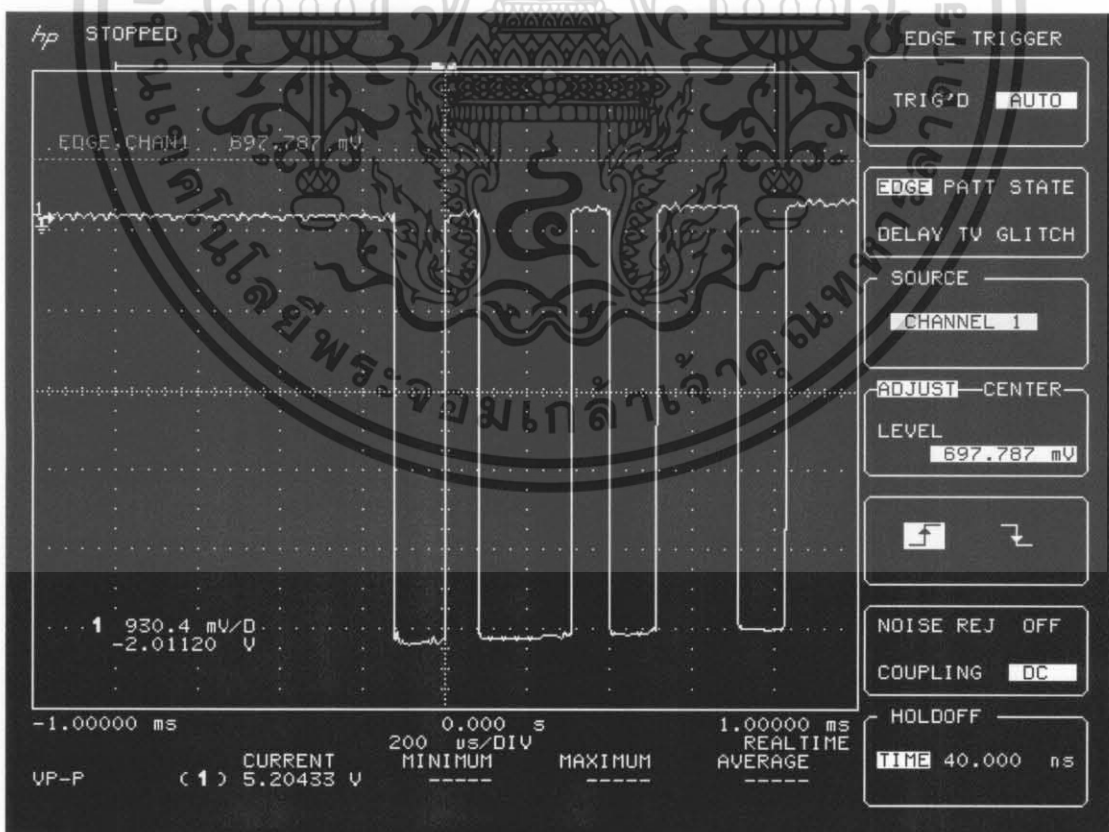


รูปที่ 4.10 สัญญาณที่วัดได้ที่ตัวหุ่นยนต์จากขา 10 ของ MCS-51 เมื่อกดตัวอักษร "a" บนแป้นคีย์บอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

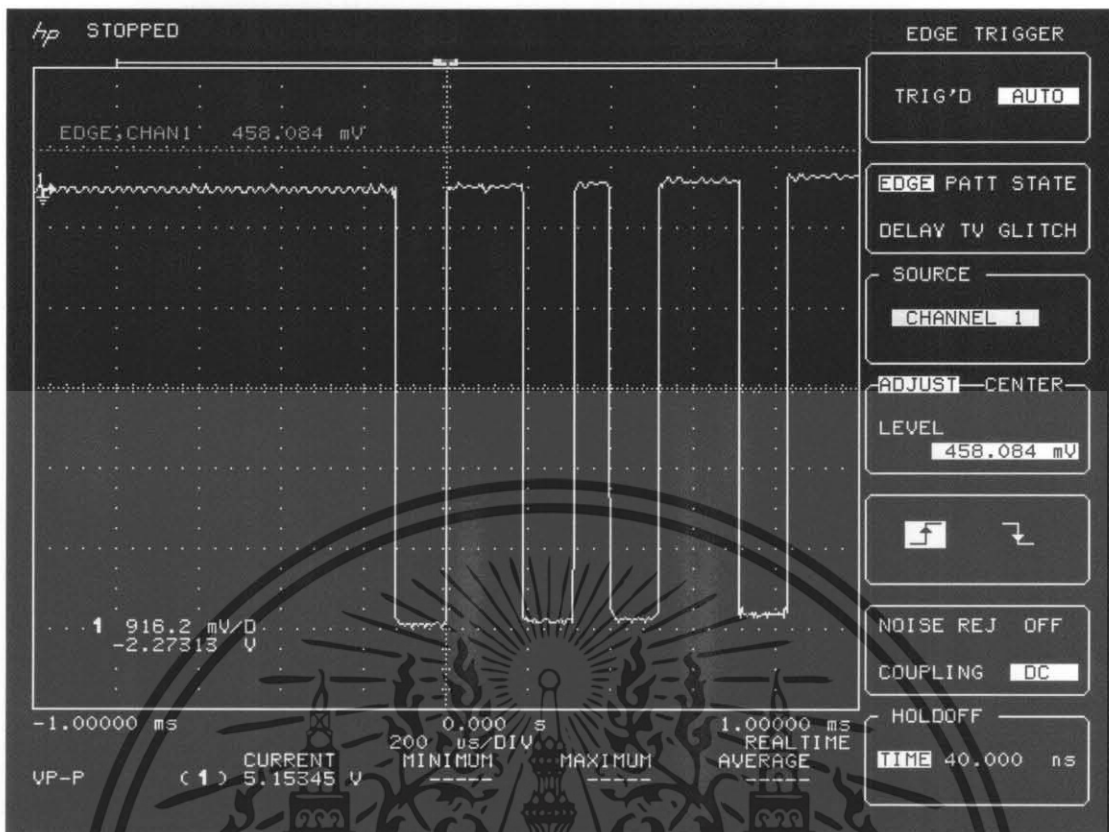


รูปที่ 4.11 สัญญาณที่วัดได้ที่ตัวหุ่นยนต์จากขา 10 ของ MCS-51 เมื่อกดตัวอักษร “d” บนแป้นคีย์บอร์ด

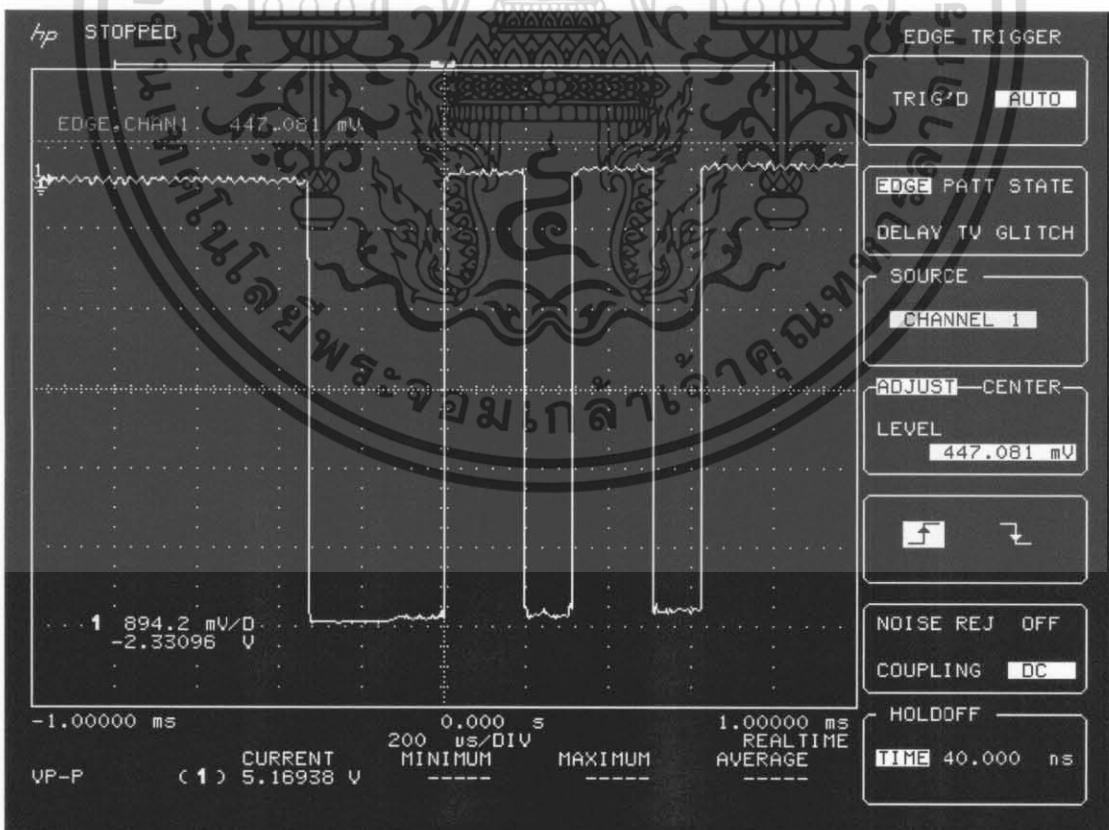


รูปที่ 4.12 สัญญาณที่วัดได้ที่ตัวหุ่นยนต์จากขา 10 ของ MCS-51 เมื่อกดตัวอักษร “i” บนแป้นคีย์บอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

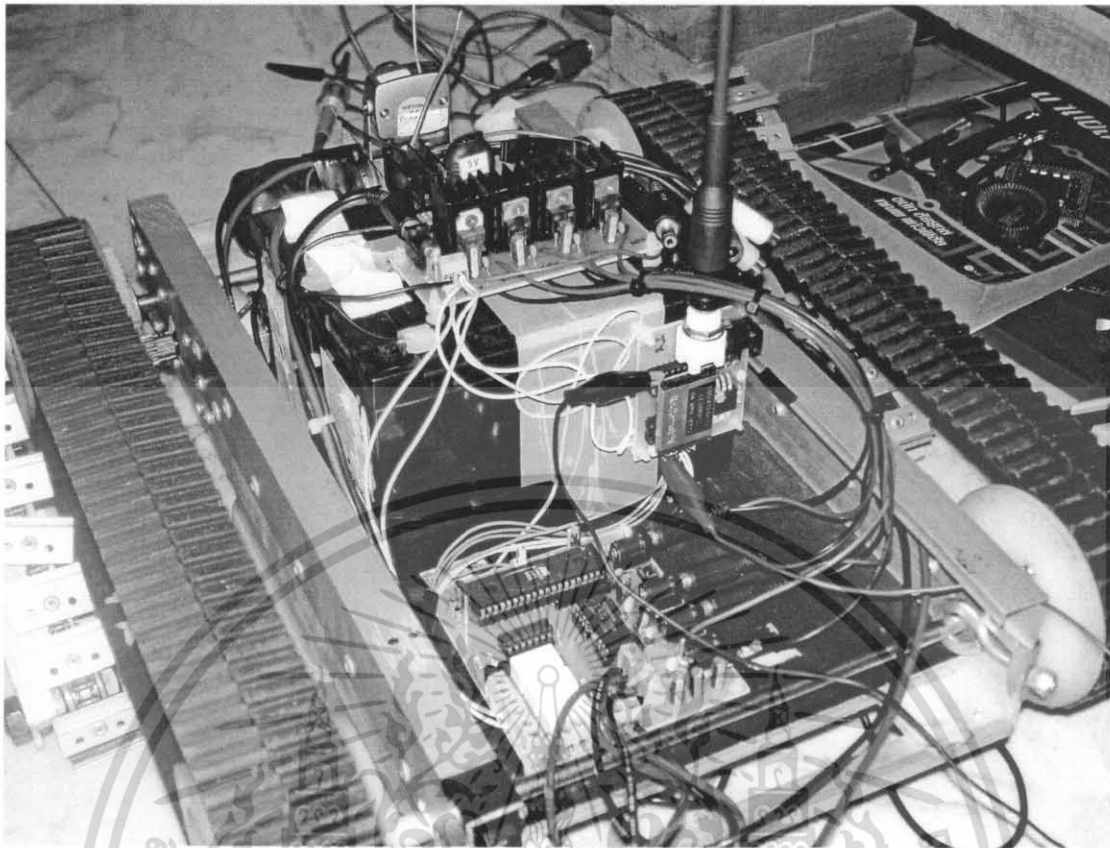


รูปที่ 4.13 สัญญาณที่วัดได้ที่ตัวหุ่นยนต์จากขา 10 ของ MCS-51 เมื่อกดตัวอักษร “k” บนแป้นคีย์บอร์ด

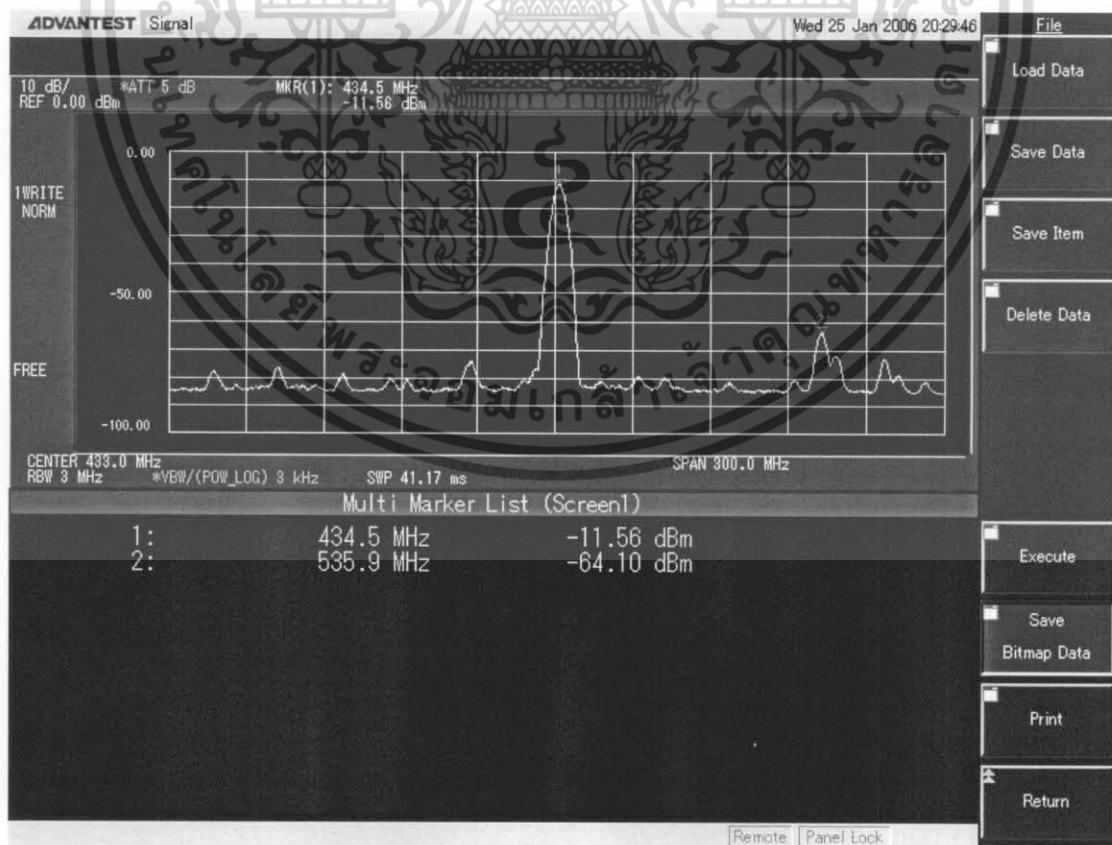


รูปที่ 4.14 สัญญาณที่วัดได้ที่ตัวหุ่นยนต์จากขา 10 ของ MCS-51 เมื่อกดตัวอักษร “1” บนแป้นคีย์บอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

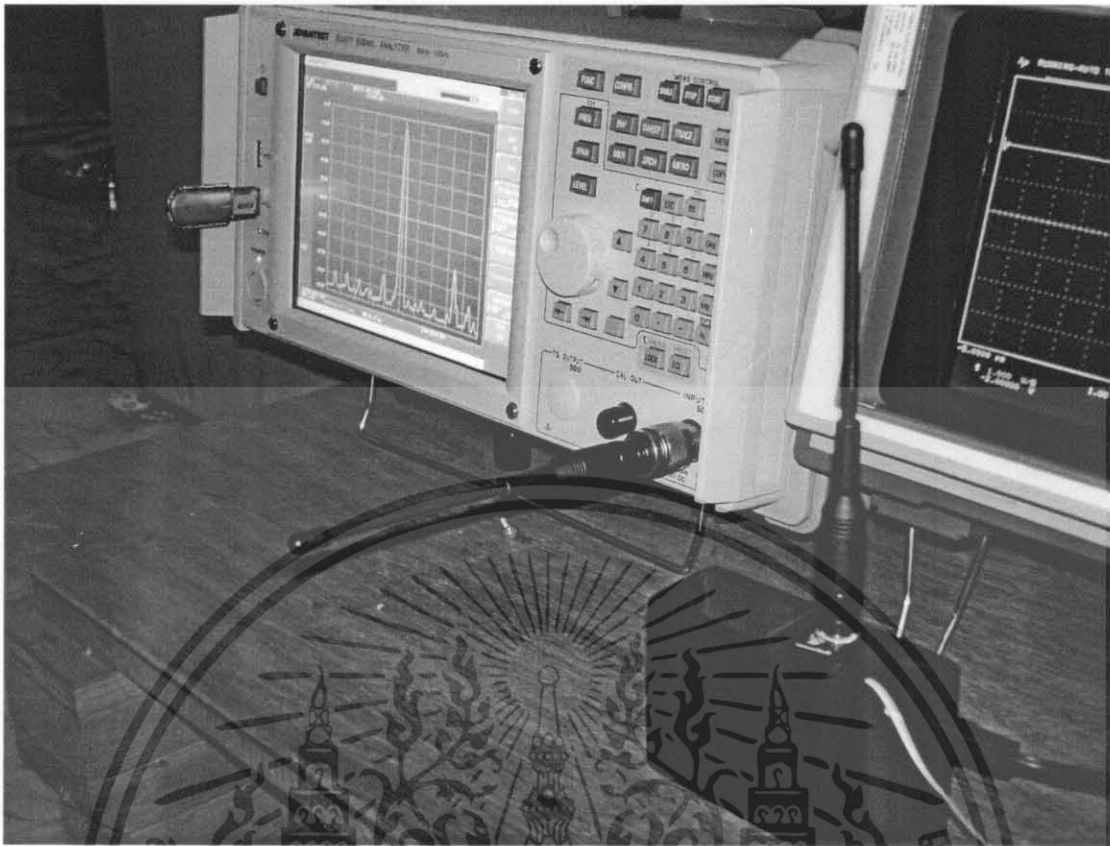


รูปที่ 4.15 การวัดสัญญาณที่รับได้ที่ตัวหุ่นยนต์ผ่าน RF Module



รูปที่ 4.16 Spectrum ที่วัดได้โดย Signal Analysis จากการส่งสัญญาณที่ตัว RF Module

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

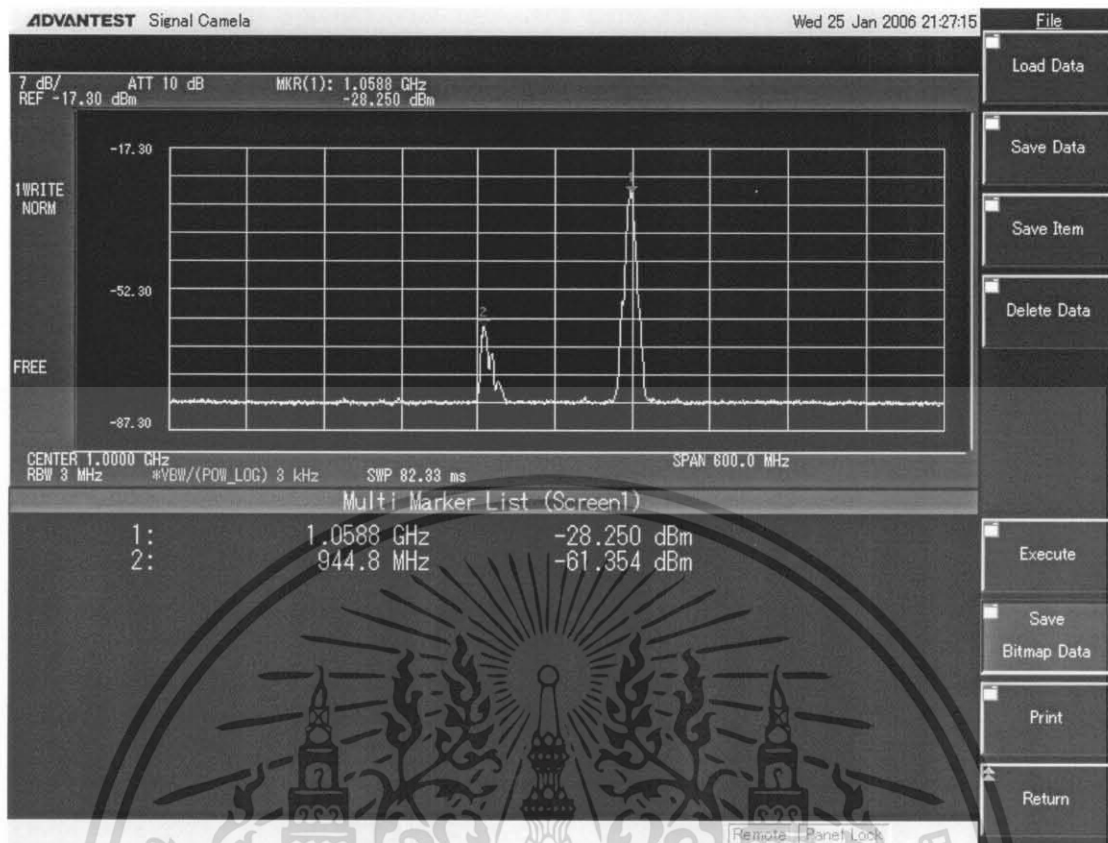


รูปที่ 4.17 การวัด Spectrum ในรูปที่ 4.16

#### 4.2 การทดลองส่งข้อมูลภาพและเสียง

สำหรับในส่วนนี้ง่ายมากสำหรับการติดตั้งและทดสอบ เนื่องจากว่าเราใช้ชุดส่ง Video Sender สำเร็จรูป ซึ่งบอกเราว่าสามารถส่งได้ไกล 50 เมตรในที่โล่ง ส่งด้วยความถี่ 1.2 GHz จากการทดลองพบว่าสามารถส่งได้ประมาณ 20 ถึง 30 เมตรเท่านั้น แต่ว่าสัญญาณภาพและเสียงนั้นจัดได้ว่าชัดเจนดีทีเดียว โดยเราสามารถจับ Spectrum ได้ดังรูปที่ 4.18 และสัญญาณภาพที่รับได้ผ่านหน้าจอกอมพิวเตอร์ดังรูปที่ 4.19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

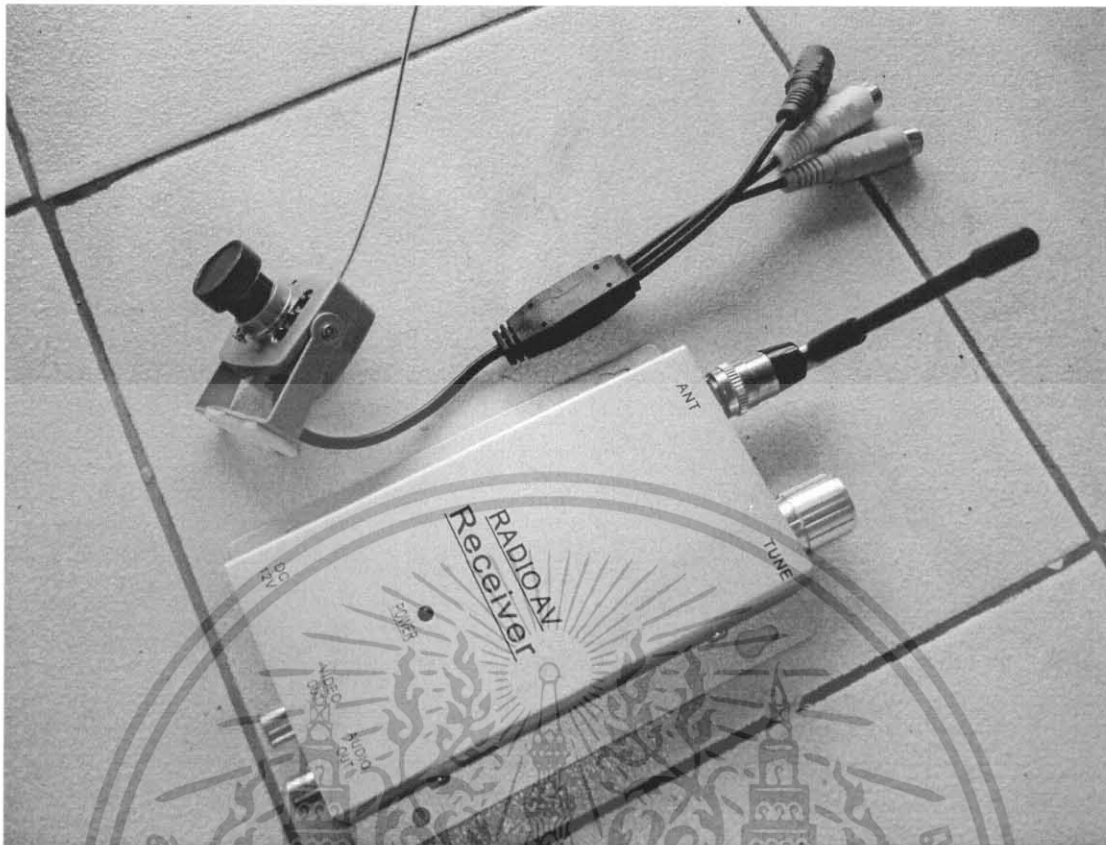


รูปที่ 4.18 Spectrum ที่จับได้จากชุดส่งสัญญาณภาพและเสียง โดยเครื่อง Signal Analysis

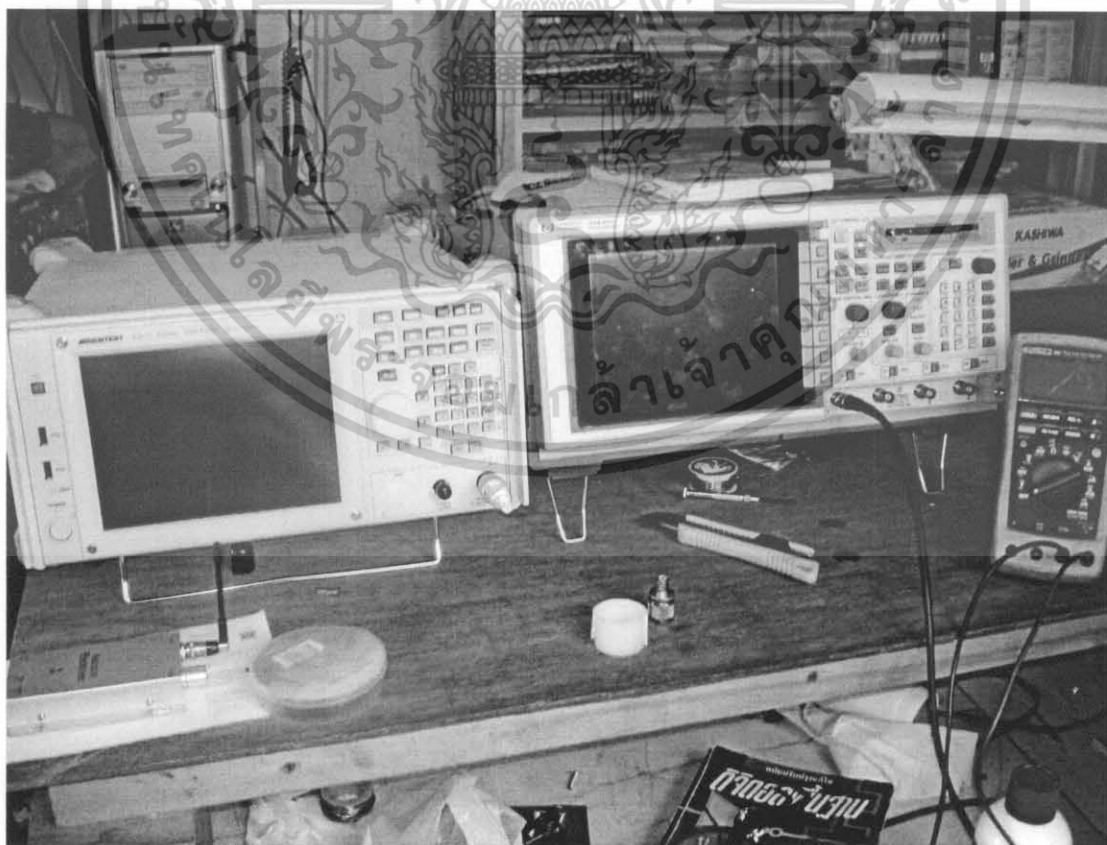


รูปที่ 4.19 สัญญาณภาพที่รับได้ที่หน้าจอคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.20 ชุดอุปกรณ์ Video Sender

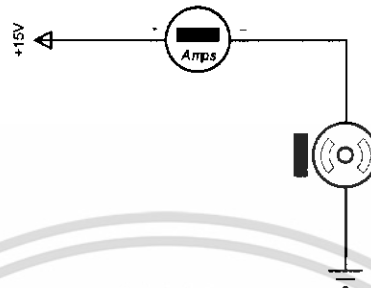


รูปที่ 4.21 อุปกรณ์ที่ใช้ในการทดลองวัดสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

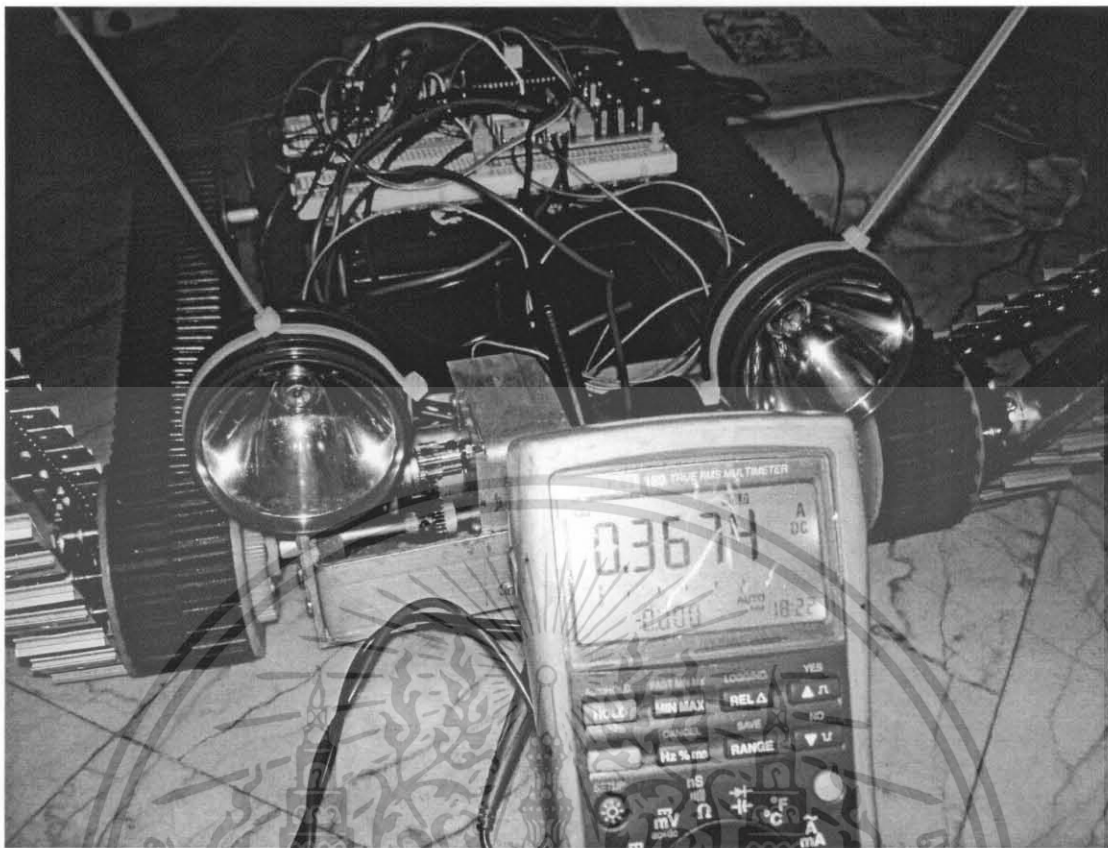
### 4.3 การทดลองวัดค่ากระแสในขณะทำการขับเคลื่อนหุ่นยนต์

ในที่นี้เราจะทำการวัดค่ากระแสไฟฟ้าที่มอเตอร์ขับเคลื่อนใช้ในการใช้งานจริง โดยการต่อแอมป์มิเตอร์อนุกรมกับมอเตอร์ตัวที่ใช้ขับเคลื่อน ดังรูป



รูปที่ 4.22 แสดงการต่อวงจรวัดค่ากระแสที่ตัวมอเตอร์ขับเคลื่อนขณะใช้งานจริง

จากการทดลองในวงจรรูปที่ 4.22 เราทำการขับเคลื่อนหุ่นยนต์ เมื่อได้ค่าของกระแสไฟแล้ว จากนั้นกดปุ่ม Hold บนหน้าปัดเพื่อเป็นการล็อกค่าไม่ให้เปลี่ยนแปลง เราพบว่ามอเตอร์ในระบบขับเคลื่อนขณะที่ใช้งานจริงกินกระแสประมาณ 367.4 mA ต่อหนึ่งตัว เพราะฉะนั้นในการใช้งานมอเตอร์ 2 ตัวพบว่าจะกินกระแสทั้งหมด 734.8 mA และนี่ก็คือเหตุผลหนึ่งที่เราจำเป็นต้องแยกแหล่งจ่ายไฟเป็น +15V ทั้งหมด 2 ตัว เนื่องจากว่าไอซีเรกูเลเตอร์หนึ่งตัวนั้น สามารถจ่ายกระแสได้ประมาณ 1 A สังเกตดีๆ จะพบว่าไอซีสามารถจ่ายกระแสได้ถึง 1 A แต่มอเตอร์ต้องการเพียง 367.4 mA แล้วทำไมต้องใช้ถึง 2 ตัว คำตอบก็คือในขณะที่มอเตอร์เริ่มทำงาน จะมีกระแสไหลในปริมาณที่สูงเพียงชั่วขณะหนึ่ง นั่นอาจทำให้ไอซีเรกูเลเตอร์เสียหายได้หากเราใช้เพียงตัวเดียว



รูปที่ 4.23 ผลการทดลองจากวงจรรูปที่ 4.22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### บทวิจารณ์และสรุปผล

#### 5.1 ปัญหาที่พบ

สำหรับในโครงการชิ้นนี้ในช่วงแรกนั้นมีปัญหามากกับการสร้างในส่วนของตัวหุ่นยนต์ว่าจะทำแบบไหนให้ดูดีและใช้ประโยชน์ได้จริงมากที่สุด ออกแบบมาหลายแบบมาก สุดท้ายก็มาลงตัวที่แบบที่ได้เห็นกัน สำหรับแบบที่เราทำนี้ก็ได้ตัวแบบมาจากกองทัพสหรัฐ นั่นคือ “ Pack Bot ” หุ่นยนต์ที่หน่วยลาดตระเวนของสหรัฐใช้สำรวจพื้นที่ ใช้ช่วยเหลือตัวประกัน ด้วยระบบที่ทันสมัยที่สุด ด้วยงบประมาณที่สูงถึง 17 ล้านดอลลาร์สหรัฐ เมื่อเทียบกับเจ้า FARR ของเราแล้วห่างกันลิบลิ้น ด้วยอุปกรณ์ที่หาได้ในพื้นที่ ด้วยงบประมาณอันน้อยนิด และด้วยเหตุเหล่านี้เองจึงเกิดปัญหาต่างๆ ตามมาและส่งผลโดยตรงกับเจ้า FARR ของเราดังต่อไปนี้

##### 5.1.1 โครงสร้างของหุ่นยนต์

ด้วยความรู้อันน้อยนิดหรือแทบจะไม่มีเลยทางด้าน การออกแบบโครงสร้าง ทำให้เกิดความไม่สมบูรณ์ขึ้นที่ตัวระบบขับเคลื่อน ระบบการป้อนยา ที่ไม่ค่อยจะประสบความสำเร็จนัก อีกทั้งงบประมาณที่ทำให้เราไม่สามารถเลือกวัสดุได้ตามที่เราต้องการ ทำให้เกิดปัญหาในเรื่องของน้ำหนักตัวที่มากถึงประมาณ 15 กิโลกรัม อีกส่วนหนึ่งที่สำคัญมากๆ คือในเรื่องของข้อต่อ และเฟืองต่างๆ มีจะการหลวมบ่อยๆ ทำให้เกิดปัญหาในเรื่องของล้อฟรีเป็นประจำ

##### 5.1.2 ความร้อนที่เกิดขึ้นกับแหล่งจ่ายไฟ

เนื่องจากแหล่งจ่ายไฟขนาด 15V มีการใช้งานสำหรับการขับเคลื่อนหุ่นยนต์ จึงเกิดปัญหาที่แผ่นระบายความร้อนที่มีขนาดเล็กเกินไป ทำให้เกิดความร้อนสูง

##### 5.1.3 ความร้อนที่ตัวไอซีขับเคลื่อนมอเตอร์

เนื่องจากเราไม่ได้คำนึงถึงการระบายความร้อนให้กับมัน จึงทำให้การออกแบบแผ่นวงจรพิมพ์นั้น เราลืมเว้นที่ว่างสำหรับใส่แผ่นระบายความร้อน ถ้าความร้อนสูงมากๆ อาจจะต้องมีการแก้ไขในส่วนนี้เพื่อไม่ให้เกิดปัญหาอื่นๆ ตามมา

##### 5.1.4 สัญญาณจากคอมพิวเตอร์

อีกปัญหาหนึ่งที่ทำให้เราต้องหยุดหงุดหงิดจนใจตลอดเวลาที่ทำการทดลองคือ ในเรื่องของไฟรั่วจากคอมพิวเตอร์ผ่านพอร์ตอนุกรมที่ใช้ในการส่งสัญญาณควบคุมที่มีขนาด -13V ถึง +13V ยังคงเป็นปัญหาที่เราต้องสะอึกไปตามๆ กัน

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 5.1.5 การเชื่อมต่อกับ Pocket PC

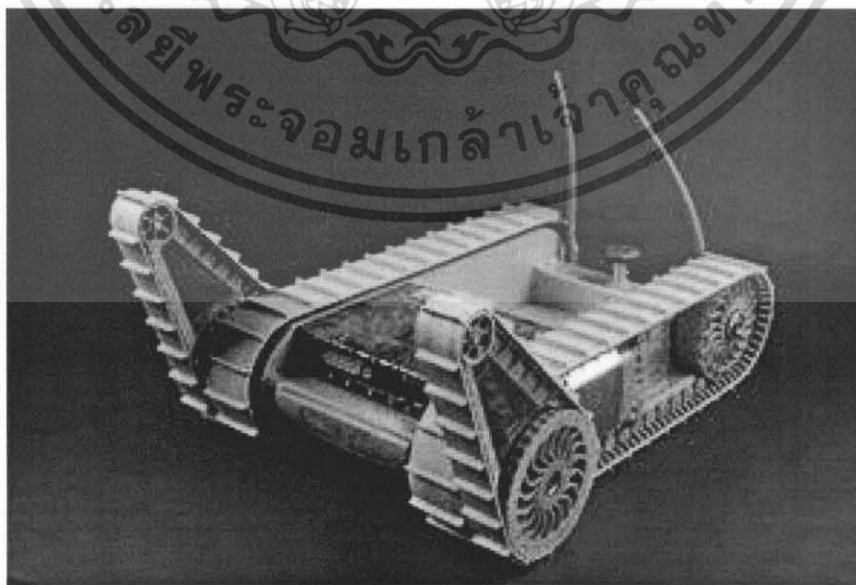
เดิมที่เราต้องการที่จะเชื่อมต่อกับ Pocket PC ให้สามารถควบคุมหุ่นยนต์ได้เหมือนกัน แต่เนื่องจากเราพยายามที่ใช้ JAVA Software ตามที่เราต้องการ เป็นผลให้ไม่สำเร็จเนื่องจากข้อมูลในเรื่องของการเขียนโปรแกรมบน Pocket PC ให้ติดต่อกับอุปกรณ์ภายนอกนั้นหายากมาก ทั้งในเรื่องของ Software ที่ใช้งานและเวอร์ชันของโปรแกรมบน Pocket PC ที่รองรับกับระบบนี้หาไม่ได้

### 5.1.6 สัญญาณรบกวนที่ไม่ทราบสาเหตุ

และสุดท้ายปัญหาที่เราพยายามแก้ไขอยู่ประมาณเกือบหนึ่งเดือน คือในเรื่องของสัญญาณรบกวนที่ทำให้ในบางครั้งที่เราเริ่มเปิดใช้งานที่ตัวหุ่นยนต์ จะเกิดการเคลื่อนที่โดยไม่รู้ทิศทางอยู่ชั่วขณะหนึ่ง หรือในขณะที่เราทำการควบคุมหุ่นยนต์อยู่นั้นก็จะเกิดการดีดคำสั่ง ไม่ยอมทำตามการควบคุมบ้างเป็นบางครั้ง ในขณะที่หากควบคุมโดยตรงผ่านสายสัญญาณจะไม่เกิดปัญหานี้

## 5.2 สรุปผล

สำหรับความสามารถในขั้นสุดท้ายของเจ้า FARR ของเรานั้น เราสามารถพัฒนาให้เป็นหุ่นยนต์ต้นแบบได้ในลักษณะที่มีขีดความสามารถในหลายๆ อย่าง แต่ความสามารถไม่สูงพอที่นำไปใช้งานจริง ความหมายคือทำได้แต่ไม่ถึงขีดสุด เช่น ส่งของมูลแบบไร้สายได้แต่ไม่ไกล ส่งภาพและเสียงได้แต่ไม่ไกล ปีนป่ายตามที่ต่างๆ ได้แต่ไม่ดีนัก สาเหตุจากโครงสร้างไม่เกี่ยวกับตัวระบบ โปรแกรมสามารถ interface กับคอมพิวเตอร์ด้วยภาษา JAVA ได้แต่กราฟิกไม่สวยงาม เป็นต้น



รูปที่ 5.1 รูปร่างหน้าตาของเจ้า Pack Bot

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บรรณานุกรม

1. รองศาสตราจารย์ สมยศ จุณณะปิยะ. การประยุกต์ใช้งานไมโครคอนโทรลเลอร์ **Microcontroller Application**. ภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
2. Gordon McComb แปลและเรียบเรียงโดย ร้อยเอก วาที ปริยพงศ์. เส้นทางสู่นักประดิษฐ์หุ่นยนต์. แปลจากหนังสือ *The Robot Builder's Bonanza* Top Publishing ร่วมกับ สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น), ส.ส.ท.
3. Albert Paul Malvino. **Electronic Principles**, Fourth Edition. McGraw-Hill.
4. James Bignell , Robert Donovan. **Digital Electronics** , Fourth Edition. Delmar Thomson Learning.
5. Stephen J. Bigelow แปลและเรียบเรียงโดย ชัชวาล ศุภเกษม. คัมภีร์ฮาร์ดแวร์ PC ฉบับมืออาชีพ เรียบเรียงจาก *Bigelow's Hardware Desk Reference*. บริษัท ซีเอ็ดดูเคชั่น จำกัด (มหาชน)
6. แปลโดย รอสัม ปริมาส. **21<sup>st</sup> Century Robotics** หุ่นยนต์แห่งศตวรรษที่ 21.สำนักพิมพ์มติชน , ตุลาคม 2547.
7. อรรถพล บุญยะโกศา , วรพจน์ กรแก้ววัฒนกุล และ ชัยวัฒน์ ลิ้มพรจิตรวิไล. เรียนรู้และปฏิบัติการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกผ่านพอร์ตอนุกรม. INEX, Innovative Experiment Co.,Ltd.
8. วรณิกา เนตรงาม. คู่มือการเขียนโปรแกรมภาษา **JAVA** ฉบับผู้เริ่มต้น. บริษัท เอช เอ็น กรุ๊ป จำกัด.
9. ดร. วีระศักดิ์ ชิงฉาวร. **JAVA Programming Volume II**. บริษัท ซีเอ็ดดูเคชั่น จำกัด (มหาชน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก

## โปรแกรมที่ใช้งาน

```

                                P6:  CJNE A,#06BH,OUT
                                LCALL DOWN
                                LJMP OUT
                                OUT:  RET
                                ;-----FORWARD-----
                                FW:  MOV P0,#0F9H
                                LCALL DELAY
                                MOV P0,#0FFH
                                RET
                                ;-----REWARD-----
                                RW:  MOV P0,#0F6H
                                LCALL DELAY
                                MOV P0,#0FFH
                                RET
                                ;-----LEFT-----
                                LL:  MOV P0,#0FAH
                                LCALL DELAY
                                MOV P0,#0FFH
                                RET
                                ;-----RIGTH-----
                                RR:  MOV P0,#0F5H
                                LCALL DELAY
                                MOV P0,#0FFH
                                RET
                                ;-----ON OFF-----
                                ONOFF: CPL P3.5
                                RET
                                ;-----ARM UP-----
                                UP:  SETB P3.6H
                                CLR P3.7
                                LCALL DELAY
                                SETB P3.7
                                RET
                                ;-----ARM DOWN-----
                                DOWN: SETB P3.7
                                CLR P3.6
                                LCALL DELAY
                                SETB P3.6
                                RET
                                LJMP OUT

```

```

ORG 0000H
;-----INITIAL SERIAL PORT---
MOV TMOD,#20H
MOV SCON,#50H
MOV TH1,#0FDH
SETB TR1
;-----MAIN PROGRAM-----
MAIN: LCALL RECV
      LCALL PROC
      LCALL SEND
      LJMP MAIN
;-----DATA RECIVE-----
RECV: NOP
LOOP: JNB RI,LOOP
      CLR RI
      MOV A,SBUF
      RET
;-----DATA SEND-----
SEND: MOV SBUF,A
LOOP1: JNB TI,LOOP1
      CLR TI
      RET
;-----DATA PROCESS-----
PROC: CJNE A,#077H,PO1
      LCALL FW
      LJMP OUT
PO1:  CJNE A,#073H,PO2
      LCALL RW
      LJMP OUT
PO2:  CJNE A,#061H,PO3
      LCALL LL
      LJMP OUT
PO3:  CJNE A,#064H,P4
      LCALL RR
      LJMP OUT
P4:   CJNE A,#06CH,P5
      LCALL ONOFF
      LJMP OUT
P5:   CJNE A,#069H,P6
      LCALL UP

```

เอกสารนี้เป็นเอกสารที่เขียนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;-----DELAY-----
DELAY:      MOV R0,#0FFH
D1:         MOV R1,#0FH
D2:         MOV R2,#0FH
D3:         DJNZ R2,D3
            DJNZ R1,D2
            DJNZ R0,D1
            RET

END

;-----

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### AlertDialog.java

```

package controlrobot;
import java.awt.*;
import java.awt.event.*;

public class AlertDialog extends Dialog implements ActionListener {

    public AlertDialog(Frame parent,
                       String title,
                       String lineOne,
                       String lineTwo,
                       String lineThree) {
        super(parent, title, true);

        Panel labelPanel = new Panel();
        labelPanel.setLayout(new GridLayout(3, 1));
        labelPanel.add(new Label(lineOne, Label.CENTER));
        labelPanel.add(new Label(lineTwo, Label.CENTER));
        labelPanel.add(new Label(lineThree, Label.CENTER));
        add(labelPanel, "Center");

        Panel buttonPanel = new Panel();
        Button okButton = new Button("OK");
        okButton.addActionListener(this);
        buttonPanel.add(okButton);
        add(buttonPanel, "South");

        FontMetrics fm = getFontMetrics(getFont());
        int width = Math.max(fm.stringWidth(lineOne),
                             Math.max(fm.stringWidth(lineTwo),
                                       fm.stringWidth(lineThree)));

        setSize(width + 40, 150);
        setLocation(parent.getLocationOnScreen().x + 30,
                   parent.getLocationOnScreen().y + 30);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent e) {
        setVisible(false);
        dispose();
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Main.java**

```

package controlrobot;

import javax.comm.*;
import java.awt.*;
import java.awt.event.*;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.FileNotFoundException;
import java.util.Properties;
import java.util.Enumeration;

public class Main extends Frame implements ActionListener {

    final int HEIGHT = 450;
    final int WIDTH = 410;

    private MenuBar mb;
    private Menu fileMenu;
    private MenuItem openItem;
    private MenuItem saveItem;
    private MenuItem exitItem;

    private Button openButton;
    private Button closeButton;
    private Button breakButton;
    private Panel buttonPanel;

    private Panel messagePanel;
    private TextArea messageAreaOut;
    private TextArea messageAreaIn;

    private ConfigurationPanel configurationPanel;
    private SerialParameters parameters;
    private SerialConnection connection;

    private Properties props = null;

    public static void main(String[] args) {
        if ((args.length > 0)
            && (args[0].equals("-h")
                || args[0].equals("-help"))) {
            System.out.println("usage: java Main [configuration
File]");
            System.exit(1);
        }

        Main serialDemo = new Main(args);
        serialDemo.setVisible(true);
        serialDemo.repaint();
    }

    public Main(String[] args){
        super("Robot Controller");

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

parameters = new SerialParameters();

// Set up the GUI for the program
addWindowListener(new CloseHandler(this));

mb = new MenuBar();

fileMenu = new Menu("File");

openItem = new MenuItem("Load");
openItem.addActionListener(this);
fileMenu.add(openItem);

saveItem = new MenuItem("Save");
saveItem.addActionListener(this);
fileMenu.add(saveItem);

exitItem = new MenuItem("Exit");
exitItem.addActionListener(this);
fileMenu.add(exitItem);

mb.add(fileMenu);

setMenuBar(mb);

messagePanel = new Panel();
messagePanel.setLayout(new GridLayout(2, 1));

messageAreaOut = new TextArea();
messagePanel.add(messageAreaOut);

messageAreaIn = new TextArea();
messageAreaIn.setEditable(false);
messagePanel.add(messageAreaIn);

add(messagePanel, "Center");

configurationPanel = new ConfigurationPanel(this);

buttonPanel = new Panel();

openButton = new Button("Open Port");
openButton.addActionListener(this);
buttonPanel.add(openButton);

closeButton = new Button("Close Port");
closeButton.addActionListener(this);
closeButton.setEnabled(false);
buttonPanel.add(closeButton);

breakButton = new Button("Send Break");
breakButton.addActionListener(this);
breakButton.setEnabled(false);
buttonPanel.add(breakButton);

Panel southPanel = new Panel();

GridBagLayout gridBag = new GridBagLayout();
GridBagConstraints cons = new GridBagConstraints();

```

```

southPanel.setLayout(gridBag);

cons.gridwidth = GridBagConstraints.REMAINDER;
gridBag.setConstraints(configurationPanel, cons);
cons.weightx = 1.0;
southPanel.add(configurationPanel);
gridBag.setConstraints(buttonPanel, cons);
southPanel.add(buttonPanel);

add(southPanel, "South");

parseArgs(args);

connection = new SerialConnection(this, parameters,
                                  messageAreaOut, messageAreaIn);
setConfigurationPanel();

Dimension screenSize =
Toolkit.getDefaultToolkit().getScreenSize();

setLocation(screenSize.width/2 - WIDTH/2,
            screenSize.height/2 - HEIGHT/2);

setSize(WIDTH, HEIGHT);
}

public void setConfigurationPanel() {
configurationPanel.setConfigurationPanel();
}

public void actionPerformed(ActionEvent e) {
String cmd = e.getActionCommand();

// Loads a configuration file.
if (cmd.equals("Load")) {
if (connection.isOpen()) {
AlertDialog ad = new AlertDialog(this, "Port Open!",
                                "Configuration may not",
                                "be loaded",
                                "while a port is open.");
} else {
FileDialog fd = new FileDialog(this,
                                "Load Port Configuration",
                                FileDialog.LOAD);

fd.setVisible(true);
String file = fd.getFile();
if (file != null) {
String dir = fd.getDirectory();
File f = new File(dir + file);
try {
FileInputStream fis = new FileInputStream(f);
props = new Properties();
props.load(fis);
fis.close();
} catch (FileNotFoundException e1) {
System.err.println(e1);
} catch (IOException e2) {
System.err.println(e2);
}

loadParams();
}
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
}
// Saves a configuration file.
if (cmd.equals("Save")) {
    configurationPanel.setParameters();
    FileDialog fd = new FileDialog(this, "Save Port
Configuration",
                                FileDialog.SAVE);
    fd.setFile("serialedemo.properties");
    fd.setVisible(true);
    String fileName = fd.getFile();
    String directory = fd.getDirectory();
    if ((fileName != null) && (directory != null)) {
        writeFile(directory + fileName);
    }
}

// Calls shutdown, which exits the program.
if (cmd.equals("Exit")) {
    shutdown();
}

// Opens a port.
if (cmd.equals("Open Port")) {
    openButton.setEnabled(false);
    Cursor previousCursor = getCursor();
setNewCursor (Cursor.getPredefinedCursor(Cursor.WAIT_CURSOR));
    configurationPanel.setParameters();
    try {
        connection.openConnection();
    } catch (SerialConnectionException e2) {
        AlertDialog ad = new AlertDialog(this,
            "Error Opening Port!",
            "Error opening port,",
            e2.getMessage() + ".",
            "Select new settings, try again.");
        openButton.setEnabled(true);
        setNewCursor(previousCursor);
        return;
    }
    portOpened();
    setNewCursor(previousCursor);
}

// Closes a port.
if (cmd.equals("Close Port")) {
    portClosed();
}

// Sends a break signal to the port.
if (cmd.equals("Send Break")) {
    connection.sendBreak();
}
}

public void portOpened() {
    openButton.setEnabled(false);
    closeButton.setEnabled(true);
    breakButton.setEnabled(true);
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการเรียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

public void portClosed() {
    connection.closeConnection();
    openButton.setEnabled(true);
    closeButton.setEnabled(false);
    breakButton.setEnabled(false);
}

private void setNewCursor(Cursor c) {
    setCursor(c);
    messageAreaIn.setCursor(c);
    messageAreaOut.setCursor(c);
}

private void writeFile(String path) {
    Properties newProps;
    FileOutputStream fileOut = null;

    newProps = new Properties();

    newProps.put("portName", parameters.getPortName());
    newProps.put("baudRate", parameters.getBaudRateString());
    newProps.put("flowControlIn",
parameters.getFlowControlInString());
    newProps.put("flowControlOut",
parameters.getFlowControlOutString());
    newProps.put("parity", parameters.getParityString());
    newProps.put("databits", parameters.getDatabitsString());
    newProps.put("stopbits", parameters.getStopbitsString());

    try {
        fileOut = new FileOutputStream(path);
    } catch (IOException e) {
        System.out.println("Could not open file for writing");
    }

    newProps.save(fileOut, "Serial Demo poperties");

    try {
        fileOut.close();
    } catch (IOException e) {
        System.out.println("Could not close file for writing");
    }
}

private void shutdown() {
    connection.closeConnection();
    System.exit(1);
}

private void parseArgs(String[] args) {
    if (args.length < 1) {
        return;
    }

    File f = new File(args[0]);

    if (!f.exists()) {

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        f = new File(System.getProperty("user.dir")
                    + System.getProperty("path.separator")
                    + args[0]);
    }

    if (f.exists()) {
        try {
            FileInputStream fis = new FileInputStream(f);
            props = new Properties();
            props.load(fis);
            fis.close();
            loadParams();
        } catch (IOException e) {
        }
    }
}

private void loadParams() {
    parameters.setPortName(props.getProperty("portName"));
    parameters.setBaudRate(props.getProperty("baudRate"));
    parameters.setFlowControlIn(props.getProperty("flowControlIn"));
;
    parameters.setFlowControlOut(props.getProperty("flowControlOut"
));
    parameters.setParity(props.getProperty("parity"));
    parameters.setDataBits(props.getProperty("dataBits"));
    parameters.setStopBits(props.getProperty("stopBits"));
    setConfigurationPanel();
}

class ConfigurationPanel extends Panel implements ItemListener {
    private Frame parent;

    private Label portNameLabel;
    private Choice portChoice;

    private Label baudLabel;
    private Choice baudChoice;

    private Label flowControlInLabel;
    private Choice flowChoiceIn;

    private Label flowControlOutLabel;
    private Choice flowChoiceOut;

    private Label dataBitsLabel;
    private Choice dataBitsChoice;

    private Label stopBitsLabel;
    private Choice stopBitsChoice;

    private Label parityLabel;
    private Choice parityChoice;

    public ConfigurationPanel(Frame parent) {
        this.parent = parent;

        setLayout(new GridLayout(4, 4));

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

portNameLabel = new Label("Port Name:", Label.LEFT);
add(portNameLabel);

portChoice = new Choice();
portChoice.addItemListener(this);
add(portChoice);
listPortChoices();
portChoice.select(parameters.getPortName());

baudLabel = new Label("Baud Rate:", Label.LEFT);
add(baudLabel);

baudChoice = new Choice();
baudChoice.addItem("300");
baudChoice.addItem("2400");
    baudChoice.addItem("4800");
baudChoice.addItem("9600");
baudChoice.addItem("14400");
baudChoice.addItem("28800");
baudChoice.addItem("38400");
baudChoice.addItem("57600");
baudChoice.addItem("152000");

baudChoice.select(Integer.toString(parameters.getBaudRate()));
baudChoice.addItemListener(this);
add(baudChoice);

flowControlInLabel = new Label("Flow Control In:",
Label.LEFT);
add(flowControlInLabel);

flowChoiceIn = new Choice();
flowChoiceIn.addItem("None");
flowChoiceIn.addItem("Xon/Xoff In");
flowChoiceIn.addItem("RTS/CTS In");
flowChoiceIn.select(parameters.getFlowControlInString());
flowChoiceIn.addItemListener(this);
add(flowChoiceIn);

flowControlOutLabel = new Label("Flow Control Out:",
Label.LEFT);
add(flowControlOutLabel);

flowChoiceOut = new Choice();
flowChoiceOut.addItem("None");
flowChoiceOut.addItem("Xon/Xoff Out");
flowChoiceOut.addItem("RTS/CTS Out");
flowChoiceOut.select(parameters.getFlowControlOutString());
flowChoiceOut.addItemListener(this);
add(flowChoiceOut);

databitsLabel = new Label("Data Bits:", Label.LEFT);
add(databitsLabel);

databitsChoice = new Choice();
databitsChoice.addItem("5");
databitsChoice.addItem("6");
databitsChoice.addItem("7");
databitsChoice.addItem("8");
databitsChoice.select(parameters.getDatabitsString());

```

เอกสารนี้เป็นเอกสารทบทวนวิชาสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่หรือใช้  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

databitsChoice.addItemListener(this);
add(databitsChoice);

stopbitsLabel = new Label("Stop Bits:", Label.LEFT);
add(stopbitsLabel);

stopbitsChoice = new Choice();
stopbitsChoice.addItem("1");
stopbitsChoice.addItem("1.5");
stopbitsChoice.addItem("2");
stopbitsChoice.select(parameters.getStopbitsString());
stopbitsChoice.addItemListener(this);
add(stopbitsChoice);

parityLabel = new Label("Parity:", Label.LEFT);
add(parityLabel);

parityChoice = new Choice();
parityChoice.addItem("None");
parityChoice.addItem("Even");
parityChoice.addItem("Odd");
parityChoice.select("None");
parityChoice.select(parameters.getParityString());
parityChoice.addItemListener(this);
add(parityChoice);
}

public void setConfigurationPanel() {
portChoice.select(parameters.getPortName());
baudChoice.select(parameters.getBaudRateString());
flowChoiceIn.select(parameters.getFlowControlInString());
flowChoiceOut.select(parameters.getFlowControlOutString());
databitsChoice.select(parameters.getDatabitsString());
stopbitsChoice.select(parameters.getStopbitsString());
parityChoice.select(parameters.getParityString());
}

public void setParameters() {
parameters.setPortName(portChoice.getSelectedItem());
parameters.setBaudRate(baudChoice.getSelectedItem());

parameters.setFlowControlIn(flowChoiceIn.getSelectedItem());

parameters.setFlowControlOut(flowChoiceOut.getSelectedItem());
parameters.setDatabits(databitsChoice.getSelectedItem());
parameters.setStopbits(stopbitsChoice.getSelectedItem());
parameters.setParity(parityChoice.getSelectedItem());
}

void listPortChoices() {
CommPortIdentifier portId;

Enumeration en = CommPortIdentifier.getPortIdentifiers();

// iterate through the ports.
while (en.hasMoreElements()) {
portId = (CommPortIdentifier) en.nextElement();
if (portId.getPortType() ==
CommPortIdentifier.PORT_SERIAL) {
portChoice.addItem(portId.getName());
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    portChoice.select(parameters.getPortName());
}

public void itemStateChanged(ItemEvent e) {
    // Check if port is open.
    if (connection.isOpen()) {
        // If port is open do not allow port to change.
        if (e.getItemSelectable() == portChoice) {
            // Alert user.
            AlertDialog ad = new AlertDialog(parent, "Port
Open!",
                                        "Port can not",
                                        "be changed",
                                        "while a port is open.");

            // Return configurationPanel to pre-choice settings.
            setConfigurationPanel();
            return;
        }
        // Set the parameters from the choice panel.
        setParameters();
        try {
            // Attempt to change the settings on an open port.
            connection.setConnectionParameters();
        } catch (SerialConnectionException ex) {
            // If setting can not be changed, alert user, return
to // pre-choice settings.
            AlertDialog ad = new AlertDialog(parent,
            "Unsupported Configuration!",
            "Configuration Parameter unsupported,",
            "select new value.",
            "Returning to previous configuration.");
            setConfigurationPanel();
        }
        } else {
            // Since port is not open just set the parameter object.
            setParameters();
        }
    }
}

class CloseHandler extends WindowAdapter {
    Main sd;

    public CloseHandler(Main sd) {
        this.sd = sd;
    }

    public void windowClosing(WindowEvent e) {
        sd.shutdown();
    }
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### PortRequestedDialog.java

```

package controlrobot;

import java.awt.*;
import java.awt.event.*;
import javax.comm.*;

public class PortRequestedDialog extends Dialog implements
ActionListener {

    private Main parent;

    public PortRequestedDialog(Main parent) {
        super(parent, "Port Requested!", true);
        this.parent = parent;

        String lineOne = "Your port has been requested";
        String lineTwo = "by an other application.";
        String lineThree = "Do you want to give up your port?";
        Panel labelPanel = new Panel();
        labelPanel.setLayout(new GridLayout(3, 1));
        labelPanel.add(new Label(lineOne, Label.CENTER));
        labelPanel.add(new Label(lineTwo, Label.CENTER));
        labelPanel.add(new Label(lineThree, Label.CENTER));
        add(labelPanel, "Center");

        Panel buttonPanel = new Panel();
        Button yesButton = new Button("Yes");
        yesButton.addActionListener(this);
        buttonPanel.add(yesButton);
        Button noButton = new Button("No");
        noButton.addActionListener(this);
        buttonPanel.add(noButton);
        add(buttonPanel, "South");

        FontMetrics fm = getFontMetrics(getFont());
        int width = Math.max(fm.stringWidth(lineOne),
            Math.max(fm.stringWidth(lineTwo),
                fm.stringWidth(lineThree)));

        setSize(width + 40, 150);
        setLocation(parent.getLocationOnScreen().x + 30,
            parent.getLocationOnScreen().y + 30);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent e) {
        String cmd = e.getActionCommand();

        if (cmd.equals("Yes")) {
            parent.portClosed();
        }

        setVisible(false);
        dispose();
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### SerialConnection.java

```

package controlrobot;

import javax.comm.*;
import java.io.*;
import java.awt.TextArea;
import java.awt.event.*;
import java.util.TooManyListenersException;

public class SerialConnection implements SerialPortEventListener,
    CommPortOwnershipListener {
    private Main parent;

    private TextArea messageAreaOut;
    private TextArea messageAreaIn;
    private SerialParameters parameters;
    private OutputStream os;
    private InputStream is;
    private KeyHandler keyHandler;

    private CommPortIdentifier portId;
    private SerialPort sPort;

    private boolean open;

    public SerialConnection(Main parent,
        SerialParameters parameters,
        TextArea messageAreaOut,
        TextArea messageAreaIn) {
        this.parent = parent;
        this.parameters = parameters;
        this.messageAreaOut = messageAreaOut;
        this.messageAreaIn = messageAreaIn;
        open = false;
    }

    public void openConnection() throws SerialConnectionException {
        // Obtain a CommPortIdentifier object for the port you want to
        open.
        try {
            portId =
CommPortIdentifier.getPortIdentifier(parameters.getPortName());
        } catch (NoSuchPortException e) {
            throw new SerialConnectionException(e.getMessage());
        }

        // Open the port represented by the CommPortIdentifier object.
        Give
        // the open call a relatively long timeout of 30 seconds to
        allow
        // a different application to relinquish the port if the user
        // wants to.
        try {
            sPort = (SerialPort)portId.open("Main", 30000);
        } catch (PortInUseException e) {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        throw new SerialConnectionException(e.getMessage());
    }

    // Set the parameters of the connection. If they won't set,
close the
    // port before throwing an exception.
    try {
        setConnectionParameters();
    } catch (SerialConnectionException e) {
        sPort.close();
        throw e;
    }

    // Open the input and output streams for the connection. If
they won't
    // open, close the port before throwing an exception.
    try {
        os = sPort.getOutputStream();
        is = sPort.getInputStream();
    } catch (IOException e) {
        sPort.close();
        throw new SerialConnectionException("Error opening i/o
streams");
    }

    // Create a new KeyHandler to respond to key strokes in the
// messageAreaOut. Add the KeyHandler as a keyListener to the
// messageAreaOut.
    keyHandler = new KeyHandler(os);
    messageAreaOut.addKeyListener(keyHandler);

    // Add this object as an event listener for the serial port.
    try {
        sPort.addEventListener(this);
    } catch (TooManyListenersException e) {
        sPort.close();
        throw new SerialConnectionException("too many listeners
added");
    }

    // Set notifyOnDataAvailable to true to allow event driven
input.
    sPort.notifyOnDataAvailable(true);

    // Set notifyOnBreakInterrupt to allow event driven break
handling.
    sPort.notifyOnBreakInterrupt(true);

    // Set receive timeout to allow breaking out of polling loop
during
    // input handling.
    try {
        sPort.enableReceiveTimeout(30);
    } catch (UnsupportedCommOperationException e) {
    }

    // Add ownership listener to allow ownership event handling.
    portId.addPortOwnershipListener(this);

    open = true;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

public void setConnectionParameters() throws
SerialConnectionException {

    // Save state of parameters before trying a set.
    int oldBaudRate = sPort.getBaudRate();
    int oldDatabits = sPort.getDataBits();
    int oldStopbits = sPort.getStopBits();
    int oldParity = sPort.getParity();
    int oldFlowControl = sPort.getFlowControlMode();

    // Set connection parameters, if set fails return parameters
object
// to original state.
try {
    sPort.setSerialPortParams(parameters.getBaudRate(),
                                parameters.getDatabits(),
                                parameters.getStopbits(),
                                parameters.getParity());
} catch (UnsupportedCommOperationException e) {
    parameters.setBaudRate(oldBaudRate);
    parameters.setDataBits(oldDatabits);
    parameters.setStopbits(oldStopbits);
    parameters.setParity(oldParity);
    throw new SerialConnectionException("Unsupported
parameter");
}

// Set flow control.
try {
    sPort.setFlowControlMode(parameters.getFlowControlIn()
                                | parameters.getFlowControlOut());
} catch (UnsupportedCommOperationException e) {
    throw new SerialConnectionException("Unsupported flow
control");
}
}

public void closeConnection() {
    // If port is already closed just return.
    if (!open) {
        return;
    }

    // Remove the key listener.
    messageAreaOut.removeKeyListener(keyHandler);

    // Check to make sure sPort has reference to avoid a NPE.
    if (sPort != null) {
        try {
            // close the i/o streams.
            os.close();
            is.close();
        } catch (IOException e) {
            System.err.println(e);
        }

        // Close the port.
        sPort.close();

        // Remove the ownership listener.

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้ในเพียงการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        portId.removePortOwnershipListener(this);
    }

    open = false;
}

public void sendBreak() {
    sPort.sendBreak(1000);
}

public boolean isOpen() {
    return open;
}

public void serialEvent(SerialPortEvent e) {
    // Create a StringBuffer and int to receive input data.
    StringBuffer inputBuffer = new StringBuffer();
    int newData = 0;

    // Determine type of event.
    switch (e.getEventType()) {

        // Read data until -1 is returned. If \r is received
        substitute // \n for correct newline handling.
        case SerialPortEvent.DATA_AVAILABLE:
            while (newData != -1) {
                try {
                    newData = is.read();
                    if (newData == -1) {
                        break;
                    }
                    if ('\r' == (char)newData) {
                        inputBuffer.append('\n');
                    } else {
                        inputBuffer.append((char)newData);
                    }
                } catch (IOException ex) {
                    System.err.println(ex);
                    return;
                }
            }

            // Append received data to messageAreaIn.
            messageAreaIn.append(new String(inputBuffer));
            break;

        // If break event append BREAK RECEIVED message.
        case SerialPortEvent.BI:
            messageAreaIn.append("\n--- BREAK RECEIVED ---\n");
    }
}

public void ownershipChange(int type) {
    if (type == CommPortOwnershipListener.PORT_OWNERSHIP_REQUESTED)
    {
        PortRequestedDialog prd = new PortRequestedDialog(parent);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
class KeyHandler extends KeyAdapter {
    OutputStream os;

    public KeyHandler(OutputStream os) {
        super();
        this.os = os;
    }

    public void keyTyped(KeyEvent evt) {
        char newCharacter = evt.getKeyChar();
        try {
            os.write((int)newCharacter);
        } catch (IOException e) {
            System.err.println("OutputStream write error: " + e);
        }
    }
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**SerialConnectionException.java**

```
package controlrobot;

public class SerialConnectionException extends Exception {

    public SerialConnectionException(String str) {
        super(str);
    }

    public SerialConnectionException() {
        super();
    }
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**SerialParameters.java**

```

package controlrobot;

import javax.comm.*;

public class SerialParameters {

    private String portName;
    private int baudRate;
    private int flowControlIn;
    private int flowControlOut;
    private int databits;
    private int stopbits;
    private int parity;

    public SerialParameters () {
        this("",
            9600,
            SerialPort.FLOWCONTROL_NONE,
            SerialPort.FLOWCONTROL_NONE,
            SerialPort.DATABITS_8,
            SerialPort.STOPBITS_1,
            SerialPort.PARITY_NONE );
    }

    public SerialParameters(String portName,
        int baudRate,
        int flowControlIn,
        int flowControlOut,
        int databits,
        int stopbits,
        int parity) {

        this.portName = portName;
        this.baudRate = baudRate;
        this.flowControlIn = flowControlIn;
        this.flowControlOut = flowControlOut;
        this.databits = databits;
        this.stopbits = stopbits;
        this.parity = parity;
    }

    public void setPortName(String portName) {
        this.portName = portName;
    }

    public String getPortName() {
        return portName;
    }

    public void setBaudRate(int baudRate) {
        this.baudRate = baudRate;
    }

    public void setBaudRate(String baudRate) {
        this.baudRate = Integer.parseInt(baudRate);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

public int getBaudRate() {
    return baudRate;
}

public String getBaudRateString() {
    return Integer.toString(baudRate);
}

public void setFlowControlIn(int flowControlIn) {
    this.flowControlIn = flowControlIn;
}

public void setFlowControlIn(String flowControlIn) {
    this.flowControlIn = stringToFlow(flowControlIn);
}

public int getFlowControlIn() {
    return flowControlIn;
}

public String getFlowControlInString() {
    return flowToString(flowControlIn);
}

public void setFlowControlOut(int flowControlOut) {
    this.flowControlOut = flowControlOut;
}

public void setFlowControlOut(String flowControlOut) {
    this.flowControlOut = stringToFlow(flowControlOut);
}

public int getFlowControlOut() {
    return flowControlOut;
}

public String getFlowControlOutString() {
    return flowToString(flowControlOut);
}

public void setDatabits(int databits) {
    this.databits = databits;
}

public void setDatabits(String databits) {
    if (databits.equals("5")) {
        this.databits = SerialPort.DATABITS_5;
    }
    if (databits.equals("6")) {
        this.databits = SerialPort.DATABITS_6;
    }
    if (databits.equals("7")) {
        this.databits = SerialPort.DATABITS_7;
    }
    if (databits.equals("8")) {
        this.databits = SerialPort.DATABITS_8;
    }
}

public int getDatabits() {
    return databits;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

public String getDatabitsString() {
    switch(databits) {
        case SerialPort.DATABITS_5:
            return "5";
        case SerialPort.DATABITS_6:
            return "6";
        case SerialPort.DATABITS_7:
            return "7";
        case SerialPort.DATABITS_8:
            return "8";
        default:
            return "8";
    }
}

public void setStopbits(int stopbits) {
    this.stopbits = stopbits;
}

public void setStopbits(String stopbits) {
    if (stopbits.equals("1")) {
        this.stopbits = SerialPort.STOPBITS_1;
    }
    if (stopbits.equals("1.5")) {
        this.stopbits = SerialPort.STOPBITS_1_5;
    }
    if (stopbits.equals("2")) {
        this.stopbits = SerialPort.STOPBITS_2;
    }
}

public int getStopbits() {
    return stopbits;
}

public String getStopbitsString() {
    switch(stopbits) {
        case SerialPort.STOPBITS_1:
            return "1";
        case SerialPort.STOPBITS_1_5:
            return "1.5";
        case SerialPort.STOPBITS_2:
            return "2";
        default:
            return "1";
    }
}

public void setParity(int parity) {
    this.parity = parity;
}

public void setParity(String parity) {
    if (parity.equals("None")) {
        this.parity = SerialPort.PARITY_NONE;
    }
    if (parity.equals("Even")) {
        this.parity = SerialPort.PARITY_EVEN;
    }
    if (parity.equals("Odd")) {

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        this.parity = SerialPort.PARITY_ODD;
    }
}

public int getParity() {
    return parity;
}

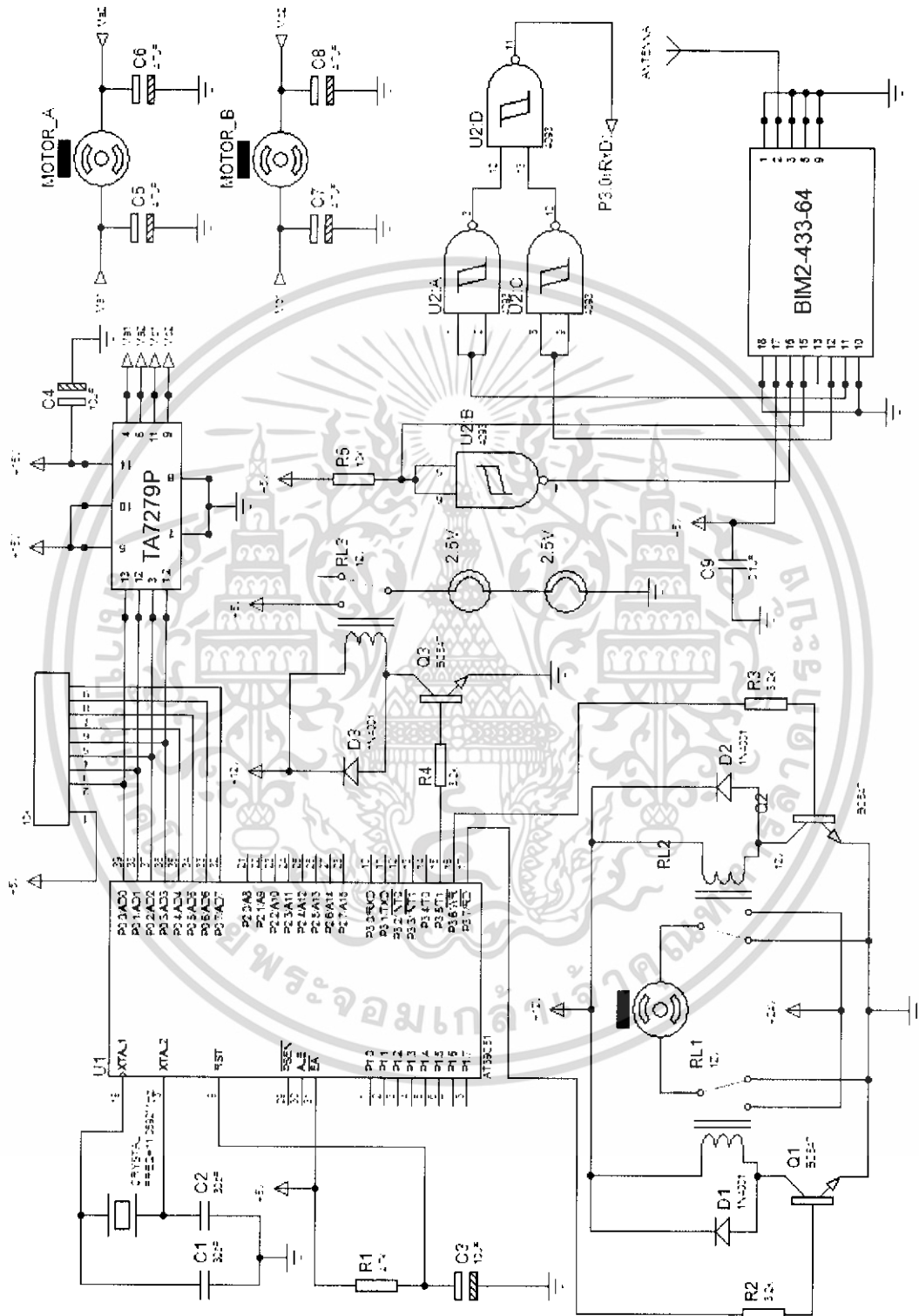
public String getParityString() {
    switch(parity) {
        case SerialPort.PARITY_NONE:
            return "None";
        case SerialPort.PARITY_EVEN:
            return "Even";
        case SerialPort.PARITY_ODD:
            return "Odd";
        default:
            return "None";
    }
}

private int stringToFlow(String flowControl) {
    if (flowControl.equals("None")) {
        return SerialPort.FLOWCONTROL_NONE;
    }
    if (flowControl.equals("Xon/Xoff Out")) {
        return SerialPort.FLOWCONTROL_XONXOFF_OUT;
    }
    if (flowControl.equals("Xon/Xoff In")) {
        return SerialPort.FLOWCONTROL_XONXOFF_IN;
    }
    if (flowControl.equals("RTS/CTS In")) {
        return SerialPort.FLOWCONTROL_RTSCCTS_IN;
    }
    if (flowControl.equals("RTS/CTS Out")) {
        return SerialPort.FLOWCONTROL_RTSCCTS_OUT;
    }
    return SerialPort.FLOWCONTROL_NONE;
}

String flowToString(int flowControl) {
    switch(flowControl) {
        case SerialPort.FLOWCONTROL_NONE:
            return "None";
        case SerialPort.FLOWCONTROL_XONXOFF_OUT:
            return "Xon/Xoff Out";
        case SerialPort.FLOWCONTROL_XONXOFF_IN:
            return "Xon/Xoff In";
        case SerialPort.FLOWCONTROL_RTSCCTS_IN:
            return "RTS/CTS In";
        case SerialPort.FLOWCONTROL_RTSCCTS_OUT:
            return "RTS/CTS Out";
        default:
            return "None";
    }
}
}
}

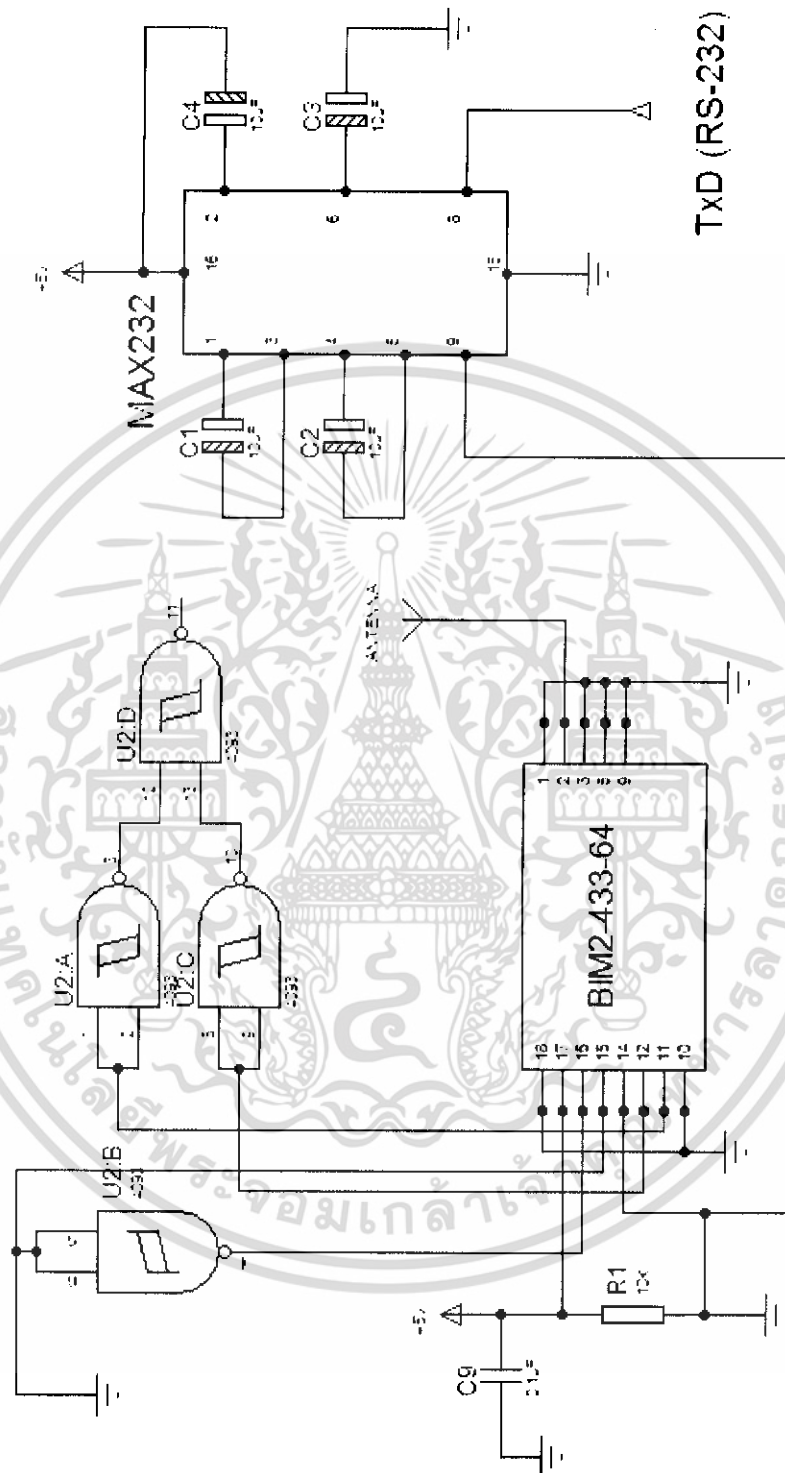
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



### วงจรมอเตอร์ของหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



วงจรในส่วนของสถานีควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TOSHIBA BIPOLAR LINEAR INTEGRATED CIRCUIT SILICON MONOLITHIC

# TA7279P, TA7279AP

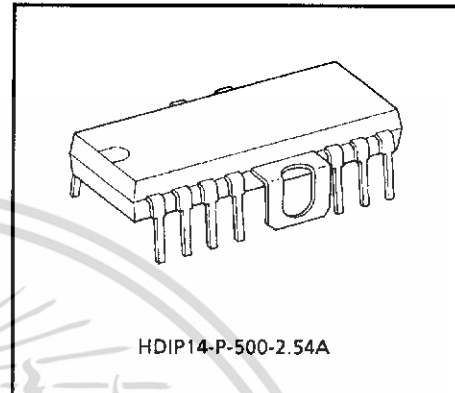
## DUAL BRIDGE DRIVER

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

### FEATURES

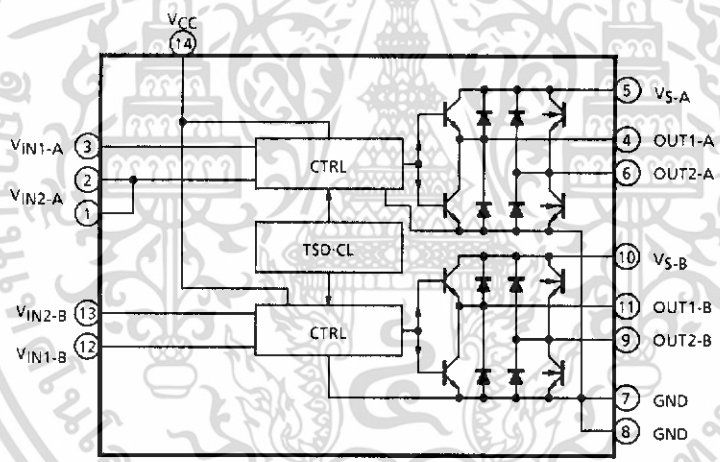
- Wide Range of Operating Voltage  
:  $V_{CC (opr.)} = 6\sim 18\text{ V (P, AP)}$ ,  
 $V_S (opr.) = 0\sim 16\text{ V (P)} / 0\sim 18\text{ V (AP)}$
- Output Current Up to 1.0 A (AVE.), 3.0 A (PEAK)
- Built-in Thermal Shut Down and Current Limiter
- Input Hysteresis for Stable Operation

### BLOCK DIAGRAM



HDIP14-P-500-2.54A

Weight: 3.00 g (Typ.)

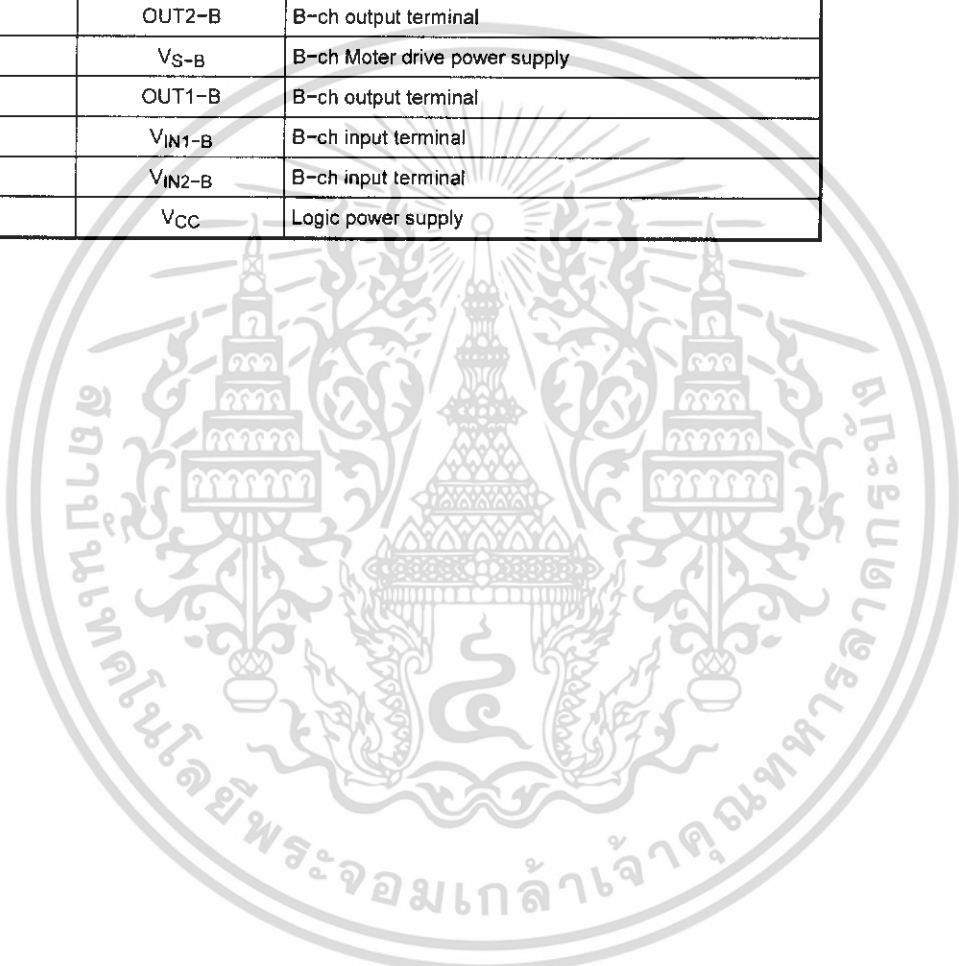


961001EBA1

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

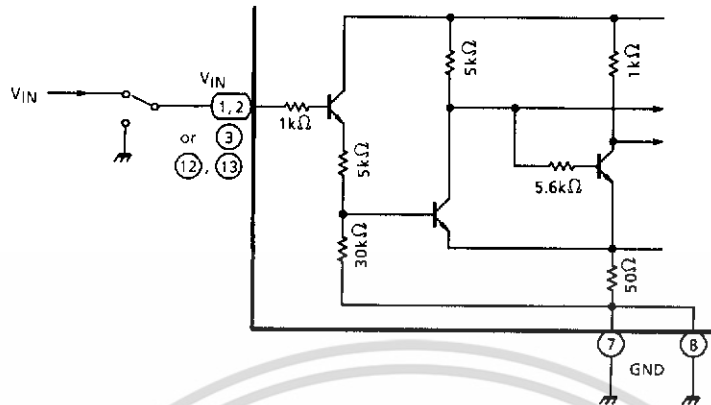
**PIN FUNCTION**

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



**APPLICATION NOTE**

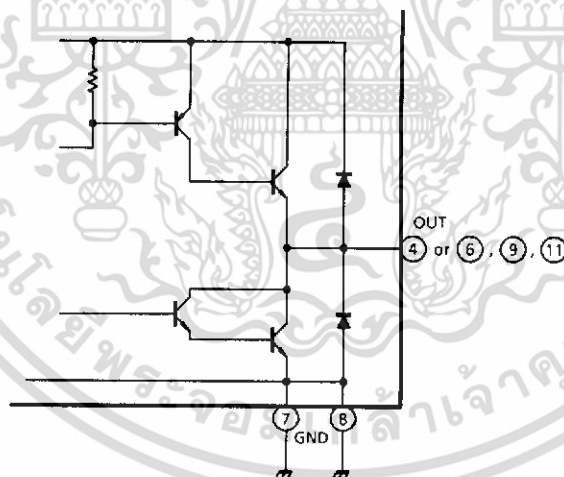
**(1) Input circuit**



Input terminals of (2), (3), (12) and (13) Pin are all high active type and have a hysteresis.  
 3  $\mu$ A Typ. of input current is required.

The input circuit is an active high type, as shown in the diagram. When voltage higher than the specified  $V_{IN(H)}$  is applied, the output is logic "H". When voltage lower than the specified  $V_{IN(L)}$  is applied or if the input is grounded, the output is logic "L". Since the input current  $I_N$  flows to the input when logic "H", be careful with the output impedance at the previous step.

**(2) Output circuit**



## FUNCTION

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

## MAXIMUM RATINGS (Ta = 25°C)

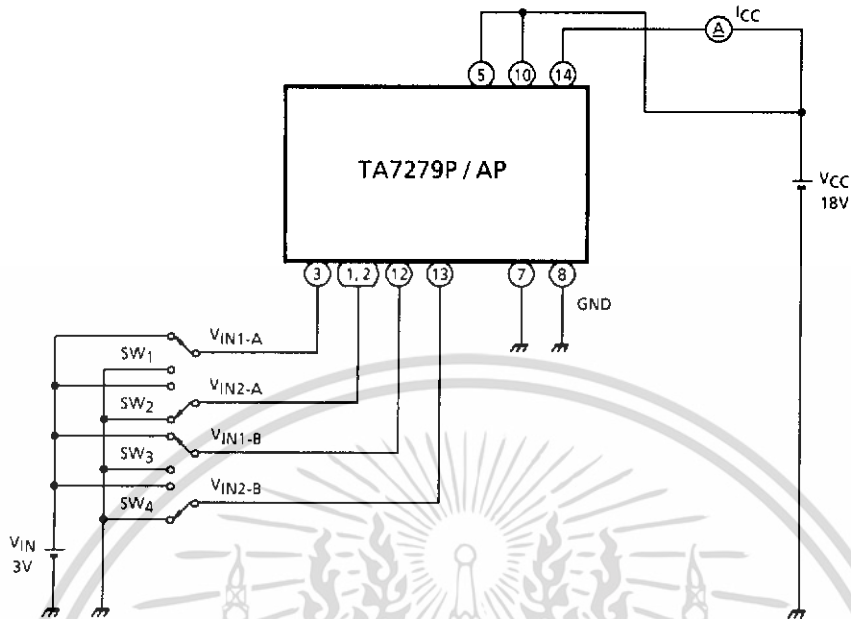
CHARACTERISTIC		SYMBOL	RATING	UNIT
Supply Voltage	AP	V <sub>CC</sub> (MAX.)	25	V
	P		20	
Motor Drive Voltage	AP	V <sub>S</sub> (MAX.)	25	V
	P		18	
Output Current	PEAK	I <sub>O</sub> (PEAK)	3.0	A
	AVE.	I <sub>O</sub> (AVE.)	1.0	
Power Dissipation		P <sub>D</sub> (Note)	2.3	W
Operating Temperature		T <sub>opr</sub>	-30~75	°C
Storage Temperature		T <sub>stg</sub>	-55~150	°C

Note: No heat sink.

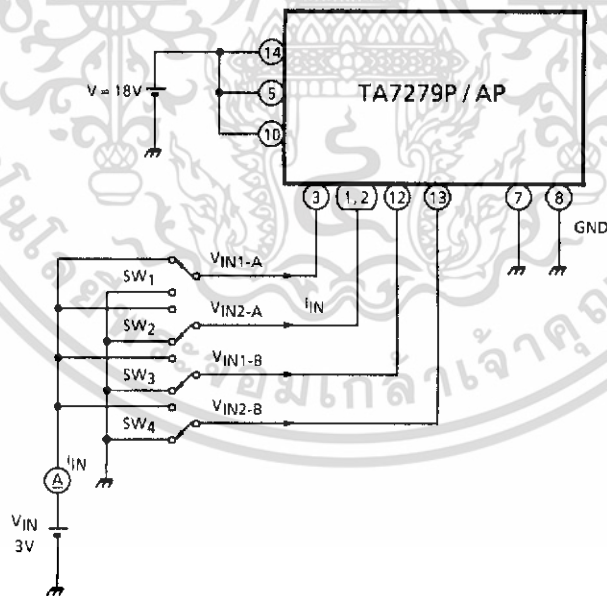
## ELECTRICAL CHARACTERISTICS (Ta = 25°C)

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

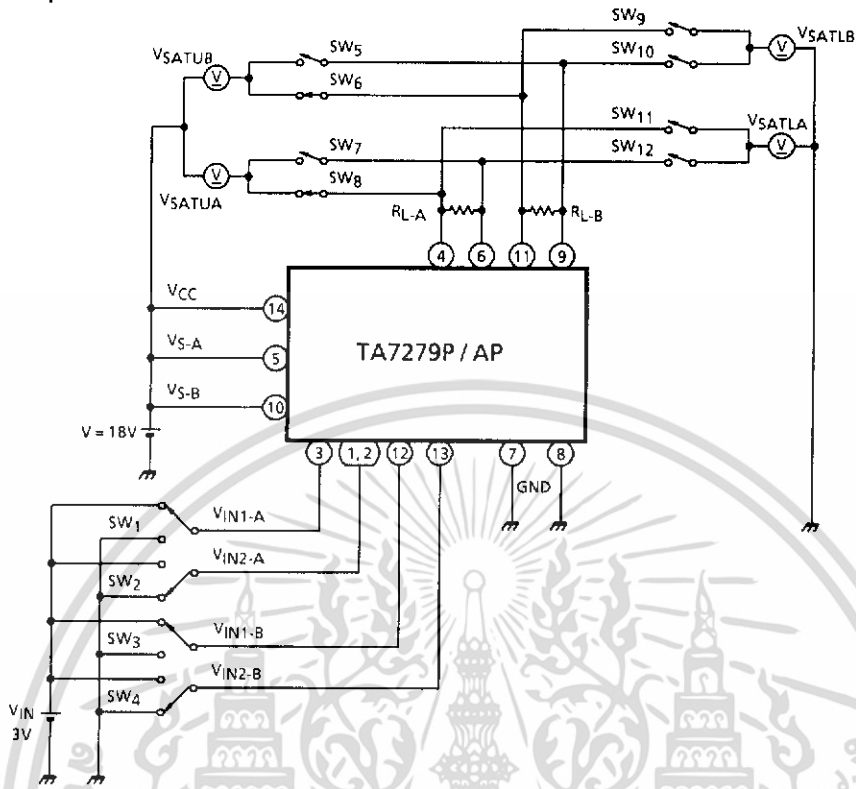
**TEST CIRCUIT 1.**  
I<sub>CC</sub> 1, 2, 3



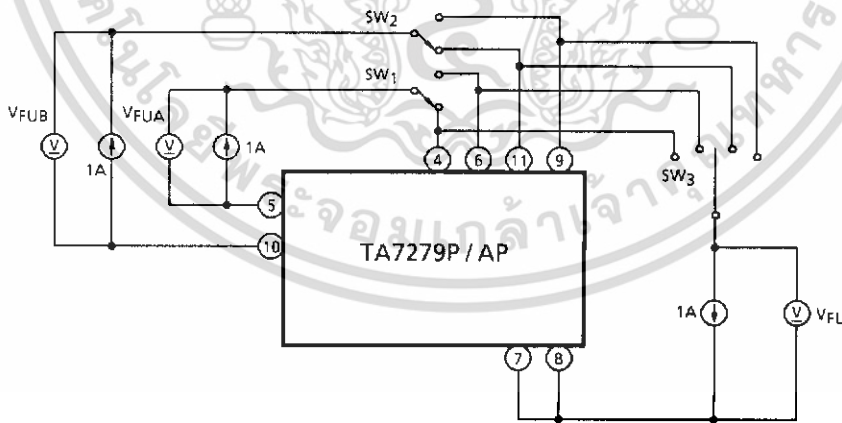
**TEST CIRCUIT 2.**  
I<sub>IN</sub> (H), (L)

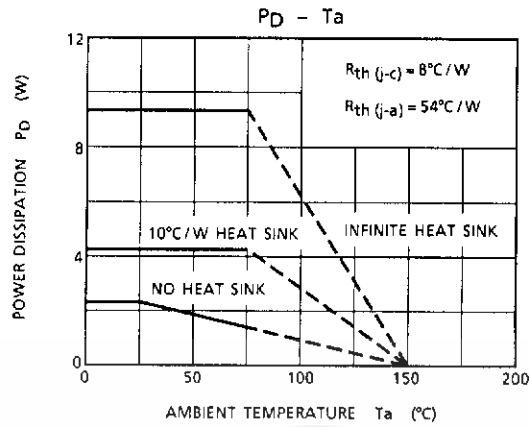


**TEST CIRCUIT 3.**  
 $V_{SATU-1, 2} / V_{SATL-1, 2}$

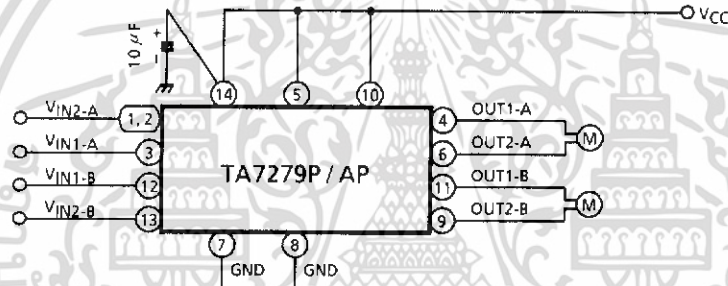


**TEST CIRCUIT 4.**  
 $V_{FU, L}$

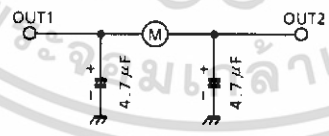




APPLICATION CIRCUIT



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

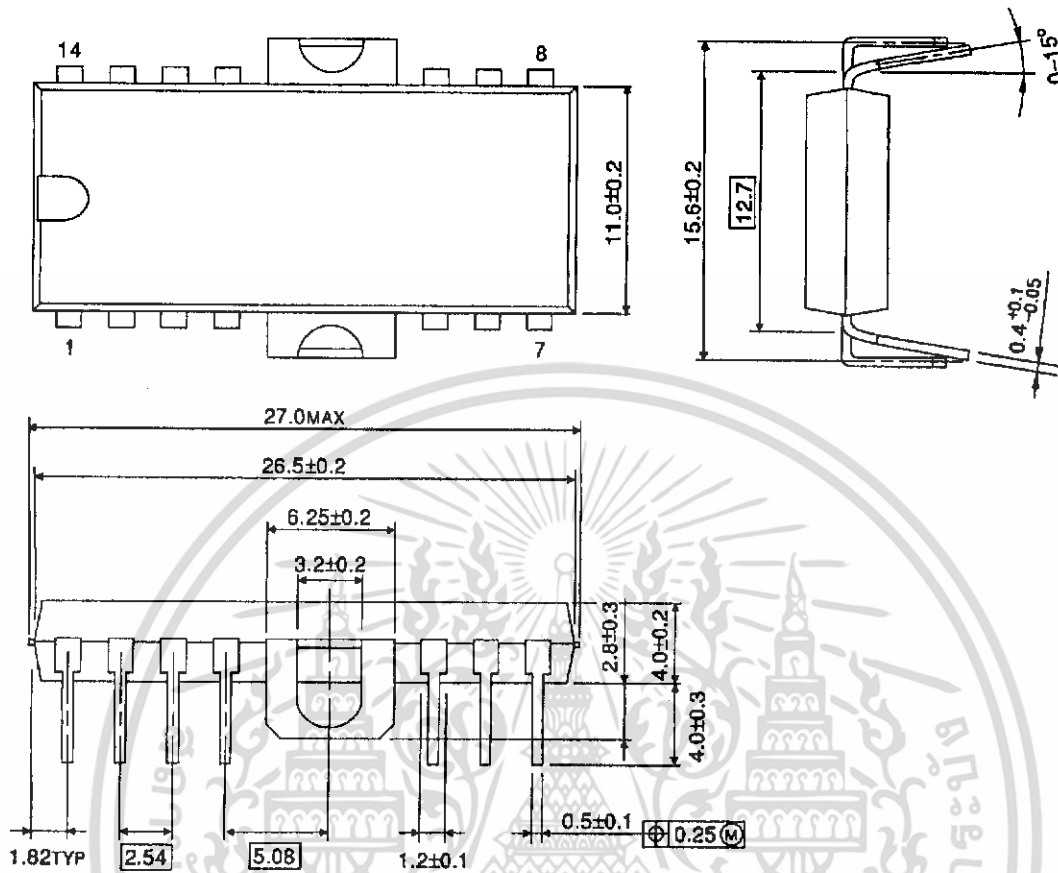


Note: Utmost care is necessary in the design of the output line, V<sub>S</sub> and GND line since IC may be destroyed due to short-circuit between outputs, air contamination fault, or fault by improper grounding.

## OUTLINE DRAWING

HDIP14-P-500-2.54A

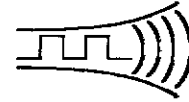
Unit: mm



Weight: 3.00 g (Typ.)

**NEW**

# Radiometrix



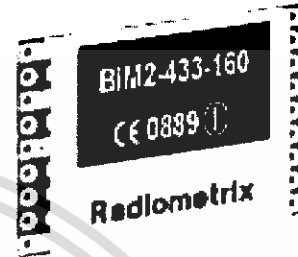
Issue 0144-2-1WU

BiM2-433

12 February 2002

## 433MHz high speed FM radio transceiver module

*The BiM2 transceiver is an enhanced replacement for our original BiM module. It offers greater transmit power, higher data rates, greatly improved receiver interference rejection and a lower profile. The module is ideal for enabling bi-directional wireless connectivity in battery powered or handheld applications*



### Features

### BiM2 Transceiver

- CE Certified by independent Notified Body according to the R&TTE Directive (1999/5/EC)
- Verified to comply with Radio standard ETSI EN 300 220-3 by UKAS accredited Test Laboratory
- Verified to comply with EMC standard ETSI EN 301 489-3 by UKAS accredited Test Laboratory
- Usable range to 200 metres external, 50 metres in building
- Data rates up to 160kbps
- SAW controlled 10mW FM transmitter
- Double conversion FM superhet receiver
- SAW front end filter and full screening
- Plug in replacement for Radiometrix BiM-433-40
- 3V or 5Volt supply at < 20mA

The BiM2 is a half duplex radio transceiver module for use in high-speed bi-directional data transfer applications at ranges up to 200metres. The module operates on the European licence exempt frequency of 433.92MHz. The small footprint of 23 x 33mm and low profile of 4mm together with low power requirements of <20mA @ 3 to 5 Volts enable convenient PCB installation. The high raw data rate capability of 64kbps and fast state change times will support high data throughput of up to 3 kbyte/s in 'streaming' applications or alternatively allows very short air time utilization in multi-node scanning networks.

### Applications

- PDA's, organizers and laptops
- Handheld terminals
- EPOS equipment, barcode scanners, belt clip printers
- Data loggers
- Audience response systems
- In Building environmental monitoring and control
- High end security and fire alarms
- Restaurant ordering systems
- Vehicle data up/download

*Further information will be available imminently, however, BiM-433-F data sheets contains applications information that is equally applicable to the new BiM2.*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

## Functional overview

The transmit section of the BiM2 comprises of a SAW stabilised and FM modulated 433.92MHz oscillator feeding a 10mW buffer/output stage. Operation is controlled by a TX select line, the output achieving full power within 100µs of this line being pulled low. Modulation is applied at the TXD input and may be either a serial digital stream at the same levels as the module's supply rails (digital drive) or a high level analogue waveform with a pk to pk amplitude close to the modules supply level (linear drive). Modulation shaping is performed internally by a 2nd order 44kHz LPF to minimize spectral spreading. The RF output is filtered to meet the requirements of EN 300-220-3 and fed via a fast antenna changeover switch to the 50Ω antenna pin.

The receive section of the BiM2 is a double conversion FM superhet with IF's of 16MHz and 150kHz. The dual gate MOSFET LNA is followed by a 750kHz bandwidth SAW filter to provide >60 dB's rejection of all out of band signals. The receiver is controlled by an active low select line and will power up in <1ms. A post-detection 2nd order 35kHz LPF establishes the signal bandwidth and ensures the clean operation of the subsequent adaptive data slicer. The slicer has a 2ms averaging time constant and is optimised for balanced data, e.g. bi-phase codes. A fast acting carrier detect output will indicate the presence of any RF signals.

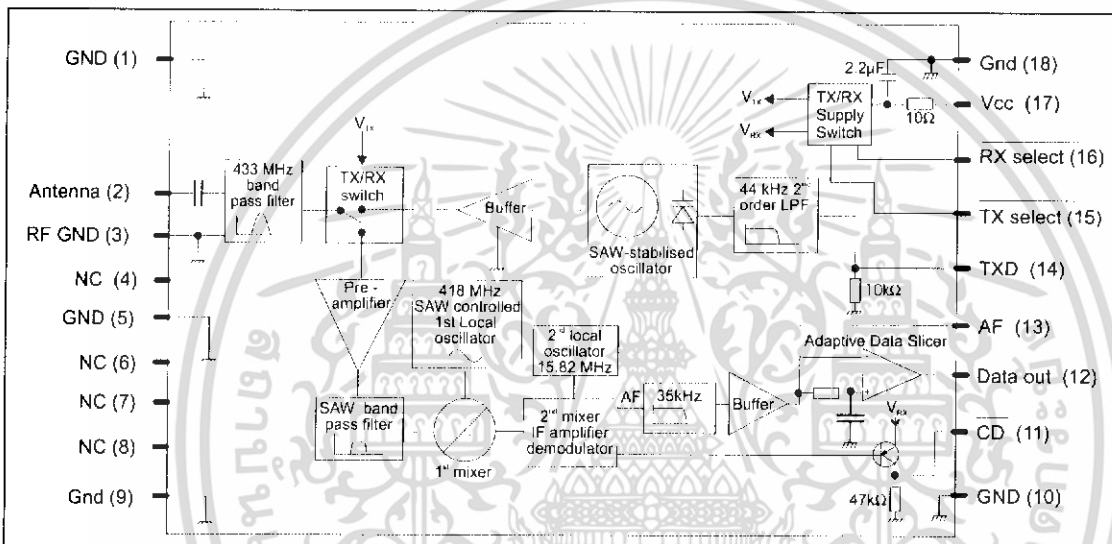


Figure 1: Block diagram

### Pin description:

**RF GND** *pin 1 & 3*

RF ground pin, internally connected to the module screen and pin 5, 9, 10, 18 (0 Volt). This pin should be connected to the RF return path (e.g. coax braid, main PCB ground plane etc.)

**Antenna** *pin 2*

50Ω RF input from the antenna, it is DC isolated internally. (see antenna for suggested antenna/feeds).

**0Volt** *pins 5, 9, 10, 18*

Supply ground connection and screen.

**CD** *pin 11*

Carrier Detect - When the receiver is enabled, a low indicates a signal above the detection threshold is being received. The output is high impedance (50kΩ) and should only be used to drive a CMOS logic input.

**RXD** *pin 12*

This digital output from the internal data slicer is a squared version of the signal on pin 13 (AF). It may be used to drive external decoders. The data is true data, i.e. as fed to the transmitter. Load impedance should be >1kΩ and <1nF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

**AF** *pin 13*

This is a buffered and filtered analogue output from the FM demodulator. It has a standing DC bias of 1.2 volts and 400mV<sub>P-P</sub> base band signal. It is useful as a test point or to drive linear decoders. Load impedance should be >2kΩ and <100pF.

**TXD** *pin 14*

This DC coupled modulation input will accept either serial digital data (0V to Vcc levels) or High level linear signals. Input impedance is 10kΩ.

**TX select** *pin 15*

Active low transmit select. 10kΩ internal pull up to Vcc.

**RX select** *pin 16*

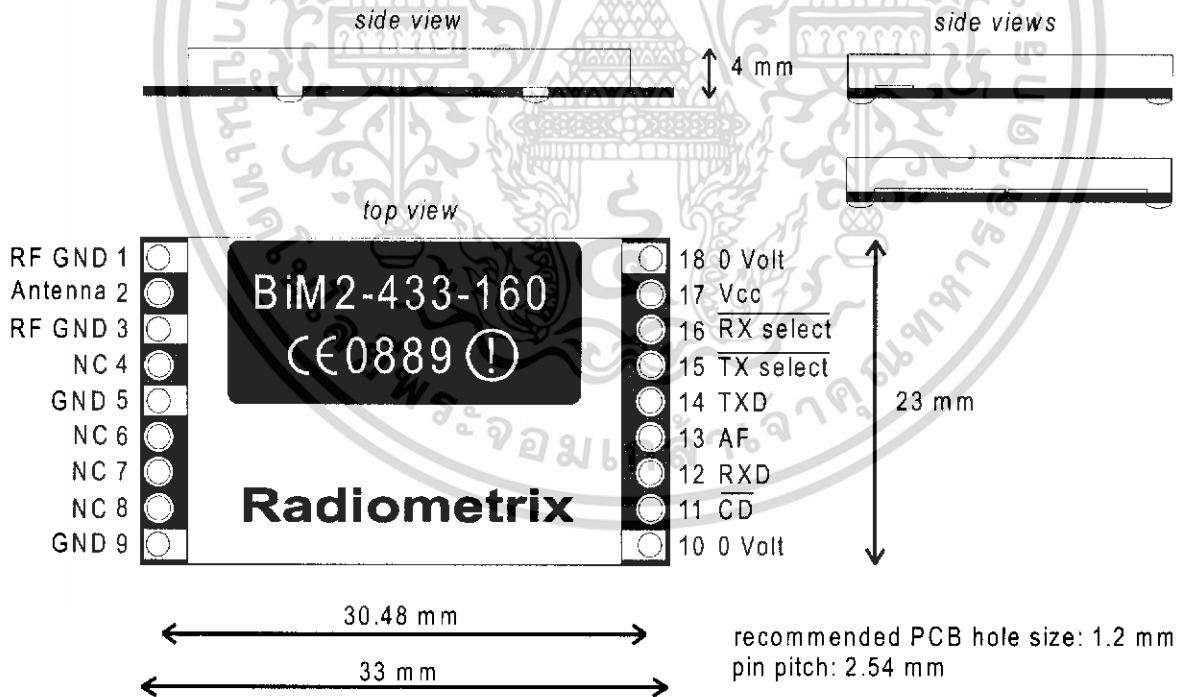
Active low receive select. 10kΩ internal pull up to Vcc.

Pin 15 TX	Pin 16 RX	Function
1	1	power down (<1μA)
1	0	receiver enabled
0	1	transmitter enabled
0	0	self test loop back

**Note:** Loop test allows the receivers to monitor the transmitted signal. The receiver will not receive external signals whilst the TX of the module is enabled.

**Vcc** *pin 17*

+ve supply pin. +3.0 to +5.5 volts @ <20mA . The supply must be clean < 20mV<sub>P-P</sub> ripple. A 2.2μF decoupling capacitor and 10Ω series resistor are used internally to filter the supply.



**Figure 2: Physical dimension**

Weight: 5g (typical with pins)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

## Survival Maximums:

Operating temperature: -10°C to + 55°C  
 Extended operation at -20°C to + 70°C  
 Reduced specification  
 Storage temperature : -40°C to + 100°C

Vcc (pin 17) -0.1V to +10.0 V  
 All other pins -0.1V to +Vcc+0.6V  
 Antenna (pin 2) ±50 V @ <10MHz , +20dBm @ > 10MHz

**Note:** Operation of the BiM2 above 5.5 volt with efficient antenna may result in radiated power levels above the licensed power level.

*Figures apply to 5V versions unless otherwise noted  
 Temperature 20° C unless noted*

Electrical Performance	pin	min.	typ.	max.	units	notes
<b>DC Levels</b>						
supply voltage, Vcc (std. version)	17	4.0	5	5.5	V	
supply voltage, Vcc (3V version)	17	3.0	3.3	4.0	V	
TX supply current, Vcc (std)	17	10	14	16	mA	
TX supply current, Vcc (3.3V)	17	6	8	10	mA	3.3V supply
RX supply current, Vcc (std)	17	12	18	21	mA	
RX supply current, Vcc (3.3V)	17	10	14	17	mA	3.3V supply
supply ripple allowed	17	-	-	20	mV <sub>pk-pk</sub>	below 1MHz
AF output DC level	13	1.0	1.25	1.5	V	
load capacitance on AF / Data	12,13	-	-	100	pF	
CD output load resistance	11	220	-	-	kΩ	
<b>Interface levels</b>						
data output high, 100µA source	12	-	Vcc-0.6	-	V	RXD high
data output low, 100µA sink	12	-	0.4	-	V	RXD low
TX & RX select, high (deselect)	15, 16	Vcc-0.5		Vcc	V	
low (select)	15, 16	0		0.5	V	
Internal select pull-ups	15,16	-	10	-	kΩ	
TXD, high	14	Vcc-0.5		Vcc	V	
low	14	0		0.5	V	

RF Parameters	pin	min.	typ.	max.	units	notes
Antenna pin impedance	2	-	50	-	Ω	TX or RX
RF centre frequency	-	-	433.92	-	MHz	
<b>Transmitter</b>						
RF power output, Vcc std	2	+7	+10	+12	dBm	5V
RF power output, Vcc 3.3V	2	+3	+6	+8	dBm	3.3V
Initial frequency accuracy	-	-50	0	+50	kHz	
Overall frequency accuracy	-	-100	0	+100	kHz	
FM deviation	-	20	30	40	kHz	
Modulation bandwidth	-	DC	-	32	kHz	
Modulation bandwidth	-	DC	-	80	kHz	160kpbs
Modulation distortion	-	-	-	15	%	
<b>Receiver</b>						
RF sensitivity, 10dB S/N	2, 13	-95	-101		dBm	
RF sensitivity, 10dB S/N	2, 13	-91	-96		dBm	3.3V
RF sensitivity, 10dB S/N	2, 13	-94	-94		dBm	160kpbs

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

RF Parameters	pin	min.	typ.	max.	units	notes
RF sensitivity, 1ppm BER	2, 12	-87	-93		dBm	5V
RF sensitivity, 1ppm BER	2, 12	-82	-88		dBm	3.3V
RF sensitivity, 1ppm BER	2, 12		-90		dBm	160kbps
CD threshold	2, 11	-98	-104		dBm	5V
CD threshold	2, 11	-92	-98		dBm	3.3V
CD threshold	2, 11		-96		dBm	160kbps
IF bandwidth	-	-	500	-	kHz	
CD bandwidth	2, 11	-	400	-	kHz	
Ultimate (S+N)/N, -70dBm input	13	-	>40	-	dB	
Ultimate (S+N)/N, -70dBm input	13		30		dB	160kbps
maximum operating RF input	2	-	+10	-	dBm	
AF output level	13	-	400	-	mV	peak to peak
Initial frequency accuracy	-	-50	0	+50	kHz	CD centre

EMC Parameters	pin	min.	typ.	max.	units	notes
<b>Rejections: rejection figures are relative to a 15dB (S+N)/N wanted signal</b>						
Co-channel rejection	2	-	-10	-	dB	
Image rejection ( $f_{RF}-2f_{IF}$ )	2	-	64	-	dB	402.0MHz
Out of band rejection	2	-	>70	-	dB	DC to 2GHz
AM rejection	2	-	>30	-	dB	
Out of band blocking level	2	-	>-15	-	dBm	
Out of band IP <sub>3</sub>	2	-	+1	-	dBm	
<b>Radiations</b>						
RX LO leakage, conducted	2	-	-60	-57	dBm	
RX LO leakage, radiated	-	-	-70	-	dBm	
TX 2 <sup>nd</sup> harmonic	2	-	-42	-36	dBm	
TX harmonics >1GHz	2	-	-40	-30	dBm	
TX spectral bandwidth @-40dBc	2	-	-	250	kHz	worst case

Baseband Transfer Performance	pin	min.	typ.	max.	units	notes
<b>TX → RX</b>						
Linear baseband BW @-3dB	13	0.08	-	34	kHz	TXD to AF
Linear baseband BW @-3dB	13	0.08	-	80	kHz	TXD to AF
Balance code bit rate	12	-	64	-	kbps	
Time between code transitions	14	15.6	-	1000	μs	
Time between code transitions	14	15.6	-	120	μs	S version
Time between code transitions	14	6.25	-	100	μs	160kbps
Averaged code mark:space	14	30	50	70	%	in any 2ms
preamble duration	14	3	-	-	ms	01010101 pattern
preamble duration	14	1	-	-	ms	S version
link delay	14, 12	-	15	-	μs	TXD to RXD
<b>Dynamic Timing</b>						
<i>Power up with signal present</i>						
Power up to valid CD, t <sub>PU-CD</sub>	11	-	0.7	1	ms	
Power up to stable AF, t <sub>PU-AF</sub>	13	-	0.5	1	ms	
Power up to stable data, t <sub>PU-data</sub>	12	-	3	5	ms	
Power up to stable data, t <sub>PU-data</sub>	12	-	-	1	ms	1, S version
Power up to stable data, t <sub>PU-data</sub>	12	-	-	0.8	ms	160kbps

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

Baseband Transfer Performance	pin	min.	typ.	max.	units	notes
<i>Signal applied with supply on</i>						
Signal to valid CD, $t_{sig-CD}$	11	-	0.25	0.5	ms	
Signal to stable data, $t_{sig-data}$	12	-	3	4	ms	
Signal to stable data, $t_{sig-data}$	12	-	-	1	ms	1, S version
Signal to stable data, $t_{sig-data}$	12	-	-	0.5	ms	160kbps
TX power up to full RF	2	-	100	-	$\mu$ s	

Note 1: from 45% to 55% duty cycle

## Antenna requirements

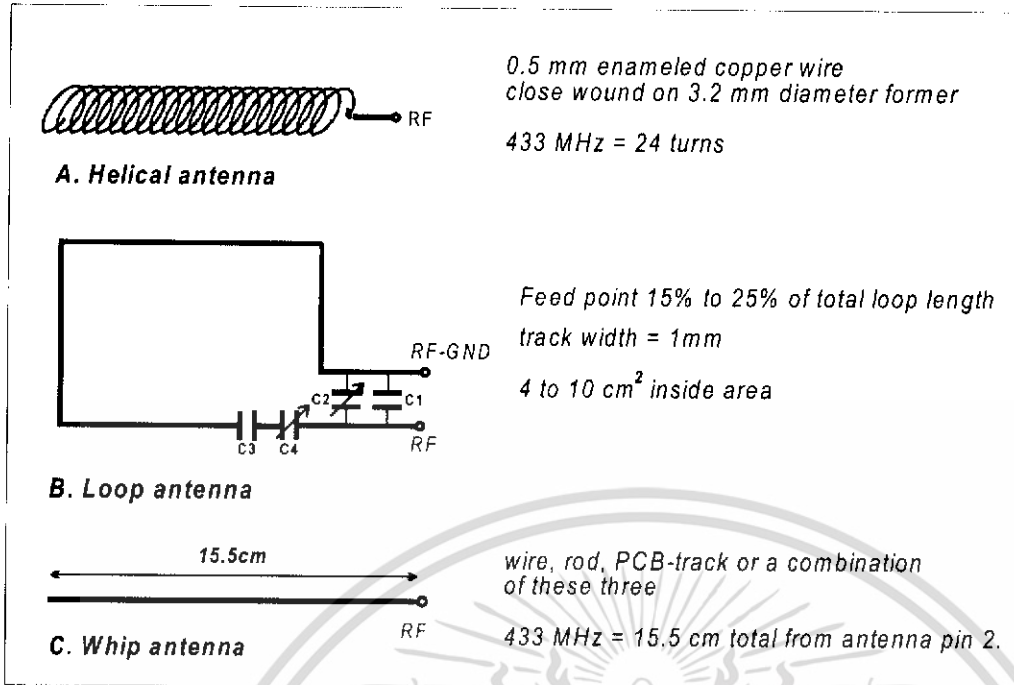
Three types of integral antenna are recommended and approved for use with the module:

- A) **Helical** Wire coil, connected directly to pin 2, open circuit at other end. This antenna is very efficient given it's small size (20mm x 4mm dia.). The helical is a high Q antenna, trim the wire length or expand the coil for optimum results. The helical de-tunes badly with proximity to other conductive objects.
- B) **Loop** A loop of PCB track tuned by a fixed or variable capacitor to ground at the 'hot' end and fed from pin 2 at a point 20% from the ground end. Loops have high immunity to proximity de-tuning.
- C) **Whip** This is a wire, rod, PCB track or combination connected directly to pin 2 of the module. Optimum total length is 16cm (1/4 wave @ 433MHz). Keep the open circuit (hot) end well away from metal components to prevent serious de-tuning. Whips are ground plane sensitive and will benefit from internal 1/4 wave earthed radial(s) if the product is small and plastic cased

	A <i>helical</i>	B <i>loop</i>	C <i>whip</i>
Ultimate performance	**	*	***
Easy of design set-up	**	*	***
Size	***	**	*
Immunity proximity effects	**	***	*
Range open ground to similar antenna			200m

The antenna choice and position directly controls the system range. Keep it clear of other metal in the system, particularly the 'hot' end. The best position by far, is sticking out the top of the product. This is often not desirable for practical/ergonomic reasons thus a compromise may need to be reached. If an internal antenna must be used, try to keep it away from other metal components, particularly large ones like transformers, batteries and PCB tracks/earth plane. The space around the antenna is as important as the antenna itself.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า



**Ordering information**

The standard BiM2, order code BiM2-433-64 is supplied with pins fitted for operation on 5 volt supplies (4 to 5.5v)

**Versions and Variants**

**BiM2-433-64-3V**

A 3 volt version is available, BiM-433-64-3V for operation at 3.3volts (3.0 to 4.0 V) it is identical to the standard version but has been tested and aligned for operation at 3.3v.

**BiM2-433-64-S**

This is intended for RPC or Manchester code only and has fast settling time (maximum 1ms)

Additionally, for volume orders, Radiometrix can supply the BiM2 to the customers' PCB pin requirements or even without any pins.



Radio Frequency Investigation Ltd  
Basingstoke, UK

Radio Frequency Investigation Ltd  
100, The Station, Basingstoke, Hampshire, UK

Telephone: +44 (0)1256 333333  
www.rfi.co.uk

# CERTIFICATE OF CONFORMANCE FROM RADIO FREQUENCY INVESTIGATION LTD.

Test Of: Radcometrix Ltd  
BiM2 433 Transceiver

To: EN 300 220-1 V1.3.1 (2000-09)  
according to the requirements of EN 300 220-4 V1.1.1 (2000-09)  
Harmonised EN covering essential requirements  
under Article 3.2 of the R&TTE Directive

Certificate of Conformance Serial No:  
RFIMPTB1/SC42137F

This is to certify that a sample of the equipment defined above has been tested by Radio Frequency Investigation Ltd (Basingstoke) to the above standard as detailed in RFI report number RFIMPTB1/SC42137F. In the configuration tested the equipment was compliant with the standard or parts of the standard applying to the sample.	
For full details please refer to document RFIMPTB1/SC42137F	
Tested By:	Approved Signatory:
Certificate Copy No: 01	Test Item Receipt Date: 27 June 2001
Issue Date: 17 October 2001	Test Date: 02 August 2001 to 10 September 2001
Issued to: Mr G. Sharples Managing Director Radcometrix Ltd Hartoran House Gibbs Couch Carpenters Park Watford Hertfordshire WD1 5EZ	Deviations from tested standard: None.
This certificate may be reproduced in full. Partial reproduction may only be made with the written consent of Radio Frequency Investigation Ltd. This certificate applies only to the sample(s) tested.	

A CATALYST IN WIRELESS DEVELOPMENT



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

EN 301 489-3 V1.2.1 (2000-08)  
Part 3: RPTT

EN 301 489-3 V1.2.1 (2000-08)  
Part 3: RPTT

EN 301 489-3 V1.2.1 (2000-08)  
Part 3: RPTT

# CERTIFICATE OF CONFORMANCE FROM RADIO FREQUENCY INVESTIGATION LTD.

Test Of Radiometrix Ltd  
B M2-433

To EN 301 489-3 V1.2.1 (2000-08)  
according to the specific conditions  
stated in EN 301 489-3 V1.2.1 (2000-08)  
(requested parts only)

Certificate of Conformance Serial No.  
RFIEMCB1/TC42137ETF13A

This is to certify that a sample of the equipment defined above has been tested by Radio Frequency Investigation Ltd (Basingstoke) to the above standard as detailed in RFI report number RFIEMCB1/TC42137ETF13A. In the configuration tested the equipment was compliant with the standard or parts of the standard applied to the sample.

For full details please refer to document RFIEMCB1/TC42137ETF13A

Tested By:

Approved Signatory:

Certificate Copy No:

Test Item Receipt Date: 11 June 2001

Issue Date: 29 July 2001

Test Date: 29 June 2001

Issued to:

Radiometrix Ltd  
Hamman House  
Gibbs Court  
Carpenters Park  
Watford  
WC1 5EZ  
Hertfordshire

Deviations from tested standard:

None

This certificate may be reproduced in full. Partial reproduction may only be made with the written consent of Radio Frequency Investigation Ltd.  
This certificate applies only to the sample(s) tested.

# CE Certificate for BiM2-433-64 and its variants



RFI Limited, 100, The Quadrant, Watlington, Oxford, OX11 0AB, UK

## NOTIFIED BODY OPINION RFI/NBCB1/42627JD03

### Applicants Details

**Applicant's Company Name:** Radiometrix Ltd  
**Address:** Hartran House, Gibbs Cough, Carpenters Park, Watford, Herts  
WD17 5EZ  
**Contact Name:** Mr S. Paramanathan

### Manufacturers Details

**Manufacturer's Company Name:** Radiometrix Ltd  
**Address:** Hartran House, Gibbs Cough, Carpenters Park, Watford, Herts  
WD17 5EZ  
**Contact Name:** Mr S. Paramanathan

### Product Details

**Product Name:** BiM2-433  
**Model Number(s):** BiM2-433-64-5V  
BiM2-433-64-3V  
BiM2-433-64-S  
**TCF Reference Number:** RM/BiM2/D1D  
**Standard(s):** EN 300 220-1 V1 3 1 (2000-09) according to the requirements of EN  
300 220-3 V1 1 1 (2000-09)  
**Description:** FM Radio Transceiver Module

In the opinion of RFI, the Technical Construction File demonstrates that the Apparatus to which it relates conforms to the essential requirements of Article 3.2 of Council Directive 1999/5/EC on radio equipment and telecommunications terminal equipment and the mutual recognition of their conformity. The Apparatus cannot be used with a loop antenna.

Authorised Signatory

Mr N Cobb

10/10/2011

Date



Notified Body Number: 0889

Company Seal

RFI Limited, 100, The Quadrant, Watlington, Oxford, OX11 0AB, UK  
Tel: +44 (0)1235 407100 Fax: +44(0)1235 407101 E-mail: info@rfi.co.uk  
registered in England No 211 090

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

## **Radiometrix Ltd**

**Hartcran House, Gibbs Couch, Watford, WD19 5EZ, ENGLAND**

**Tel: +44 (0)20 8428 1220, Fax: +44 (0)20 8428 1221**

**info@radiometrix.co.uk**

**www.radiometrix.co.uk**

### **UK Distributors**

#### **Low Power Radio Solutions Ltd**

Two Rivers Industrial Estate, Station Lane, Witney, Oxon OX28 4BH

Freephone: 0800 0744157, Tel: 01993 709418, Fax: 01993 708575

info@lprs.co.uk

#### **RS Components**

Tel: 01536 201201

#### **Farnell**

Tel: 0870 1200 200

#### **CPC**

Tel: 08701 20 25 30

### **Copyright notice**

*This product data sheet is the original work and copyrighted property of Radiometrix Ltd. Reproduction in whole or in part must give clear acknowledgement to the copyright owner.*

### **Limitation of liability**

*The information furnished by Radiometrix Ltd is believed to be accurate and reliable. Radiometrix Ltd reserves the right to make changes or improvements in the design, specification or manufacture of its subassembly products without notice. Radiometrix Ltd does not assume any liability arising from the application or use of any product or circuit described herein, nor for any infringements of patents or other rights of third parties which may result from the use of its products. This data sheet neither states nor implies warranty of any kind, including fitness for any particular application. These radio devices may be subject to radio interference and may not function as intended if interference is present. We do NOT recommend their use for life critical applications.*

*The Intrastat commodity code for all our modules is: 8542 4090.*

### **R&TTE Directive**

*After 7 April 2001 the manufacturer can only place finished product on the market under the provisions of the R&TTE Directive. Equipment within the scope of the R&TTE Directive may demonstrate compliance to the essential requirements specified in Article 3 of the Directive, as appropriate to the particular equipment.*

*Further details are available on Radiocommunications Agency (RA) web site:*

**<http://www.radio.gov.uk/topics/conformity/conform-index.htm>**

The Library and Information Service

The Radiocommunications Agency

Wyndham House

189 Marsh Wall

London E14 9SX

United Kingdom

Tel: +44 (0)20 7211 0502/0505

Fax: +44 (0)20 7211 0507

library@ra.gsi.gov.uk

*For further information on radio matters*

*contact the Agency's 24 Hour Telephone*

*Enquiry Point: +44 (0)20 7211 0211*

European Radiocommunications Office (ERO)

Midtermolen 1

DK 2100 Copenhagen

Denmark

Tel. +45 35250300

Fax +45 35250330

ero@ero.dk

www.ero.dk

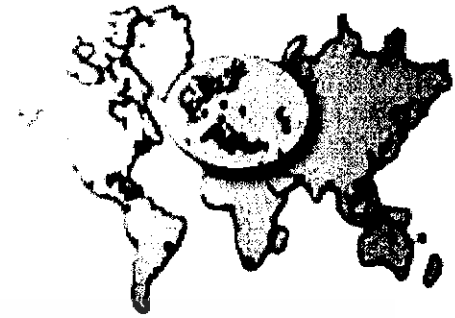
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

# Radiometrix Ltd

## Worldwide Distributors

### Radiometrix Ltd

Hartcran House  
Gibbs Couch  
Watford  
WD19 5EZ  
ENGLAND  
Tel: +44 (0)20 8428 1220  
Fax: +44 (0)20 8428 1221  
info@radiometrix.com  
www.radiometrix.com



#### Telelink Communications

PO Box 5457, North Rockhampton, Queensland 4702  
AUSTRALIA (including South Pacific)  
Tel: +61 7 49340413, Fax +61 7 49340311  
24 hour Message Service +61 7 38300233  
sales@telelink.com.au

#### RS do Brasil Ltda.

Av. Brigadeiro Faria Lima 2413 (6º andar)  
01451-001 São Paulo - SP

#### BRAZIL

Tel: +55 11 3819 0429, Fax: +55 11 3097 0009 or 11 3815 1162  
vendas@rsdobrasil.com.br

#### C-88 AS

Savsvinget 7, DK-2970 Hørsholm

#### DENMARK

Tel: +45 7010 4888, Mobile: +45 2320 8589, Fax: +45 7010 4889  
jr@c88.dk

#### Lextronic

36/40 Rue du Gal de Gaulle, 94510 La Queue en Brie

#### FRANCE

Tél: +33 (0)1 4576 8388, Fax: +33 (0)1 4576 8141  
infos@lextronic.fr

#### Haril

2A Tsokopoulou Street, 152 37 Filothei, Athens

#### GREECE

Tel : +30 1 6810338, Fax: +30 1 6853359  
harico@otenet.gr

#### Microrobot Co. Ltd.

Bowoo Bldg., 40-26, Cheongdam-dong, Kangnam-gu,  
Seoul, 135-102

#### KOREA

Tel: +82 2 540 1710, Fax: +82 2 540 1720  
maroboss@microrobot.com

#### Wireless World AG

Im Gree 79, CH-8566 Ellighausen

#### SWITZERLAND

Tel: +41-71-698 6480, Fax: +41-71-698 6481  
info@wirelessworldag.ch

#### IDVISION B.V.B.A.

Augustijnenstraat 44  
B-8900 Ieper

#### BELGIUM (including NETHERLANDS, LUXEMBOURG)

Tel.: +32 57 216141, Fax: +32 57 216434  
info@idvision.net

#### Advanced Radio Telemetry

Francouzská 82, 602 00 Brno

Tel.: +420 (5)4521 1403, Fax: +420 (5)4521 0506

#### CZECH REPUBLIC

art@artbrno.cz

#### TQ Electronic Oy

Suurtohjankatu 22, 08100 Lohja

#### FINLAND

Tel: +358 19 326451, Mobile: +358 400 670 697, Fax: +358 19  
326452  
raimo@tqelectronic.fi

#### HY-LINE Communication Products GmbH

Inselkammerstraße 10, D-82008 Unterhaching

#### GERMANY (including AUSTRIA)

Tel: +49 89 61450319, Fax: +49 89 6140960  
communication@hy-line.de

#### Avnet Components Israel Ltd

P.O. BOX 48, Tel Mond, 40600

#### ISRAEL

Tel : +972 9 7966999, Fax: +972 3 7601115  
shuki.herzlinger@avnet.com

#### Rotrax Technologies

P.O. Box 2499, North Riding, 2162,

#### SOUTH AFRICA

Tel: +27 11 794 4270, Fax: +27 11 794 5527  
rotrax@iafrica.com

#### Lemos International Co.Inc.

48 Sword Street, Auburn, MA 01501

#### UNITED STATES OF AMERICA

Tel: +1 508 798 5004, Fax: +1 508 798 4782  
sales@lemosint.com

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

December 1993

### Features

- Meets All RS-232C Specifications
- Requires Only Single +5V Power Supply
- Onboard Voltage Doubler/Inverter
- Low Power Consumption
- 2 Drivers
  - $\pm 9V$  Output Swing for +5V Input
  - $300\Omega$  Power-off Source Impedance
  - Output Current Limiting
  - TTL/CMOS Compatible
  - $30V/\mu s$  Maximum Slew Rate
- 2 Receivers
  - $\pm 30V$  Input Voltage Range
  - $3k\Omega$  to  $7k\Omega$  Input Impedance
  - 0.5V Hysteresis to Improve Noise Rejection
- All Critical Parameters are Guaranteed Over the Entire Commercial, Industrial and Military Temperature Ranges

### Applications

- Any System Requiring RS-232 Communications Port
  - Computer - Portable and Mainframe
  - Peripheral - Printers and Terminals
  - Portable Instrumentation
  - Modems
  - Dataloggers

### Description

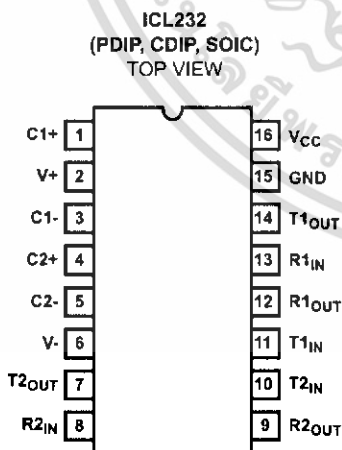
The ICL232 is a dual RS-232 transmitter/receiver interface circuit that meets all EIA RS-232C specifications. It requires a single +5V power supply, and features two onboard charge pump voltage converters which generate +10V and -10V supplies from the 5V supply.

The drivers feature true TTL/CMOS input compatibility, slew-rate-limited output, and  $300\Omega$  power-off source impedance. The receivers can handle up to +30V, and have a  $3k\Omega$  to  $7k\Omega$  input impedance. The receivers also have hysteresis to improve noise rejection.

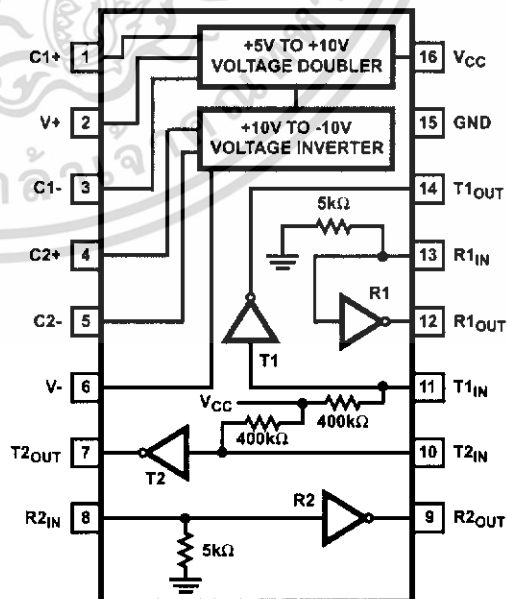
### Ordering Information

PART NUMBER	TEMPERATURE RANGE	PACKAGE
ICL232CPE	0°C to +70°C	16 Lead Plastic DIP
ICL232CJE	0°C to +70°C	16 Lead Ceramic DIP
ICL232CBE	0°C to +70°C	16 Lead SOIC (W)
ICL232IPE	-40°C to +85°C	16 Lead Plastic DIP
ICL232IJE	-40°C to +85°C	16 Lead Ceramic DIP
ICL232IBE	-40°C to +85°C	16 Lead SOIC (W)
ICL232MJE	-55°C to +125°C	16 Lead Ceramic DIP

### Pinouts



### Functional Diagram



CAUTION: These devices are sensitive to electrostatic discharge. Users should follow proper I.C. Handling Procedures.

File Number 3020.2

Copyright © Harris Corporation 1993

11-8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Specifications ICL232

### Absolute Maximum Ratings

$V_{CC}$ to Ground	(GND -0.3V) < $V_{CC}$ < 6V
V+ to Ground	( $V_{CC}$ -0.3V) < V+ < 12V
V- to Ground	-12V < V- < (GND +0.3V)
Input Voltages	
$T1_{IN}, T2_{IN}$	(V- -0.3V) < $V_{IN}$ < (V+ +0.3V)
$R1_{IN}, R2_{IN}$	$\pm 30V$
Output Voltages	
$T1_{OUT}, T2_{OUT}$	(V- -0.3V) < $V_{TXOUT}$ < (V+ +0.3V)
$R1_{OUT}, R2_{OUT}$	(GND -0.3V) < $V_{RXOUT}$ < ( $V_{CC}$ +0.3V)
Short Circuit Duration	
$T1_{OUT}, T2_{OUT}$	Continuous
$R1_{OUT}, R2_{OUT}$	Continuous
Storage Temperature Range	-65°C to +150°C
Lead Temperature (Soldering 10s)	+300°C

### Thermal Information

Thermal Resistance	$\theta_{JA}$	$\theta_{JC}$
Ceramic DIP Package	80°C/W	24°C/W
Plastic DIP Package	100°C/W	-
SOIC Package	100°C/W	-
Maximum Power Dissipation	250mW	
Operating Temperature Range		
ICL232C	0°C to +70°C	
ICL232I	-40°C to +85°C	
ICL232M	-55°C to +125°C	

CAUTION: Stresses above those listed in "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress only rating and operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

### Electrical Specifications

Test Conditions:  $V_{CC} = +5V \pm 10\%$ ,  $T_A =$  Operating Temperature Range. Test Circuit as in Figure 8 Unless Otherwise Specified

PARAMETER	TEST CONDITIONS	LIMITS			UNITS
		MIN	TYP	MAX	
Transmitter Output Voltage Swing, $T_{OUT}$	$T1_{OUT}$ and $T2_{OUT}$ loaded with 3k $\Omega$ to Ground	$\pm 5$	$\pm 9$	$\pm 10$	V
Power Supply Current, $I_{CC}$	Outputs Unloaded, $T_A = +25^\circ C$	-	5	10	mA
$T_{IN}$ , Input Logic Low, $V_{IL}$		-	-	0.8	V
$T_{IN}$ , Input Logic High, $V_{IH}$		2.0	-	-	V
Logic Pullup Current, $I_p$	$T1_{IN}, T2_{IN} = 0V$	-	15	200	$\mu A$
RS-232 Input Voltage Range, $V_{IN}$		-30	-	+30	V
Receiver Input Impedance, $R_{IN}$	$V_{IN} = \pm 3V$	3.0	5.0	7.0	k $\Omega$
Receiver Input Low Threshold, $V_{IN}$ (H-L)	$V_{CC} = 5.0V, T_A = +25^\circ C$	0.8	1.2	-	V
Receiver Input High Threshold, $V_{IN}$ (L-H)	$V_{CC} = 5.0V, T_A = +25^\circ C$	-	1.7	2.4	V
Receiver Input Hysteresis, $V_{HYST}$		0.2	0.5	1.0	V
TTL/CMOS Receiver Output Voltage Low, $V_{OL}$	$I_{OUT} = 3.2mA$	-	0.1	0.4	V
TTL/CMOS Receiver Output Voltage High, $V_{OH}$	$I_{OUT} = -1.0mA$	3.5	4.6	-	V
Propagation Delay, $t_{PD}$	RS-232 to TTL	-	0.5	-	$\mu s$
Instantaneous Slew Rate, SR	$C_L = 10pF, R_L = 3k\Omega, T_A = +25^\circ C$ (Notes 1, 2)	-	-	30	V/ $\mu s$
Transition Region Slew Rate, $SR_T$	$R_L = 3k\Omega, C_L = 2500pF$ Measured from +3V to -3V or -3V to +3V	-	3	-	V/ $\mu s$
Output Resistance, $R_{OUT}$	$V_{CC} = V+ = V- = 0V, V_{OUT} = \pm 2V$	300	-	-	$\Omega$
RS-232 Output Short Circuit Current, $I_{SC}$	$T1_{OUT}$ or $T2_{OUT}$ shorted to GND	-	$\pm 10$	-	mA

#### NOTES:

1. Guaranteed by design.
2. See Figure 4 for definition.

## ICL232

### Typical Performance Curves

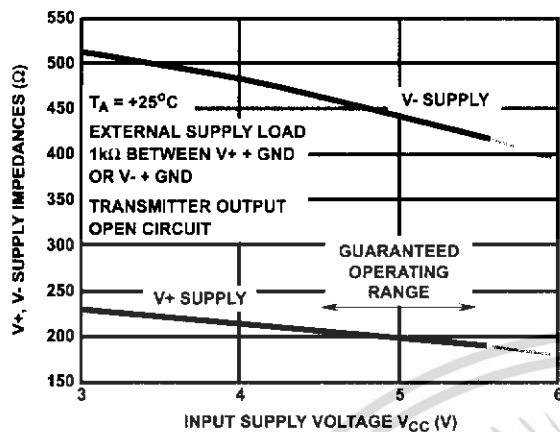


FIGURE 1. V+, V- OUTPUT IMPEDANCES vs  $V_{CC}$

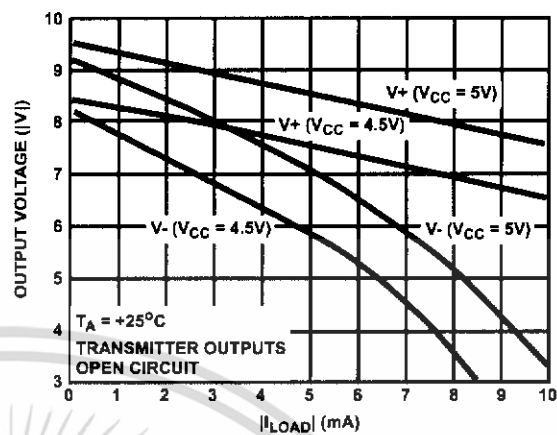


FIGURE 2. V+, V- OUTPUT VOLTAGES vs LOAD CURRENT

### Pin Descriptions

PLASTIC DIP, CERAMIC DIP	SOIC	PIN NAME	DESCRIPTION
1	1	C1+	External capacitor "+" for internal voltage doubler.
2	2	V+	Internally generated +10V (typical) supply.
3	3	C1-	External capacitor "-" for internal voltage doubler.
4	4	C2+	External capacitor "+" internal voltage inverter.
5	5	C2-	External capacitor "-" internal voltage inverter.
6	6	V-	Internally generated -10V (typical) supply.
7	7	T2 <sub>OUT</sub>	RS-232 Transmitter 2 output $\pm 10V$ (typical).
8	8	R2 <sub>IN</sub>	RS-232 Receiver 2 input, with internal 5K pulldown resistor to GND.
9	9	R2 <sub>OUT</sub>	Receiver 2 TTL/CMOS output.
10	10	T2 <sub>IN</sub>	Transmitter 2 TTL/CMOS input, with internal 400K pullup resistor to $V_{CC}$ .
11	11	T1 <sub>IN</sub>	Transmitter 1 TTL/CMOS input, with internal 400K pullup resistor to $V_{CC}$ .
12	12	R1 <sub>OUT</sub>	Receiver 1 TTL/CMOS output.
13	13	R1 <sub>IN</sub>	RS-232 Receiver 1 input, with internal 5K pulldown resistor to GND.
14	14	T1 <sub>OUT</sub>	RS-232 Transmitter 1 output $\pm 10V$ (typical).
15	15	GND	Supply Ground.
16	16	VCC	Positive Power Supply +5V $\pm 10\%$

**Detailed Description**

The ICL232 is a dual RS-232 transmitter/receiver powered by a single +5V power supply which meets all EIA RS232C specifications and features low power consumption. The functional diagram illustrates the major elements of the ICL232. The circuit is divided into three sections: a voltage doubler/inverter, dual transmitters, and dual receivers.

**Voltage Converter**

An equivalent circuit of the dual charge pump is illustrated in Figure 3.

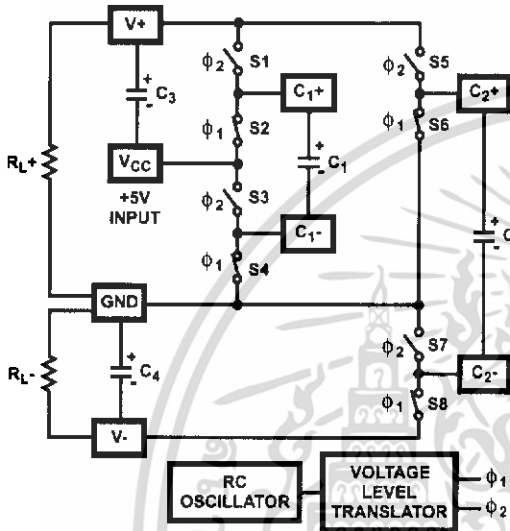


FIGURE 3. DUAL CHARGE PUMP

The voltage quadrupler contains two charge pumps which use two phases of an internally generated clock to generate +10V and -10V. The nominal clock frequency is 16kHz. During phase one of the clock, capacitor C1 is charged to V<sub>CC</sub>. During phase two, the voltage on C1 is added to V<sub>CC</sub>, producing a signal across C2 equal to twice V<sub>CC</sub>. At the same time, C3 is also charged to 2V<sub>CC</sub>, and then during phase one, it is inverted with respect to ground to produce a signal across C4 equal to -2V<sub>CC</sub>. The voltage converter accepts input voltages up to 5.5V. The output impedance of the doubler (V<sub>+</sub>) is approximately 200Ω, and the output impedance of the inverter (V<sub>-</sub>) is approximately 450Ω. Typical graphs are presented which show the voltage converters output vs input voltage and output voltages vs load characteristics. The test circuit (Figure 8) uses 1μF capacitors for C1-C4, however, the value is not critical. Increasing the values of C1 and C2 will lower the output impedance of the voltage doubler and inverter, and increasing the values of the reservoir capacitors, C3 and C4, lowers the ripple on the V<sub>+</sub> and V<sub>-</sub> supplies.

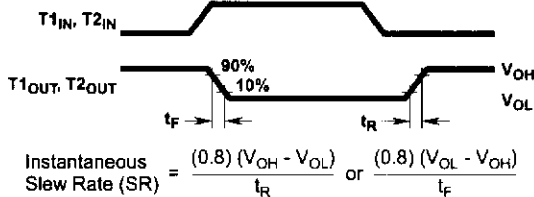


FIGURE 4. SLEW RATE DEFINITION

$$\text{Instantaneous Slew Rate (SR)} = \frac{(0.8)(V_{OH} - V_{OL})}{t_r} \text{ OR } \frac{(0.8)(V_{OL} - V_{OH})}{t_f}$$

**Transmitters**

The transmitters are TTL/CMOS compatible inverters which translate the inputs to RS-232 outputs. The input logic threshold is about 26% of V<sub>CC</sub>, or 1.3V for V<sub>CC</sub> = 5V. A logic 1 at the input results in a voltage of between -5V and V<sub>-</sub> at the output, and a logic 0 results in a voltage between +5V and (V<sub>+</sub> - 0.6V). Each transmitter input has an internal 400kΩ pullup resistor so any unused input can be left unconnected and its output remains in its low state. The output voltage swing meets the RS-232C specification of ±5V minimum with the worst case conditions of: both transmitters driving 3kΩ minimum load impedance, V<sub>CC</sub> = 4.5V, and maximum allowable operating temperature. The transmitters have an internally limited output slew rate which is less than 30V/μs. The outputs are short circuit protected and can be shorted to ground indefinitely. The powered down output impedance is a minimum of 300Ω with ±2V applied to the outputs and V<sub>CC</sub> = 0V.

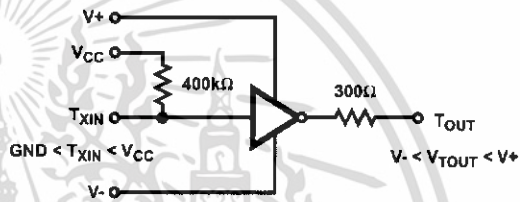


FIGURE 5. TRANSMITTER

**Receivers**

The receiver inputs accept up to ±30V while presenting the required 3kΩ to 7kΩ input impedance even if the power is off (V<sub>CC</sub> = 0V). The receivers have a typical input threshold of 1.3V which is within the ±3V limits, known as the transition region, of the RS-232 specification. The receiver output is 0V to V<sub>CC</sub>. The output will be low whenever the input is greater than 2.4V and high whenever the input is floating or driven between +0.8V and -30V. The receivers feature 0.5V hysteresis to improve noise rejection.

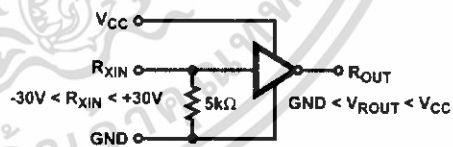


FIGURE 6. RECEIVER

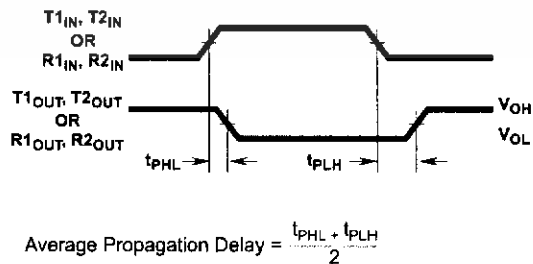


FIGURE 7. PROPAGATION DELAY DEFINITION

$$\text{Average Propagation Delay} = \frac{t_{PLH} + t_{PLH}}{2}$$

Test Circuits

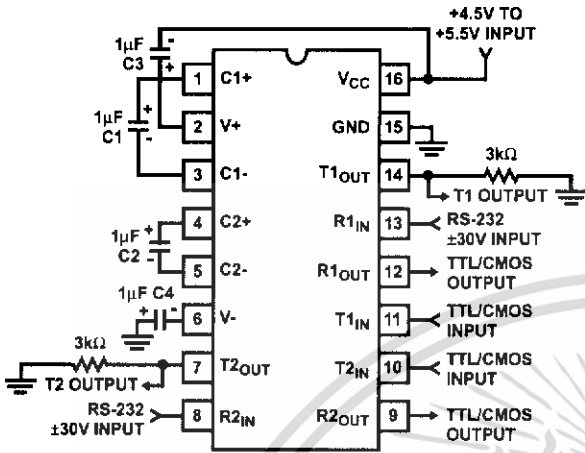


FIGURE 8. GENERAL TEST CIRCUIT

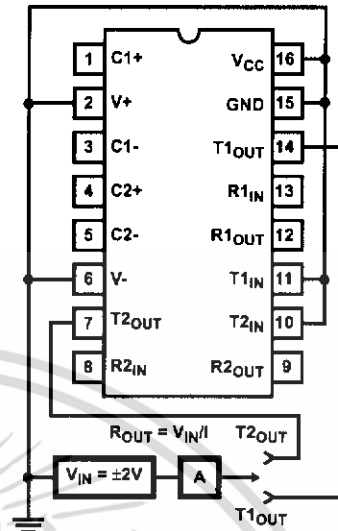


FIGURE 9. POWER-OFF SOURCE RESISTANCE CONFIGURATION

Applications

The ICL232 may be used for all RS-232 data terminal and communication links. It is particularly useful in applications where ±12V power supplies are not available for conventional RS-232 interface circuits. The applications presented represent typical interface configurations.

A simple duplex RS-232 port with CTS/RTS handshaking is illustrated in Figure 10. Fixed output signals such as DTR (data terminal ready) and DSRs (data signaling rate select) is generated by driving them through a 5kΩ resistor connected to V+.

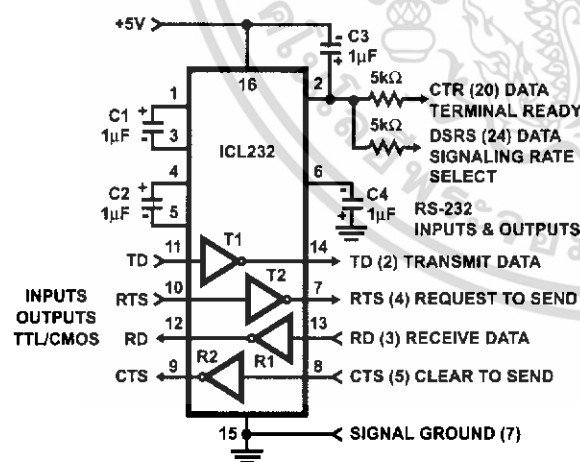


FIGURE 10. SIMPLE DUPLEX RS-232 PORT WITH CTS/RTS HANDSHAKING

In applications requiring four RS-232 inputs and outputs (Figure 11), note that each circuit requires two charge pump capacitors (C1 and C2) but can share common reservoir

capacitors (C3 and C4). The benefit of sharing common reservoir capacitors is the elimination of two capacitors and the reduction of the charge pump source impedance which effectively increases the output swing of the transmitters.

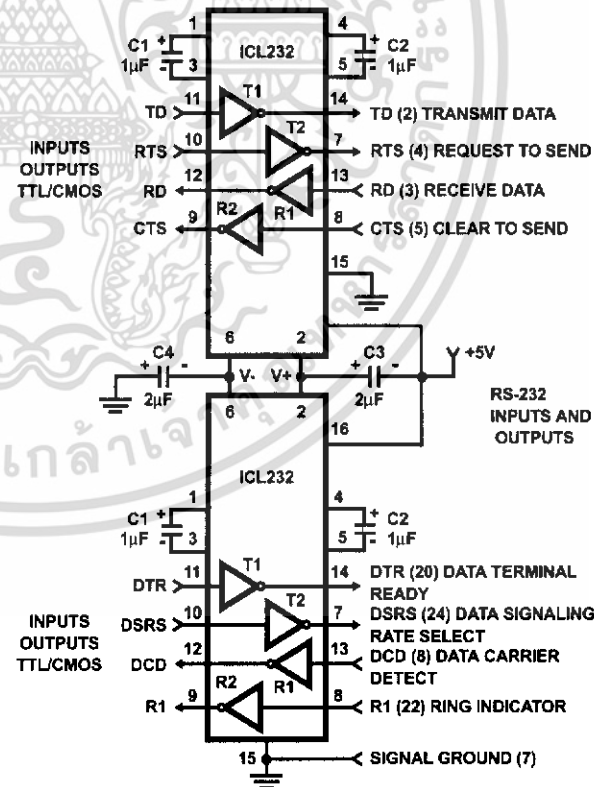


FIGURE 11. COMBINING TWO ICL232s FOR 4 PAIRS OF RS-232 INPUTS AND OUTPUTS



# 8-bit Microcontroller with 4K Bytes Flash

## AT89C51

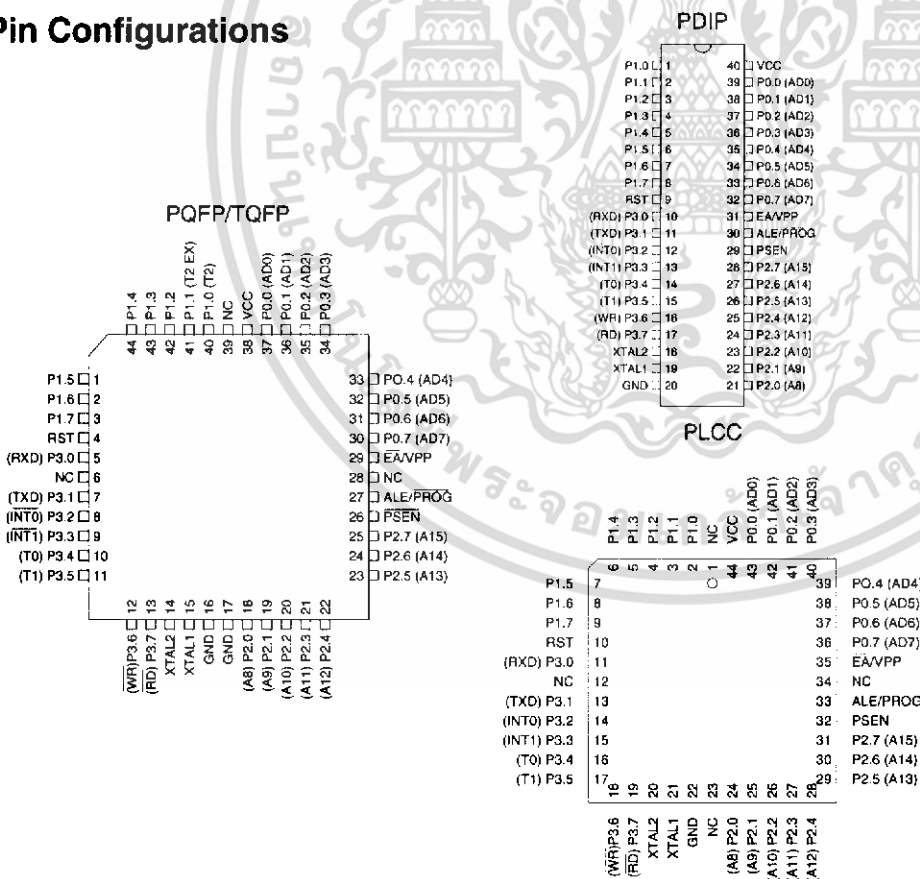
### Features

- Compatible with MCS-51™ Products
- 4K Bytes of In-System Reprogrammable Flash Memory
  - Endurance: 1,000 Write/Erase Cycles
- Fully Static Operation: 0 Hz to 24 MHz
- Three-level Program Memory Lock
- 128 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-bit Timer/Counters
- Six Interrupt Sources
- Programmable Serial Channel
- Low-power Idle and Power-down Modes

### Description

The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4K bytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard MCS-51 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C51 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications.

### Pin Configurations

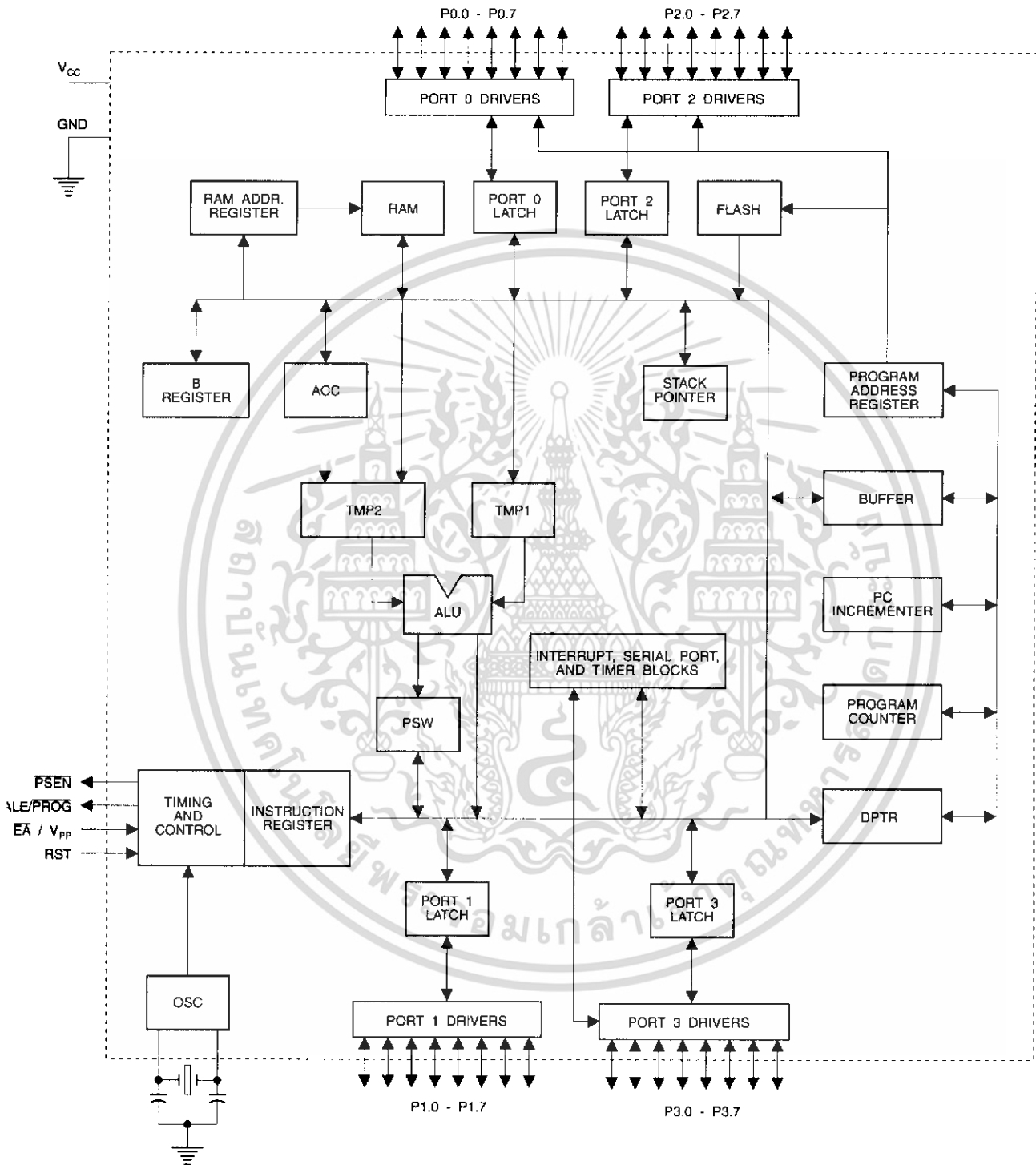


Rev. 0265G-02/00



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The AT89C51 provides the following standard features: 4K bytes of Flash, 128 bytes of RAM, 32 I/O lines, two 16-bit timer/counters, a five vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator and clock circuitry. In addition, the AT89C51 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port and interrupt system to continue functioning. The Power-down Mode saves the RAM contents but freezes the oscillator disabling all other chip functions until the next hardware reset.

## Pin Description

### VCC

Supply voltage.

### GND

Ground.

### Port 0

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

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

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

### Port 1

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

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

### Port 2

Port 2 is an 8-bit bi-directional I/O port with internal pullups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins they are pulled high by the internal pullups and can be used as inputs. As inputs,

Port 2 pins that are externally being pulled low will source current ( $I_{IL}$ ) because of the internal pullups.

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

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

### Port 3

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

Port 3 also serves the functions of various special features of the AT89C51 as listed below:

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)

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

### RST

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

### ALE/PROG

Address Latch Enable output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input (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 weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

### PSEN

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

When the AT89C51 is executing code from external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory.

### EA/VPP

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

EA should be strapped to V<sub>CC</sub> for internal program executions.

This pin also receives the 12-volt programming enable voltage (V<sub>PP</sub>) during Flash programming, for parts that require 12-volt V<sub>PP</sub>.

### XTAL1

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

### XTAL2

Output from the inverting oscillator amplifier.

## Oscillator Characteristics

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left

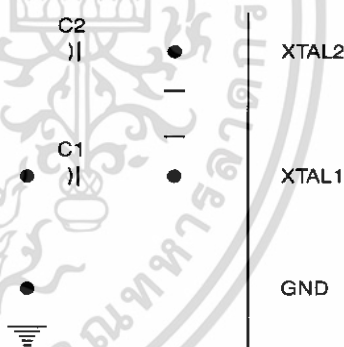
unconnected while XTAL1 is driven as shown in Figure 2. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

## Idle Mode

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

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

Figure 1. Oscillator Connections

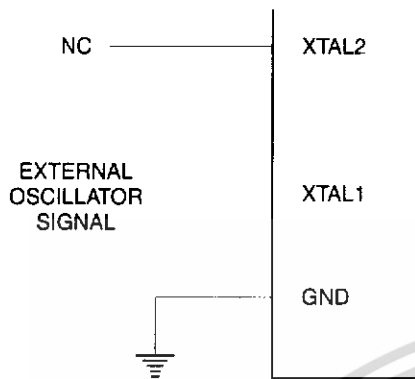


Note: C1, C2 = 30 pF ± 10 pF for Crystals  
= 40 pF ± 10 pF for Ceramic Resonators

## Status of External Pins During Idle and Power-down Modes

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

**Figure 2.** External Clock Drive Configuration



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

### Program Memory Lock Bits

On the chip are three lock bits which can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the table below.

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

### Power-down Mode

In the power-down mode, the oscillator is stopped, and the instruction that invokes power-down is the last instruction executed. The on-chip RAM and Special Function Regis-

### Lock Bit Protection Modes

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



## Programming the Flash

The AT89C51 is normally shipped with the on-chip Flash memory array in the erased state (that is, contents = FFH) and ready to be programmed. The programming interface accepts either a high-voltage (12-volt) or a low-voltage ( $V_{CC}$ ) program enable signal. The low-voltage programming mode provides a convenient way to program the AT89C51 inside the user's system, while the high-voltage programming mode is compatible with conventional third-party Flash or EPROM programmers.

The AT89C51 is shipped with either the high-voltage or low-voltage programming mode enabled. The respective top-side marking and device signature codes are listed in the following table.

	$V_{PP} = 12V$	$V_{PP} = 5V$
Top-Side Mark	AT89C51 xxxx yyww	AT89C51 xxxx-5 yyww
Signature	(030H) = 1EH (031H) = 51H (032H) = FFH	(030H) = 1EH (031H) = 51H (032H) = 05H

The AT89C51 code memory array is programmed byte-by-byte in either programming mode. *To program any non-blank byte in the on-chip Flash Memory, the entire memory must be erased using the Chip Erase Mode.*

**Programming Algorithm:** Before programming the AT89C51, the address, data and control signals should be set up according to the Flash programming mode table and Figure 3 and Figure 4. To program the AT89C51, take the following steps.

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

and data for the entire array or until the end of the object file is reached.

**Data Polling:** The AT89C51 features Data Polling to indicate the end of a write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written datum on PO.7. Once the write cycle has been completed, true data are 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. P3.4 is pulled low after ALE goes high during programming to indicate BUSY. P3.4 is pulled high again when programming is done to indicate READY.

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

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

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

- (030H) = 1EH indicates manufactured by Atmel
- (031H) = 51H indicates 89C51
- (032H) = FFH indicates 12V programming
- (032H) = 05H indicates 5V programming

## Programming Interface

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

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

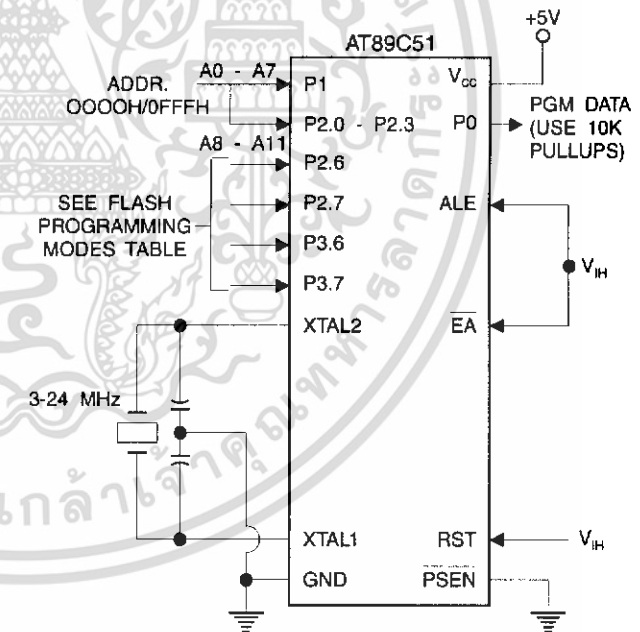
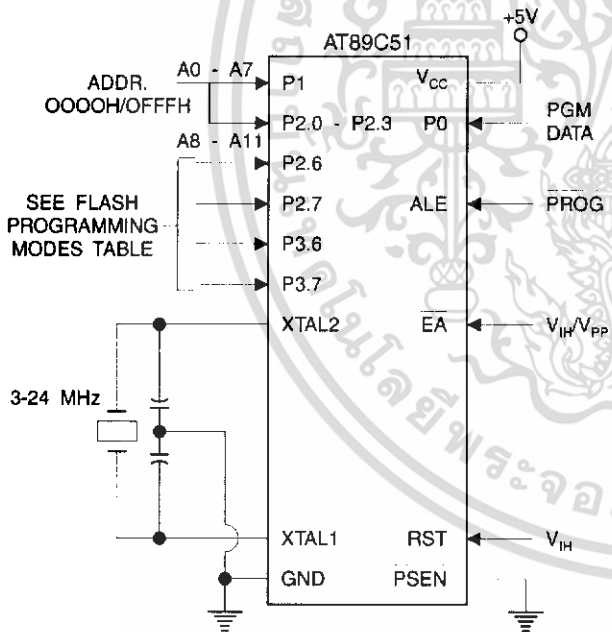
## Flash Programming Modes

Mode	RST	PSEN	ALE/PROG	EA/V <sub>PP</sub>	P2.6	P2.7	P3.6	P3.7
Write Code Data	H	L		H/12V	L	H	H	H
Read Code Data	H	L	H	H	L	L	H	H
Write Lock	Bit - 1	H		H/12V	H	H	H	H
	Bit - 2	H		H/12V	H	H	L	L
	Bit - 3	H		H/12V	H	L	H	L
Chip Erase	H	L		H/12V	H	L	L	L
Read Signature Byte	H	L	H	H	L	L	L	L

Note: 1. Chip Erase requires a 10 ms PROG pulse.

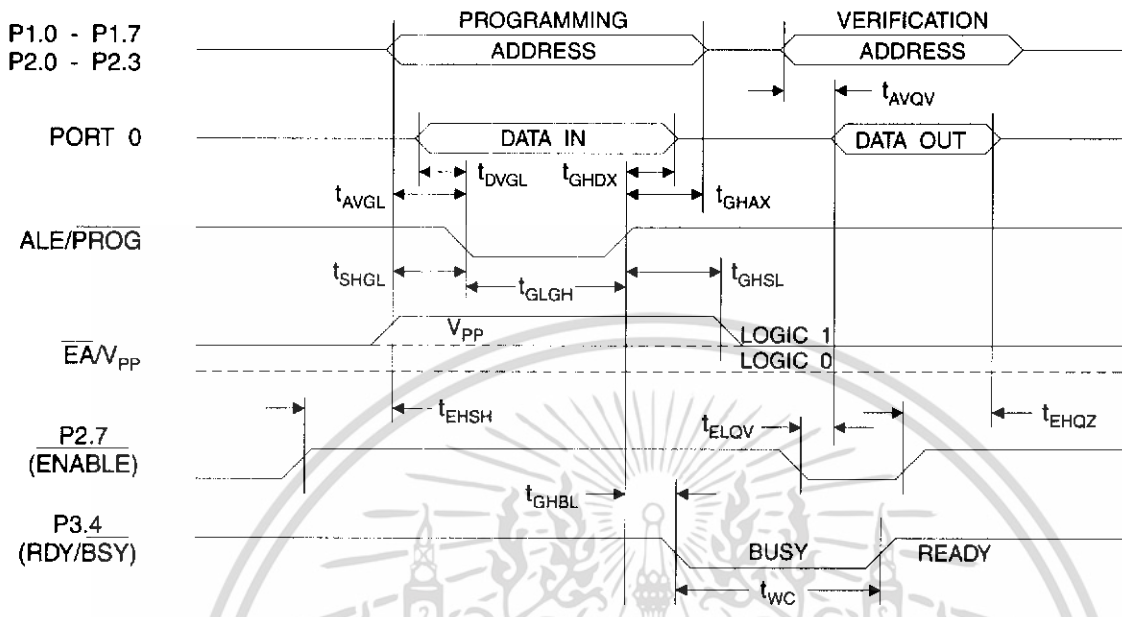
Figure 3. Programming the Flash

Figure 4. Verifying the Flash

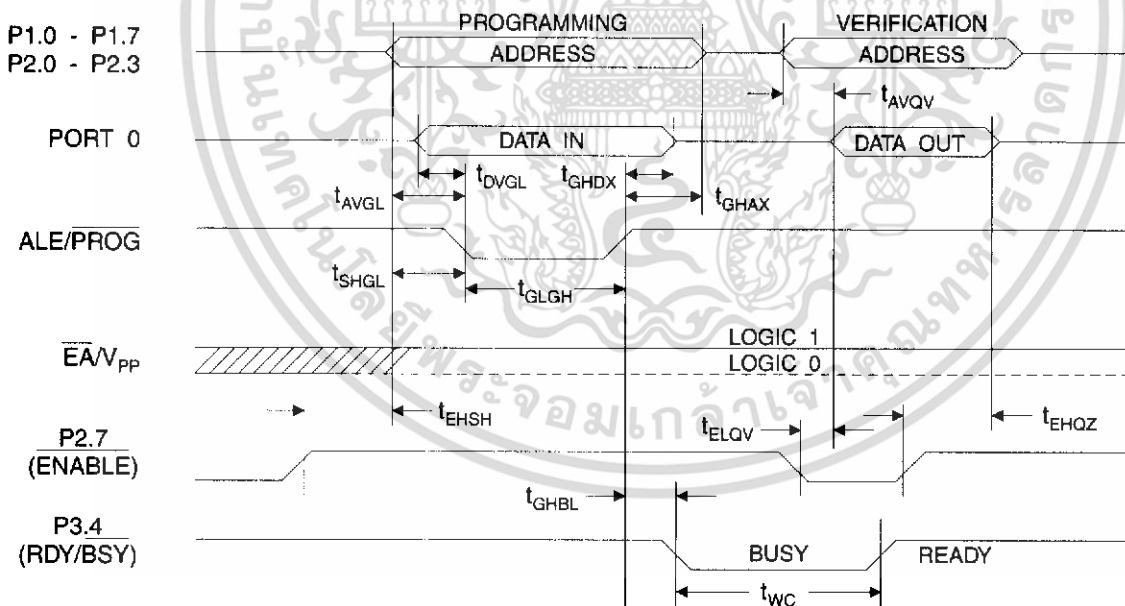




### Flash Programming and Verification Waveforms - High-voltage Mode ( $V_{PP} = 12V$ )



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



## Flash Programming and Verification Characteristics

$T_A = 0^\circ\text{C}$  to  $70^\circ\text{C}$ ,  $V_{CC} = 5.0 \pm 10\%$

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

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



## Absolute Maximum Ratings\*

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

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

## DC Characteristics

$T_A = -40^\circ\text{C}$  to  $85^\circ\text{C}$ ,  $V_{CC} = 5.0\text{V} \pm 20\%$  (unless otherwise noted)

Symbol	Parameter	Condition	Min	Max	Units
$V_{IL}$	Input Low-voltage	(Except EA)	-0.5	$0.2 V_{CC} - 0.1$	V
$V_{IL1}$	Input Low-voltage (EA)		-0.5	$0.2 V_{CC} - 0.3$	V
$V_{IH}$	Input High-voltage	(Except XTAL1, RST)	$0.2 V_{CC} + 0.9$	$V_{CC} + 0.5$	V
$V_{IH1}$	Input High-voltage	(XTAL1, RST)	$0.7 V_{CC}$	$V_{CC} + 0.5$	V
$V_{OL}$	Output Low-voltage <sup>(1)</sup> (Ports 1,2,3)	$I_{OL} = 1.6 \text{ mA}$		0.45	V
$V_{OL1}$	Output Low-voltage <sup>(1)</sup> (Port 0, ALE, PSEN)	$I_{OL} = 3.2 \text{ mA}$		0.45	V
$V_{OH}$	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
$V_{OH1}$	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
$I_{IL}$	Logical 0 Input Current (Ports 1,2,3)	$V_{IN} = 0.45\text{V}$		-50	$\mu\text{A}$
$I_{TL}$	Logical 1 to 0 Transition Current (Ports 1,2,3)	$V_{IN} = 2\text{V}$ , $V_{CC} = 5\text{V} \pm 10\%$		-650	$\mu\text{A}$
$I_{LI}$	Input Leakage Current (Port 0, EA)	$0.45 < V_{IN} < V_{CC}$		$\pm 10$	$\mu\text{A}$
RRST	Reset Pull-down Resistor		50	300	K $\Omega$
$C_{IO}$	Pin Capacitance	Test Freq. = 1 MHz, $T_A = 25^\circ\text{C}$		10	pF
$I_{CC}$	Power Supply Current	Active Mode, 12 MHz		20	mA
		Idle Mode, 12 MHz		5	mA
	Power-down Mode <sup>(2)</sup>	$V_{CC} = 6\text{V}$		100	$\mu\text{A}$
		$V_{CC} = 3\text{V}$		40	$\mu\text{A}$

- 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.

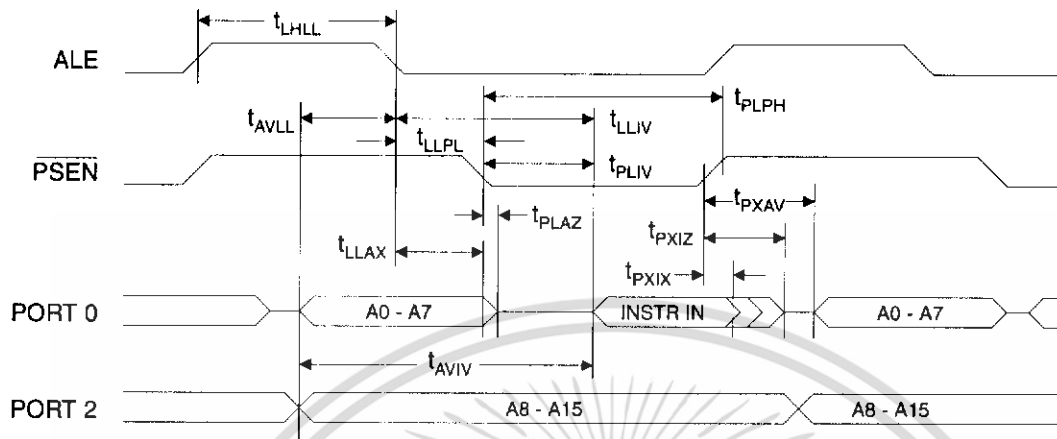
## AC Characteristics

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

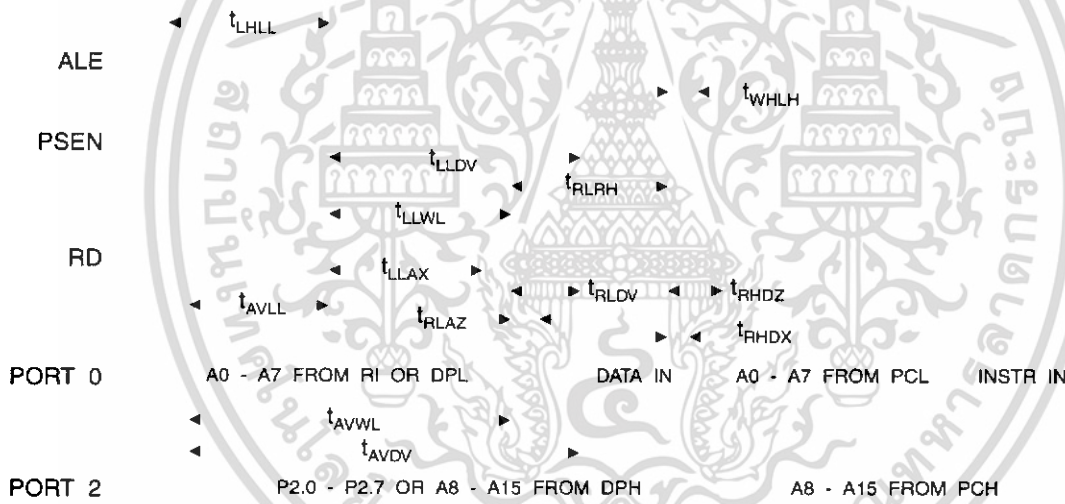
## External Program and Data Memory Characteristics

Symbol	Parameter	12 MHz Oscillator		16 to 24 MHz Oscillator		Units
		Min	Max	Min	Max	
$1/t_{\text{CLCL}}$	Oscillator Frequency			0	24	MHz
$t_{\text{LHLL}}$	ALE Pulse Width	127		$2t_{\text{CLCL}}-40$		ns
$t_{\text{AVLL}}$	Address Valid to ALE Low	43		$t_{\text{CLCL}}-13$		ns
$t_{\text{LLAX}}$	Address Hold After ALE Low	48		$t_{\text{CLCL}}-20$		ns
$t_{\text{LLIV}}$	ALE Low to Valid Instruction In		233		$4t_{\text{CLCL}}-65$	ns
$t_{\text{LLPL}}$	ALE Low to $\overline{\text{PSEN}}$ Low	43		$t_{\text{CLCL}}-13$		ns
$t_{\text{PLPH}}$	$\overline{\text{PSEN}}$ Pulse Width	205		$3t_{\text{CLCL}}-20$		ns
$t_{\text{PLIV}}$	$\overline{\text{PSEN}}$ Low to Valid Instruction In		145		$3t_{\text{CLCL}}-45$	ns
$t_{\text{PXIX}}$	Input Instruction Hold After $\overline{\text{PSEN}}$	0		0		ns
$t_{\text{PXIZ}}$	Input Instruction Float After $\overline{\text{PSEN}}$		59		$t_{\text{CLCL}}-10$	ns
$t_{\text{PXAV}}$	$\overline{\text{PSEN}}$ to Address Valid	75		$t_{\text{CLCL}}-8$		ns
$t_{\text{AVIV}}$	Address to Valid Instruction In		312		$5t_{\text{CLCL}}-55$	ns
$t_{\text{PLAZ}}$	$\overline{\text{PSEN}}$ Low to Address Float		10		10	ns
$t_{\text{RLRH}}$	$\overline{\text{RD}}$ Pulse Width	400		$6t_{\text{CLCL}}-100$		ns
$t_{\text{WLWH}}$	$\overline{\text{WR}}$ Pulse Width	400		$6t_{\text{CLCL}}-100$		ns
$t_{\text{RLDV}}$	$\overline{\text{RD}}$ Low to Valid Data In		252		$5t_{\text{CLCL}}-90$	ns
$t_{\text{RHDX}}$	Data Hold After $\overline{\text{RD}}$	0		0		ns
$t_{\text{RHDX}}$	Data Float After $\overline{\text{RD}}$		97		$2t_{\text{CLCL}}-28$	ns
$t_{\text{LLDV}}$	ALE Low to Valid Data In		517		$8t_{\text{CLCL}}-150$	ns
$t_{\text{AVDV}}$	Address to Valid Data In		585		$9t_{\text{CLCL}}-165$	ns
$t_{\text{LLWL}}$	ALE Low to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ Low	200	300	$3t_{\text{CLCL}}-50$	$3t_{\text{CLCL}}+50$	ns
$t_{\text{AVWL}}$	Address to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ Low	203		$4t_{\text{CLCL}}-75$		ns
$t_{\text{QVWX}}$	Data Valid to $\overline{\text{WR}}$ Transition	23		$t_{\text{CLCL}}-20$		ns
$t_{\text{QVWH}}$	Data Valid to $\overline{\text{WR}}$ High	433		$7t_{\text{CLCL}}-120$		ns
$t_{\text{WHQX}}$	Data Hold After $\overline{\text{WR}}$	33		$t_{\text{CLCL}}-20$		ns
$t_{\text{RLAZ}}$	$\overline{\text{RD}}$ Low to Address Float		0		0	ns
$t_{\text{WHLH}}$	$\overline{\text{RD}}$ or $\overline{\text{WR}}$ High to ALE High	43	123	$t_{\text{CLCL}}-20$	$t_{\text{CLCL}}+25$	ns

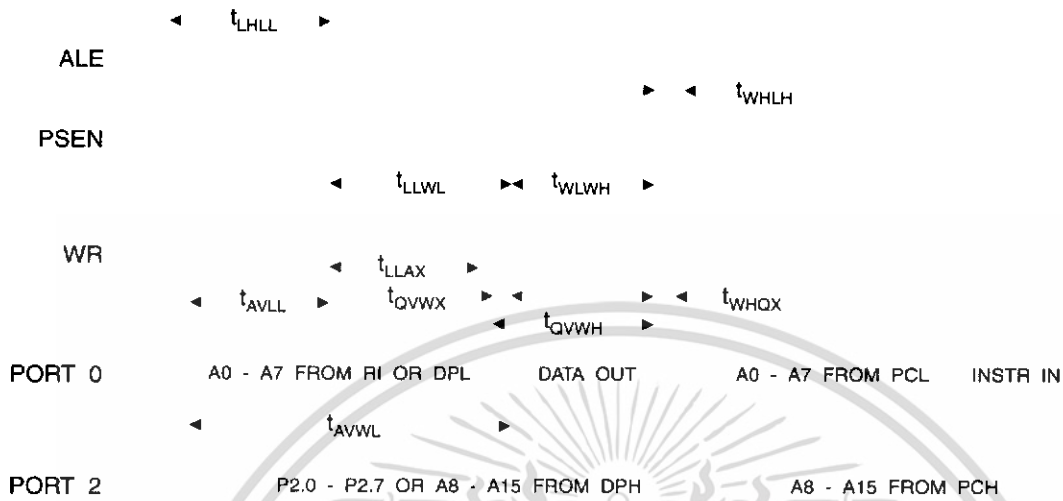
### External Program Memory Read Cycle



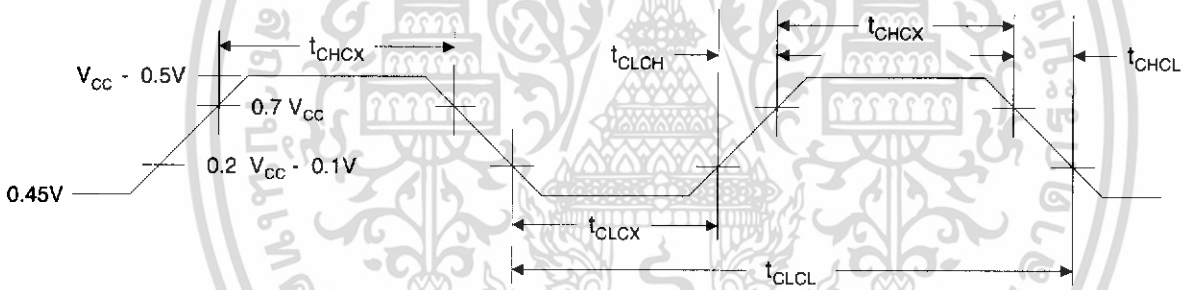
### External Data Memory Read Cycle



**External Data Memory Write Cycle**



**External Clock Drive Waveforms**



**External Clock Drive**

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



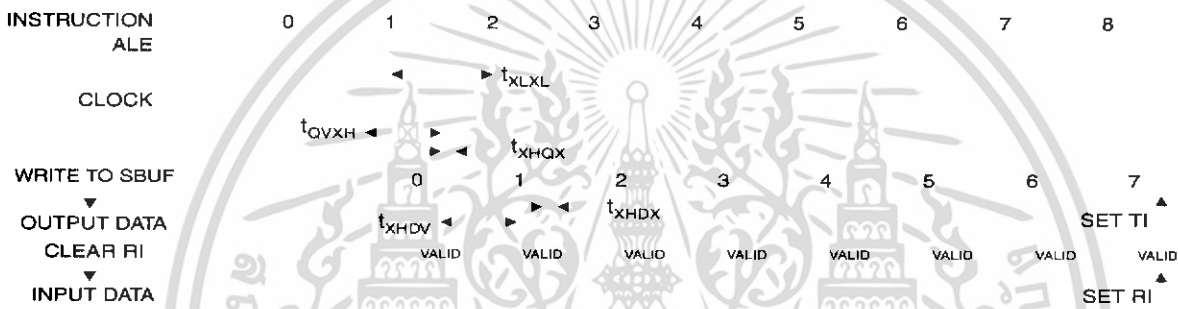


## Serial Port Timing: Shift Register Mode Test Conditions

( $V_{CC} = 5.0\text{ V} \pm 20\%$ ; Load Capacitance = 80 pF)

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

## Shift Register Mode Timing Waveforms



## AC Testing Input/Output Waveforms (1)

## Float Waveforms (1)



Note: 1. AC Inputs during testing are driven at  $V_{CC} - 0.5\text{V}$  for a logic 1 and  $0.45\text{V}$  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 mV change from load voltage occurs. A port pin begins to float when 100 mV change from the loaded  $V_{OH}/V_{OL}$  level occurs.

## Ordering Information

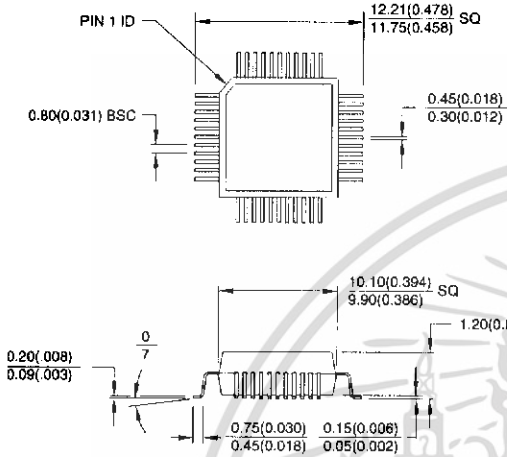
Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
12	5V ± 20%	AT89C51-12AC	44A	Commercial (0°C to 70°C)
		AT89C51-12JC	44J	
		AT89C51-12PC	40P6	
		AT89C51-12QC	44Q	
	5V ± 20%	AT89C51-12AI	44A	Industrial (-40°C to 85°C)
		AT89C51-12JI	44J	
		AT89C51-12PI	40P6	
		AT89C51-12QI	44Q	
16	5V ± 20%	AT89C51-16AC	44A	Commercial (0°C to 70°C)
		AT89C51-16JC	44J	
		AT89C51-16PC	40P6	
		AT89C51-16QC	44Q	
	5V ± 20%	AT89C51-16AI	44A	Industrial (-40°C to 85°C)
		AT89C51-16JI	44J	
		AT89C51-16PI	40P6	
		AT89C51-16QI	44Q	
20	5V ± 20%	AT89C51-20AC	44A	Commercial (0°C to 70°C)
		AT89C51-20JC	44J	
		AT89C51-20PC	40P6	
		AT89C51-20QC	44Q	
	5V ± 20%	AT89C51-20AI	44A	Industrial (-40°C to 85°C)
		AT89C51-20JI	44J	
		AT89C51-20PI	40P6	
		AT89C51-20QI	44Q	
24	5V ± 20%	AT89C51-24AC	44A	Commercial (0°C to 70°C)
		AT89C51-24JC	44J	
		AT89C51-24PC	40P6	
		AT89C51-24QC	44Q	
	5V ± 20%	AT89C51-24AI	44A	Industrial (-40°C to 85°C)
		AT89C51-24JI	44J	
		AT89C51-24PI	40P6	
		AT89C51-24QI	44Q	

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

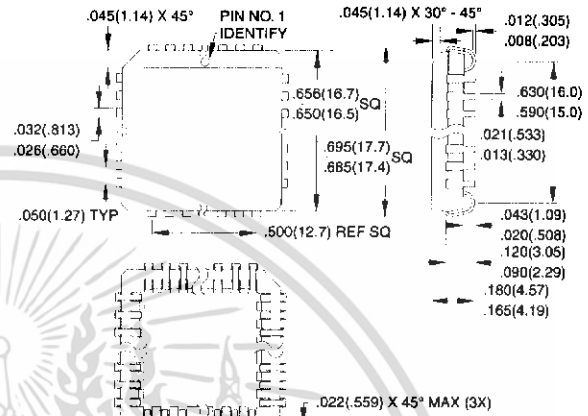


## Packaging Information

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

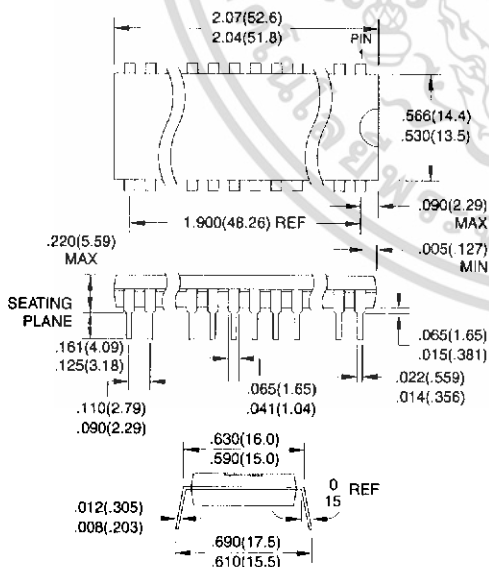


**44J**, 44-lead, Plastic J-leaded Chip Carrier (PLCC)  
 Dimensions in Inches and (Millimeters)  
 JEDEC STANDARD MS-018 AC

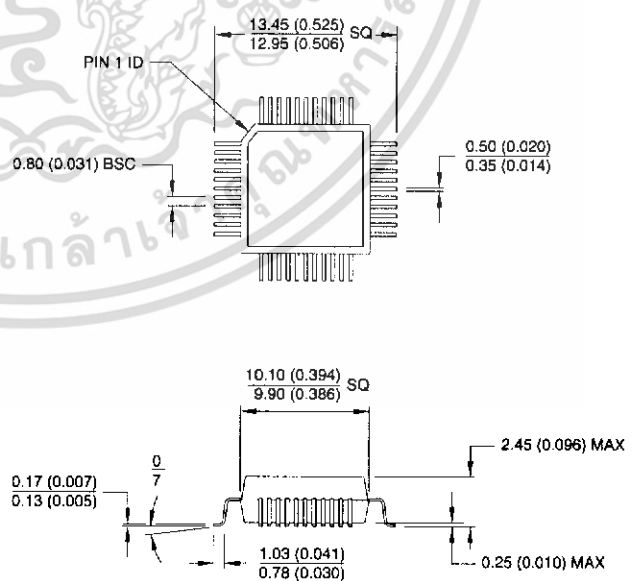


Controlling dimension: millimeters

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



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



Controlling dimension: millimeters



## Atmel Headquarters

### Corporate Headquarters

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

### Europe

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

### Asia

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

### Japan

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

## Atmel Operations

### Atmel Colorado Springs

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

### Atmel Rousset

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

### Fax-on-Demand

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

*e-mail*  
literature@atmel.com

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

*BBS*  
1-(408) 436-4309

### © Atmel Corporation 2000.

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

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

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

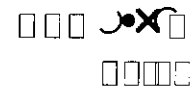
Printed on recycled paper.

0285G-02/00/xM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

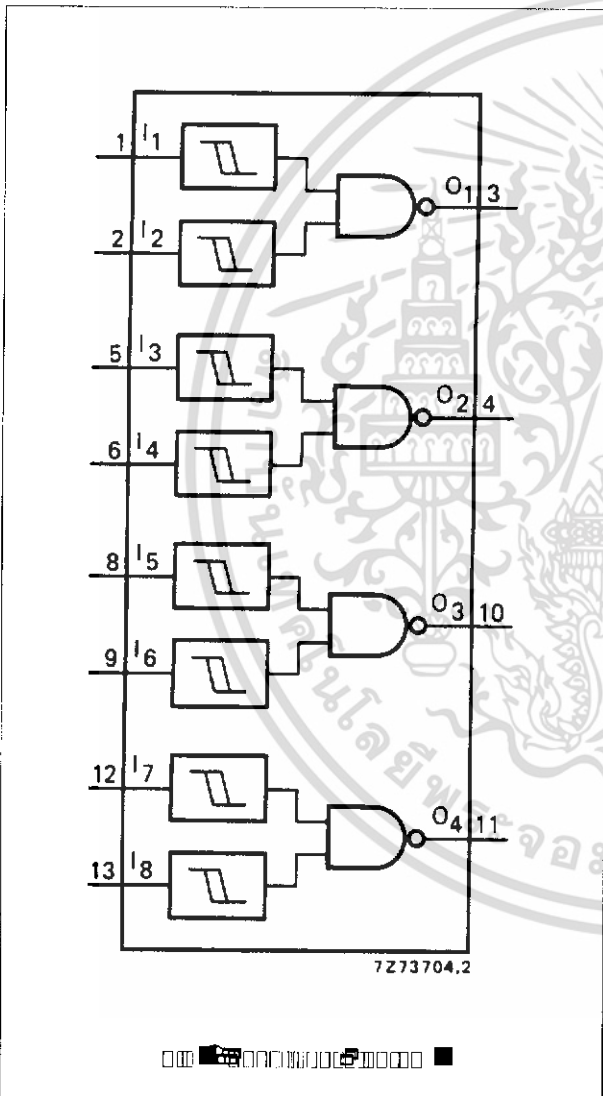
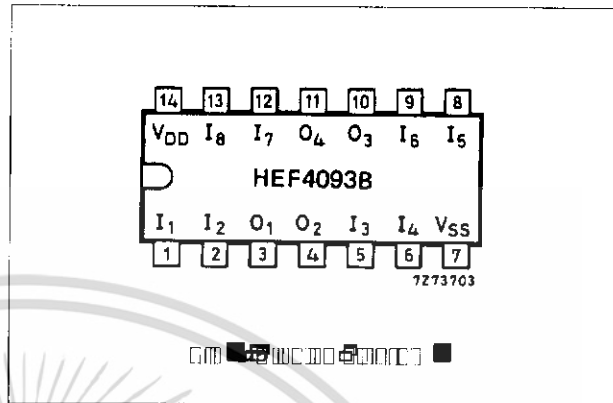


วัตถุประสงค์ของงานวิจัย

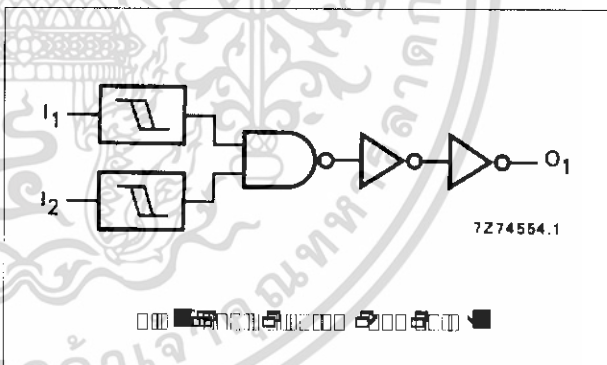


วัตถุประสงค์ของงานวิจัย

วัตถุประสงค์ของงานวิจัยนี้คือ เพื่อศึกษาและเปรียบเทียบการทำงานของวงจรลอจิกที่ประกอบด้วยอินเวอร์เตอร์และเกต AND ที่มีจำนวนขาเข้าที่แตกต่างกัน โดยจะทำการวัดค่าความถี่ของสัญญาณขาออกและเปรียบเทียบกับค่าที่คำนวณได้จากตารางความจริงของวงจร



วัตถุประสงค์ของงานวิจัยนี้คือ เพื่อศึกษาและเปรียบเทียบการทำงานของวงจรลอจิกที่ประกอบด้วยอินเวอร์เตอร์และเกต AND ที่มีจำนวนขาเข้าที่แตกต่างกัน โดยจะทำการวัดค่าความถี่ของสัญญาณขาออกและเปรียบเทียบกับค่าที่คำนวณได้จากตารางความจริงของวงจร



วัตถุประสงค์ของงานวิจัยนี้คือ เพื่อศึกษาและเปรียบเทียบการทำงานของวงจรลอจิกที่ประกอบด้วยอินเวอร์เตอร์และเกต AND ที่มีจำนวนขาเข้าที่แตกต่างกัน โดยจะทำการวัดค่าความถี่ของสัญญาณขาออกและเปรียบเทียบกับค่าที่คำนวณได้จากตารางความจริงของวงจร

วัตถุประสงค์ของงานวิจัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้







