

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

หุ่นยนต์เตะฟุตบอล

ROBOT SOCCER



เลขามู.....
เลขทะเบียน..... 72975
วัน,เดือน,ปี..... 26 ส.ย. 2550

b. 1123288
i.

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หุ่นยนต์เตะฟุตบอล

ROBOT SOCCER



ปฏิญานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโท ปีการศึกษา 2549

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง หุ่นยนต์เตะฟุตบอล

ผู้จัดทำ

นายอศุทธิ์เดช นิ่มนวล เลขประจำตัว 47015232



รศ.ดร. ชูชาติ ปิณฑวิรุจน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หุ่นยนต์เตะฟุตบอล

นายอดุลย์เดช นิ่มนวล รหัส 47015232
รศ.ดร.ชูชาติ ปิณฑวิรุจน์ อาจารย์ที่ปรึกษา
ปีการศึกษา 2549

บทคัดย่อ

รายงานฉบับนี้ อธิบายการออกแบบและสร้าง หุ่นยนต์เตะฟุตบอล โดยใช้หลักการประมวลผลด้วยสัญญาณภาพ เพื่อนำมาควบคุมการทำงานของหุ่นยนต์ ขั้นตอนการทำงาน หุ่นยนต์นั้นจะอยู่ในสนามที่กำหนดหุ่นยนต์จะเคลื่อนที่เข้าหาลูกบอล และ ยิงบอลเข้าประตู การควบคุมหุ่นยนต์จะใช้กล้องจากทางด้านบนส่งภาพมาให้คอมพิวเตอร์ประมวลผล และ ทำการประมวลผลที่ได้ด้วยโปรแกรม และควบคุมหุ่นยนต์โดยเครื่องส่งไร้สาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ROBOT SOCCER

Aduldedch Nimnual ID 47015232

Assoc. Prof. Dr. Chuchart Pintavirooj Advisor

Educational Year 2006

Abstract

This report introduces a design and a construction of robot soccer using image processing for controlling the robot movement. The robot can automatically move in a definite field. It will go to the ball and shoot the ball to goal. The concept of control robot is transfer image from digital camera that location above the field to computer and process by program and sends data to control robot by wireless transmitter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

หลังจากทำโครงการนี้ทำให้ผู้จัดทำได้รับความรู้มากมายหลายด้าน ทั้งด้าน Image Processing การเขียนโปรแกรมด้วย ++Builder6 การใช้งานไมโครคอนโทรลเลอร์ การรับส่งข้อมูลแบบไร้สาย และการออกแบบวงจรที่ต้องใช้งานจริง ซึ่งความรู้ดังกล่าวมาจากการศึกษาค้นคว้าด้วยตัวเองและคำแนะนำจากผู้อุปการะหลายท่าน โดยเฉพาะอาจารย์ ชูชาติ ปิณฑวิรุจน์ ที่ให้คำแนะนำโดยตลอด

ขอบพระคุณ คุณพ่อ และ คุณแม่ ที่ให้การสนับสนุนด้านการศึกษา การเงินและกำลังใจ ความห่วงใย มาโดยตลอด จนทำให้การทำงานในครั้งนี้สำเร็จลุล่วงไปได้ดี

ขอบคุณภรรยาอันเป็นที่รักที่คอยให้กำลังใจและอยู่ข้างๆ ทั้งในยามที่ทุกข์ยากลำบาก ไม่หนีไปไหนถึงแม้จะประสบปัญหาในหลายด้าน

ขอบคุณพี่ๆ ป.โท ที่ให้คำแนะนำ เพื่อนๆ น้องๆ ที่ทำโปรเจกต์เรื่อง Image เหมือนกัน ที่ให้ความช่วยเหลือไม่ว่าจะต้องการอะไรทุกคนก็เต็มใจช่วย มาโดยตลอด

.....
นายอตุลย์เดช นิ่มนวล

สารบัญ

	หน้า
บทคัดย่อ	I
Abstract	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูปภาพ	VI
สารบัญตาราง	VIII
บทที่ 1 บทนำ	1
บทที่ 2 หลักการทำงานของการทำงานควบคุมหุ่นยนต์เตะฟุตบอล	2
2.1 กล้อง VDO	3
2.2 คอมพิวเตอร์และ โปรแกรม	3
2.3 พอร์ตขนาน (Parallel port)	3
2.4 อุปกรณ์ส่งรับสัญญาณไร้สาย	4
2.4.1 ภาคส่งไร้สาย	5
2.4.2 ภาครับแบบไร้สาย	9
2.5 ไมโครคอนโทรลเลอร์ และ Drive Motor	13
2.5.1 โครงสร้างภายในของไมโครคอนโทรลเลอร์ MCS-51	13
2.5.2 โครงสร้างหน่วยความจำภายในไมโครคอนโทรลเลอร์ MCS-51	15
2.5.3 หน่วยความจำโปรแกรม	15
2.5.4 หน่วยความจำข้อมูล	17
2.5.5 รีจิสเตอร์ที่เกี่ยวกับไมโครคอนโทรลเลอร์ MCS-51	19
2.5.6 ชุดคำสั่งของไมโครคอนโทรลเลอร์ MCS-51	21
2.5.7 โครงสร้างการอินเตอร์รัปต์ของไมโครคอนโทรลเลอร์ MCS-51	22
2.5.8 การรีเซต	23
2.6 การใช้งานไมโครคอนโทรลเลอร์กับพอร์ตอนุกรม	23
2.6.1 หน้าที่ของสัญญาณต่าง ๆ	26
2.6.2 Null Modems	27
2.6.3 Flow Control	28
2.6.4 รูปคลื่น สัญญาณ RS-232	29
2.6.5 ตัวแปลงสัญญาณ RS-232	29
2.7 การสื่อสารข้อมูล	31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.1 ประเภทของการสื่อสารข้อมูล	31
2.8 ดิซีมอเตอร์ (DC Motor)	36
2.8.1 หลักการทำงานของดิซีมอเตอร์	36
2.8.2 ดิซีมอเตอร์แบบเส้นแรงแม่เหล็กคงที่	37
2.9 วงจร Drive Motor	38
บทที่ 3 หลักการประมวลผลภาพดิจิทัล	41
3.1 รูปภาพดิจิทัล	41
3.2 การปรับปรุงภาพ	41
3.3 พิกเซล (Pixel)	46
3.4 รูปแบบข้อมูล	46
3.5 เกรย์สเกล	46
บทที่ 4 ผลการทดลอง	48
ผลการทดลองที่ 1	49
ผลการทดลองที่ 2	50
ผลการทดลองที่ 3	51
ผลการทดลองที่ 4	51
บทที่ 5 ปัญหาและวิธีการแก้ปัญหา	56
บทที่ 6 สรุปและวิจารณ์ผลการทดลอง	58

สารบัญรูปภาพ

	หน้า
รูปที่ 1.1 แสดงรูปร่างของโครงการ	1
รูปที่ 2.1 Block diagram การทำงานของหุ่นยนต์เตะฟุตบอล	2
รูปที่ 2.2 คอนเนคเตอร์ DB25 Parallel port	4
รูปที่ 2.3 แสดงตัวส่งสัญญาณ Transmitter	4
รูปที่ 2.4 แสดงตัวรับสัญญาณ Receiver	5
รูปที่ 2.5 แสดงการทำงานของภาคส่งข้อมูลไร้สาย	5
รูปที่ 2.6 แสดง Pin ของ ICเบอร์ HT12E	6
รูปที่ 2.7 กราฟแสดงความสัมพันธ์ระหว่าง VDDและค่าความต้านทาน	7
รูปที่ 2.8 Block Diagram โครงสร้างภายในของ Encoder HT12E	8
รูปที่ 2.9 ลักษณะของ Function Code ที่ใช้ในการ Encoding ของ HT12E	8
รูปที่ 2.10 Block Diagram แสดงการทำงานของภาครับไร้สาย RF	9
รูปที่ 2.11 Block Diagram การทำงานของ HT12D	10
รูปที่ 2.12 แสดง Pin ของ ICเบอร์ HT12D	10
รูปที่ 2.13 แสดงการทำงานของ Decoder timing	11
รูปที่ 2.14 กราฟแสดงความสัมพันธ์ระหว่าง VDDและค่าความต้านทาน	12
รูปที่ 2.15 Diagram การเชื่อมต่อ AT89C1051 กับ L293D	13
รูปที่ 2.16 แสดงโครงสร้างภายในของไมโครคอนโทรลเลอร์ MCS-51	14
รูปที่ 2.17 แสดงรูปร่างและการจัดวางขาต่างๆ ของไมโครคอนโทรลเลอร์ MCS-51	15
รูปที่ 2.18 แสดงการใช้หน่วยความจำสำหรับเก็บโปรแกรม	16
รูปที่ 2.19 แสดงการจัดพื้นที่ของหน่วยความจำโปรแกรมภายในและภายนอก	16
รูปที่ 2.20 แสดงหน่วยความจำสำหรับเก็บข้อมูลภายในไมโครคอนโทรลเลอร์ MCS-51	17
รูปที่ 2.21 แสดงการจัดหน่วยความจำข้อมูล	18
รูปที่ 2.22 แสดงการต่อกับหน่วยความจำข้อมูลภายนอกไอซี	19
รูปที่ 2.23 แสดงการเลือกใช้รีจิสเตอร์ใช้งานทั่วไป R0-R7 ในแต่ละกลุ่ม	20
รูปที่ 2.24 ตารางแสดงถึงหน้าที่ของขาต่าง ๆ	25
รูปที่ 2.25 แสดงสัญญาณต่างๆที่ส่งในรูปแบบอนุกรม	26
รูปที่ 2.26 แสดงการเชื่อมต่อของสายสัญญาณ	28
รูปที่ 2.27 รูปคลื่นของสัญญาณที่ส่ง	29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.28 แสดงโครงสร้างภายในและตำแหน่งขาต่างๆของ Max232	30
รูปที่ 2.29 แสดงการรับส่งข้อมูลแบบทางเดียว (Simplex)	32
รูปที่ 2.30 แสดงการรับส่งข้อมูลสวนทางกันได้แบบผลัดการส่ง(Half Duplex)	32
รูปที่ 2.31 แสดงการรับส่งข้อมูลแบบสวนทางกันได้พร้อมกัน (Full Duplex)	33
รูปที่ 2.32 แสดงการส่งข้อมูลแบบขนาน	34
รูปที่ 2.33 แสดงการส่งข้อมูลแบบอนุกรม	34
รูปที่ 2.34 แสดงการส่งข้อมูลแบบสัมพันธ์	35
รูปที่ 2.35 แสดงการส่งข้อมูลแบบ ไม่สัมพันธ์	36
รูปที่ 2.36 แสดงลักษณะ Pin ของ IC เบอร์ TA7279P	38
รูปที่ 2.38 แสดงการทำงานของ IC เบอร์ TA7279P กับมอเตอร์ชนิด DC	39
รูปที่ 3.1 แสดงการใช้หน้าต่างกรอบภาพที่รับเข้ามา	42
รูปที่ 3.2 แสดงภาพต้นฉบับและ histogram ที่ไม่ถูก normalized	43
รูปที่ 3.3 ภาพต้นฉบับ	43
รูปที่ 3.4 ผ่านการทำ histogram equalization	44
รูปที่ 3.5 แสดง histogram ของภาพที่และภาพที่	44
รูปที่ 3.6 แสดงการทำ averaging filters	45
รูปที่ 3.7 แสดงรูปต้นฉบับและรูปที่ผ่าน averaging filters	45
รูปที่ 3.8 แสดงภาพต้นฉบับและภาพหลังจากการทำ median filters	45
รูปที่ 4.1 ภาดส่งในการใช้งานจริง	48
รูปที่ 4.2 เป็นการแสดงการหาค่าความต้านทาน	48
รูปที่ 4.3 ชุดความคุมRobot หิ้งหมดเมื่อใช้งานจริง	49
รูปที่ 4.4 เป็นการแสดงการหาค่าความต้านทาน	50

สารบัญตาราง

	หน้า
ตารางที่ 2.1 คำอธิบายขาสัญญาณของพอร์ตนาน	4
ตารางที่ 2.2 ตารางแสดงคำอธิบายของ IC เบอร์ HT12E	6
ตารางที่ 2.3 ตารางแสดงคำอธิบายของ Pin IC เบอร์ HT12D	11
ตารางที่ 2.4 ตารางแสดงคำอธิบายของ Pin IC เบอร์ TA9297P	39
ตารางที่ 4.1 แสดงการทำงานของหุ่นยนต์แบบ Binary	51
ตารางที่ 4.2 แสดงคำสั่งต่างๆที่สั่งให้หุ่นยนต์ทำงาน	51



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

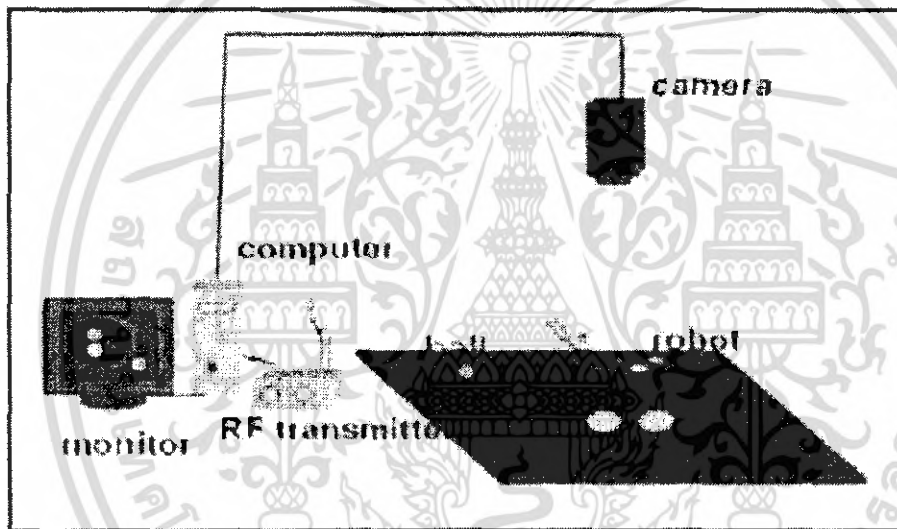
บทที่ 1

บทนำ

โครงการนี้ผู้พัฒนาหุ่นยนต์เตะฟุตบอล ได้สร้างให้หุ่นยนต์นี้จะมีคุณสมบัติดังนี้

1. หุ่นยนต์จะถูกควบคุมการเคลื่อนไหวด้วยระบบคอมพิวเตอร์ที่อยู่ระยะไกล โดยการส่งสัญญาณควบคุมจะส่งผ่านระบบความถี่วิทยุ

2. การบอกตำแหน่งการเคลื่อนที่ของหุ่นยนต์และลูกฟุตบอลจะต้องอาศัยตำแหน่งอ้างอิงจากกล้องVDO ซึ่งกล้องจะถูกติดตั้งอยู่ทางด้านบนของสนามที่ถูกกำหนดให้รถเคลื่อนที่



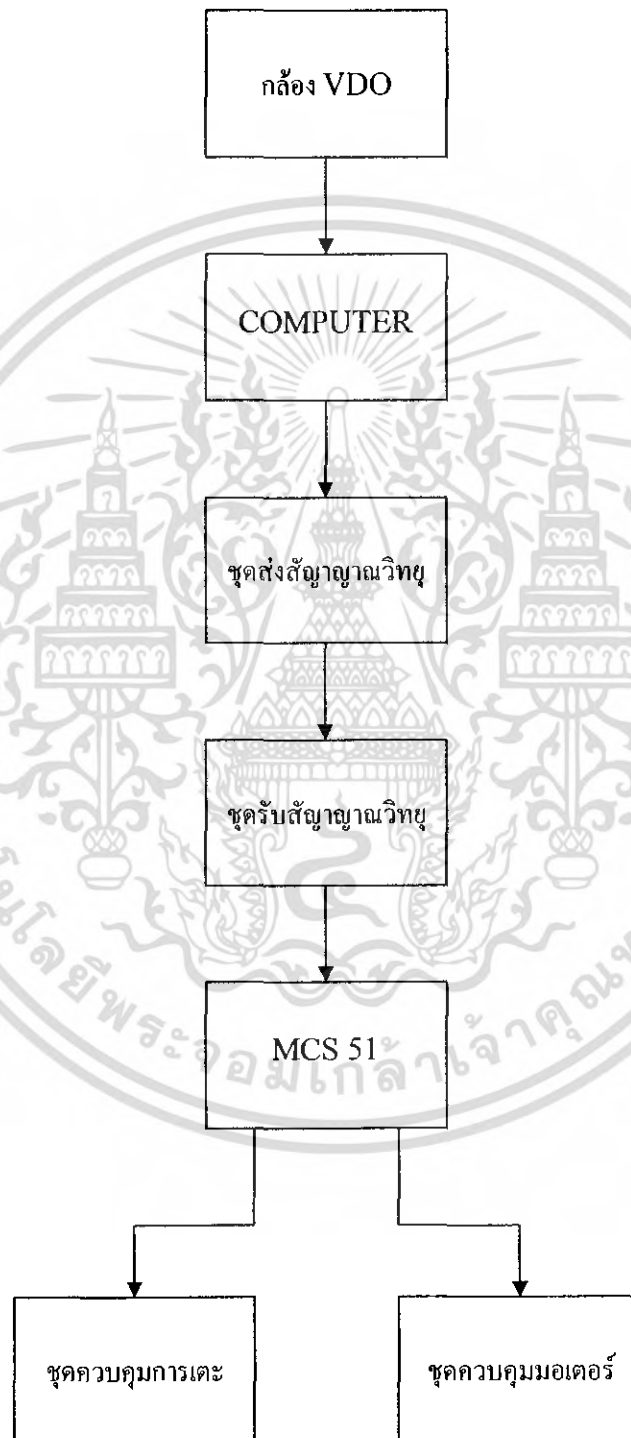
รูปที่ 1.1 แสดงรูปร่างของ โครงการ

ในการควบคุมหุ่นยนต์ด้วยคอมพิวเตอร์นั้น เกิดจากการรับภาพจากกล้อง ที่ถ่ายมุมสูงบริเวณสนามตามรูปที่ 1.1 คอมพิวเตอร์จะประมวลผลภาพด้วยหลักการของอิมเมจโปรเซสซิ่ง โดยใช้ภาษา c++builder ในการเขียนโปรแกรมเพื่อประมวลผลอิมเมจโปรเซสซิ่ง คอมพิวเตอร์จะสามารถพบตำแหน่งของ หุ่นยนต์และลูกบอลแล้วหาวิธีเดินทางไปที่ลูกฟุตบอล และ จะส่งข้อมูลเพื่อให้หุ่นยนต์ผ่านทางความถี่วิทยุเพื่อควบคุมให้หุ่นยนต์เดินทางไปที่ลูกบอล และ เตะฟุตบอลได้ถูกต้อง ตามตำแหน่งที่วิเคราะห์ได้จากกล้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

หลักการทํางานของการควบคุมหุ่นยนต์เตะฟุตบอล



รูปที่ 2.1 Block diagram การทํางานของหุ่นยนต์เตะฟุตบอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1 กล้อง VDO

กล้อง VDO ที่ใช้ในโครงการนี้ เป็นกล้อง มีคุณสมบัติดังนี้

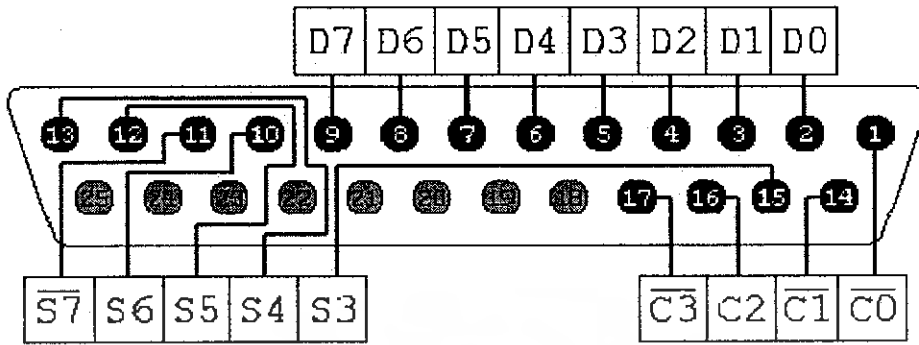
1.ขนาดภาพ	354 x 298 Pixels
2. Interface	LifeView Card
3. Frame rate	100 Frame/sec
4. เลนส์โฟกัส	4mm
5.แรงดัน	12V

2.2 คอมพิวเตอร์และโปรแกรม

คอมพิวเตอร์จะเชื่อมต่อกับกล้องและตัวส่งสัญญาณวิทยุเมื่อคอมพิวเตอร์รับข้อมูลจากกล้อง VDO นำภาพมาทที่ได้แสดงผลที่จอคอมพิวเตอร์ และ ในการควบคุมหุ่นยนต์ให้เคลื่อนที่ผ่านทางพอร์ตขนาน โดยใช้ C++Builder ในการควบคุมการทำงานทั้งหมดของคอมพิวเตอร์

2.3 พอร์ตขนาน (Parallel port)

Parallel Port ก็คือ ช่องทาง ที่ CPU ใช้ติดต่อกับ อุปกรณ์พื้นฐานต่าง ๆ โดยทั่วไปแล้ว Port จะสามารถส่งและ รับข้อมูลได้ หรืออาจจะ ส่ง หรือ รับ อย่างเดียว พอร์ตจะมีหมายเลขประจำตัวสำหรับ เพื่อที่ว่า CPU และ Software รวมทั้ง Programmer ที่เขียน โปรแกรม จะได้ระบุถึง อุปกรณ์ที่จะติดต่อได้ Printer Port หรือ Com Port มันคือ พอร์ตภายนอก หรือ External Port มีหลายชนิดเยอะแยะมากมาย เช่น USB Fire wire SCSI PS2 KEYBOARD พวกนี้ External Port ทั้งหมด เพราะอยู่นอกเครื่อง แต่ส่วนใหญ่พอร์ต พวกนี้ต้องส่งข้อมูลผ่าน Hardware Port หรือควบคุมผ่านทาง Hardware Port โดยลึก ๆ แล้ว Parallel Port จะทำงานภายใต้ BUS ซึ่งเป็นช่องทางที่ CPU จะติดต่อกับ อุปกรณ์ทุกอย่าง ภายในเครื่อง การรับส่งของ Parallel Port จะทำงานโดยการสั่งงานจาก CPU ดังนั้น การเขียน ก็คือ CPU ต้องการส่งข้อมูลออกไปสู่ Parallel Port และ การอ่าน คือ CPU ต้องการรับข้อมูลจาก Parallel Port พอร์ตขนาน เป็นช่องทางในการติดต่อสื่อสารระหว่างตัวคอมพิวเตอร์กับอุปกรณ์ภายนอก มีการรับส่งข้อมูลมากกว่า 1 Bit ต่อครั้ง พอร์ตขนานถูกพัฒนาขึ้นเพื่อนแก้ปัญหาอัตราการส่งข้อมูลของคอมพิวเตอร์



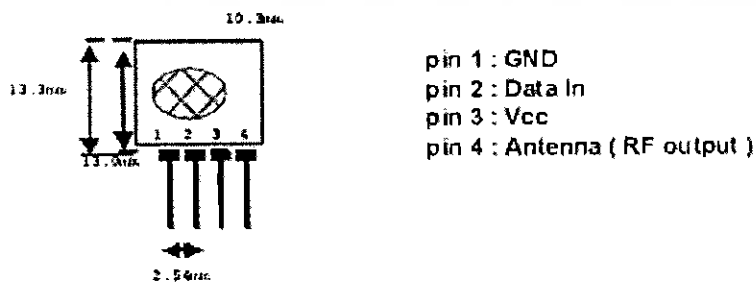
รูปที่ 2.2 คอนเนคเตอร์ DB25 Parallel port

ตารางที่ 2.1 คำอธิบายขาสัญญาณของพอร์ตขนาน

ขาสัญญาณ	คำอธิบาย	ขาสัญญาณ	คำอธิบาย
1	ขา Strobe (Control Bit0)	10	ขา ACK (Status Bit6)
2	ขา Data 0 (Data Bit0)	11	ขา Busy (Status Bit7)
3	ขา Data 1 (Data Bit1)	12	ขา Paper Empty (Status Bit5)
4	ขา Data 2 (Data Bit2)	13	ขา Select(Status Bit4)
5	ขา Data 3 (Data Bit3)	14	ขา Auto Feed(Control Bit1)
6	ขา Data 4 (Data Bit4)	15	ขา Error (Status Bit3)
7	ขา Data 5 (Data Bit5)	16	ขา Initialize Printer(Control Bit2)
8	ขา Data 6 (Data Bit6)	17	ขา Select Input(Control Bit3)
9	ขา Data 7 (Data Bit7)	18-25	ขากราวด์(Ground Signal)

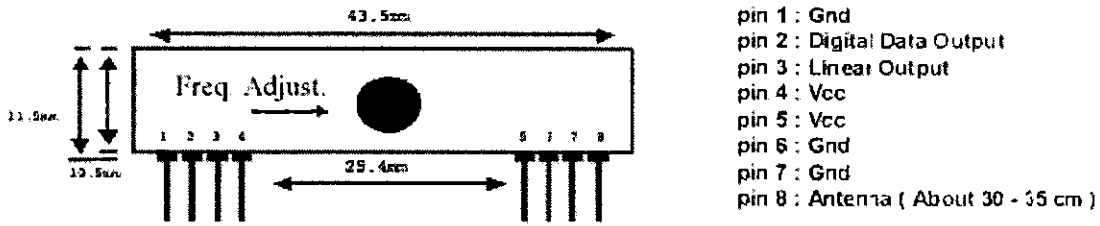
2.4 อุปกรณ์ส่งรับสัญญาณไร้สาย

ในโครงการนี้คือ TLP434A และ RLP434 Wireless ความถี่ 433.92MHz



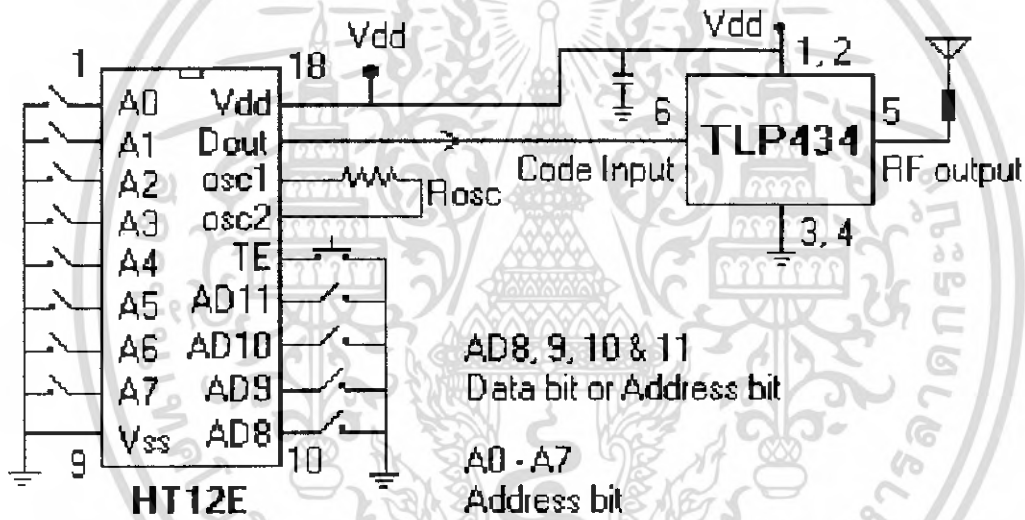
รูปที่ 2.3 แสดงตัวส่งสัญญาณ Transmitter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



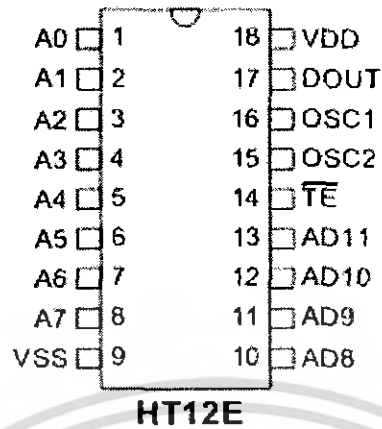
รูปที่ 2.4 แสดงตัวรับสัญญาณ Receiver

2.4.1 ภาคส่งไร้สาย



รูปที่ 2.5 แสดงการทำงานของภาคส่งข้อมูล ไร้สาย

จากรูปที่ 2.5 เป็นตัว เป็นการทำงานของภาคส่งสัญญาณ ไร้สาย จาก AD8 – AD11 จะทำหน้าที่รับข้อมูลขนาด 4 บิต จาก port printer แล้ว HT12E จะทำหน้าที่เข้ารหัส เป็น Encoder เพื่อส่งข้อมูลเป็น Digital Code ข้อมูลที่ถูกเข้ารหัสจะถูกเปลี่ยนเป็นความถี่เสียงแล้วจะถูกส่งให้ Wireless เพื่อ Mod ความถี่เสียงที่เป็นข้อมูล Digital Code เข้ากับความถี่วิทยุเพื่อส่งออกไปยังภาครับของ Robot เพื่อควบคุมให้ Robot ทำงานตามที่ต้องการ



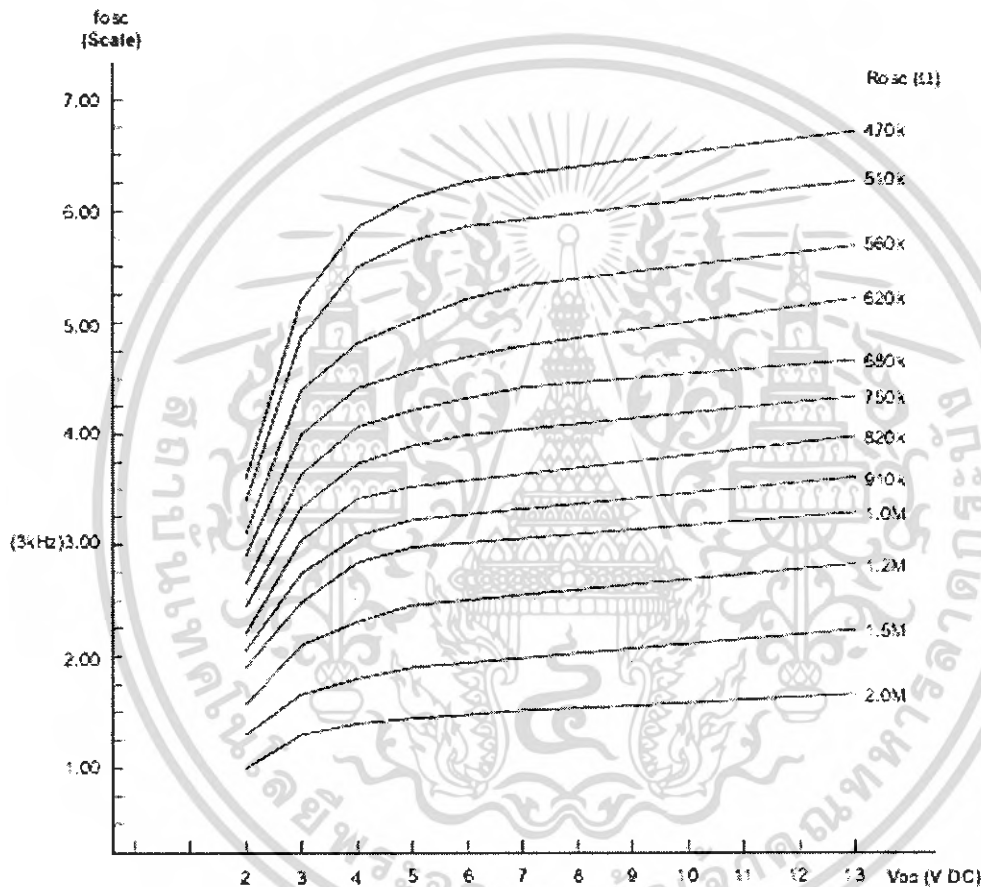
รูปที่ 2.6 แสดง Pin ของ IC เบอร์ HT12E

ตารางที่ 2.2 ตารางแสดงคำอธิบายของ IC เบอร์ HT12E

Pin Name	I/O	Internal Connection	Description
A0 ~ A7	I	NMOS TRANSMISSION GATE PROTECTION DIODE	Input pins for address A0-A7 setting. These pins can be externally set to VSS or left open
AD8 ~ AD11	I	NMOS TRANSMISSION GATE PROTECTION DIODE	Input pins for address/data AD8 – AD11 setting. These pins can be externally set to VSS or left open
DOUT	O	CMOS OUT	Encode data serial transmission out put
TE	I	CMOS IN PULL HIGH	Transmission enable , active low
OSC1	I	OSCILLATOR 1	Oscillator input pin
OSC1	O	OSCILLATOR 2	Oscillator out pin
VSS	I	-	Ground
VDD	I	-	Positive Power Supply

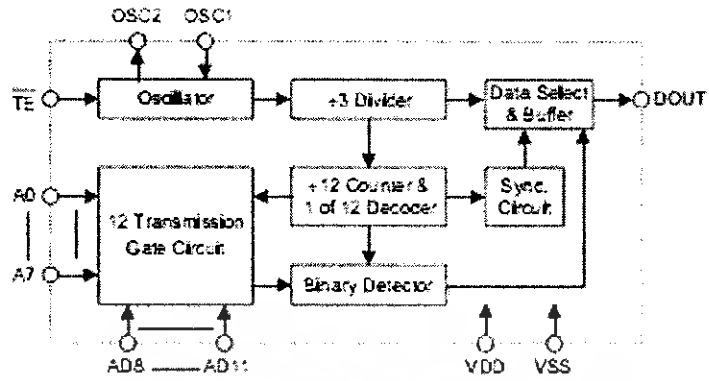
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางที่ 2.2 เราสามารถเห็นหน้าที่การทำงานของขา IC แต่ละขาได้ ซึ่งแต่ละขาจะทำหน้าที่ต่างกันไปดังตารางที่แสดงให้ดู โดยที่ขา OSC1 และ ขา OSC2 ต้องต่อค่าความต้านทานที่เหมาะสมเพื่อกำหนดสัญญาณความถี่ ที่เราต้องการได้ถูกต้อง ตรงกับภาครับในชุดรับสัญญาณ ที่ได้ออกแบบไว้เพื่อให้วงจรสามารถถอดรหัส ได้ถ้าใส่ตัวต้านทานที่ไม่เหมาะสมภาครับ จะไม่สามารถ ถอดรหัสได้ถูกต้อง ความสัมพันธ์ของความถี่ที่ผลิตออกมาจะขึ้นอยู่กับค่า VDD ที่เราป้อนและค่าความต้านทานที่เราใส่เป็นสำคัญ สามารถดูได้จากกราฟ



รูปที่ 2.7 การแสดงความสัมพันธ์ระหว่าง VDD และค่าความต้านทาน

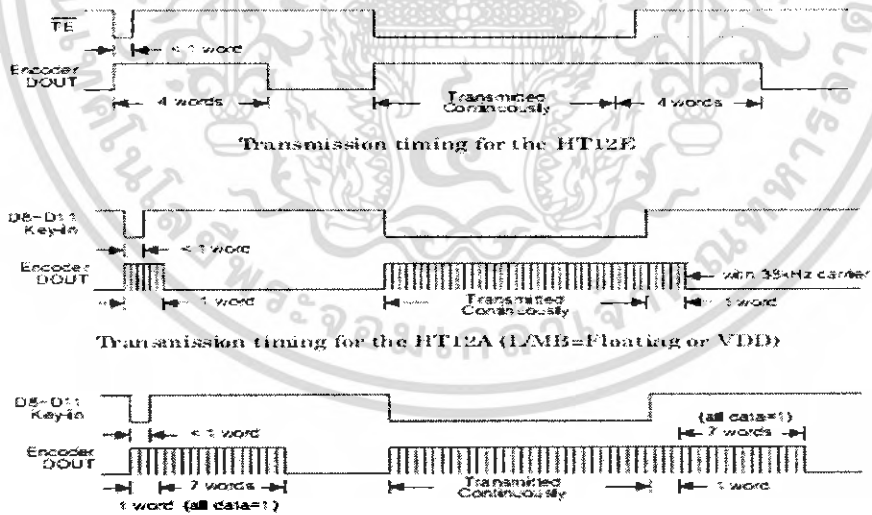
จากกราฟ รูปที่ 2.7 สามารถหาความถี่ที่ต้องการได้จากการอ่านกราฟ โดยต้องรู้ค่า VDD และความถี่ที่ต้องการ แล้วจึงหาจุดตัดกราฟเพื่อให้ได้ค่า ความต้านทานที่ต้องการเพื่อไปใส่ในวงจรที่ได้ออกแบบไว้ได้อย่างถูกต้อง



รูปที่ 2.8 Block Diagram โครงสร้างภายในของ Encoder HT12E

จากรูปที่ 2.8 ภายใน IC จะประกอบไปด้วยตัวกำเนิดสัญญาณนาฬิกาเพื่อใช้ในการกำหนด Digital Code ของคำสั่งต่างๆที่เข้ามาทาง AD8 – AD11 เพื่อทำการ Encoding แล้วส่งออกไปที่ Out Put ของ HT12E สัญญาณที่ออกมาจะเป็นความถี่เสียงไม่มี คลื่นที่เป็นพาหะออกมา

ลักษณะการเข้ารหัสของ Encoding



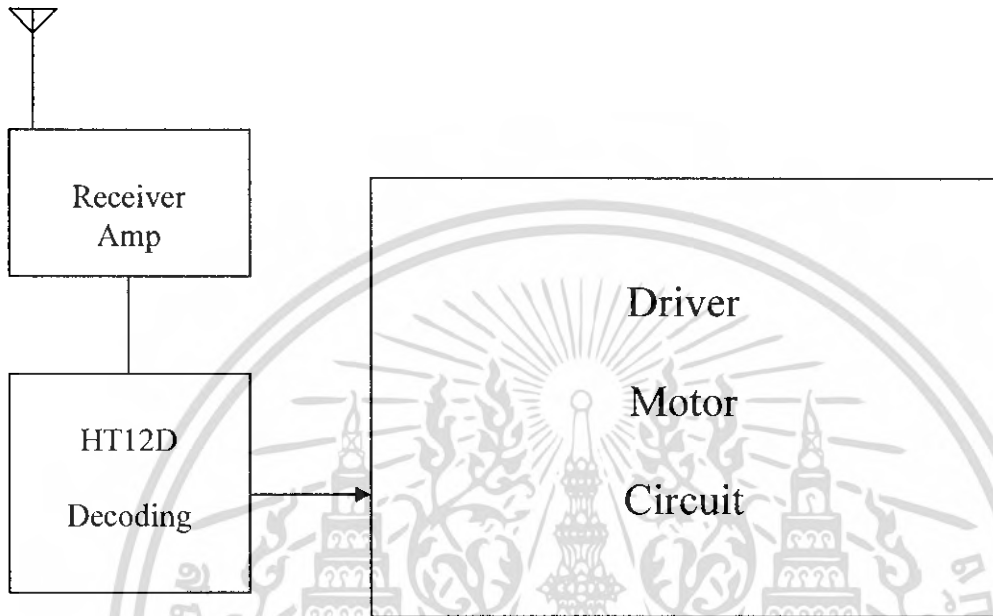
รูปที่ 2.9 ลักษณะของ Function Code ที่ใช้ในการ Encoding ของ HT12E

2.4.2 ภาครับแบบไร้สาย

เมื่อข้อมูลถูกส่งออกมาจากตัวส่งซึ่งเป็น ข้อมูลที่ใช้ในการควบคุมการทำงานของ Robot จำเป็นอย่างยิ่งที่ต้องมีตัวรับสัญญาณ และ กำจัดคลื่นพาหะออกเพื่อให้ได้ สัญญาณ Digital Code ที่

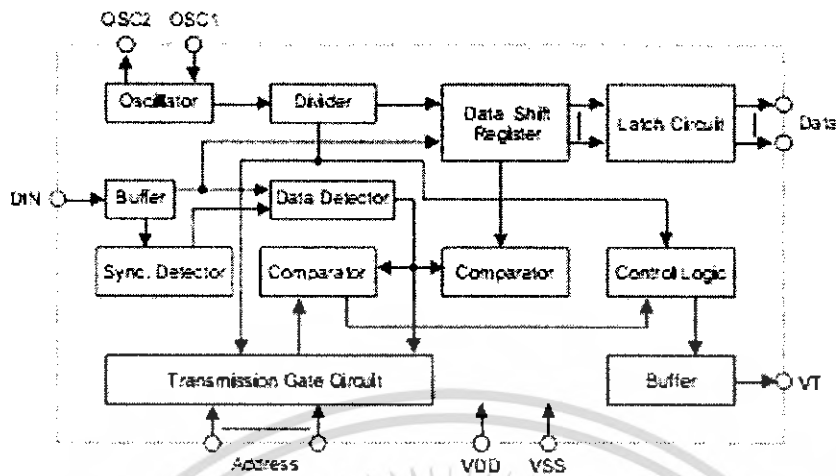
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นความถี่เสียง แล้วนำสัญญาณนั้นมาขยายแล้วเข้าสู่กระบวนการถอดรหัส หรือ Decoding นำคำสั่งที่ถูกกำหนดมาควบคุมมอเตอร์ให้หมุนตามที่ต้องการที่ได้ออกแบบไว้

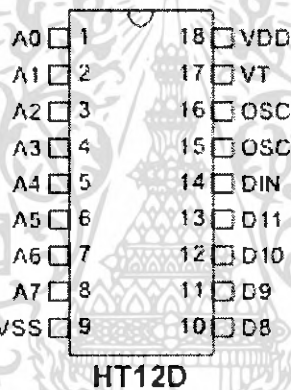


รูปที่ 2.10 Block Diagram แสดงการทำงานของภาครับไร้สาย RF

จาก Block Diagram จะเห็นว่าเมื่อภาครับ รับสัญญาณมาแล้ว Receiver จะทำการ แยกสัญญาณ Digital Code ออกจากคลื่นพาหะ แล้วขยายสัญญาณ เพื่อให้ข้อมูลที่ถูกลดทอนไประหว่างการส่งไร้สายคือส่งผ่านอากาศ ให้สัญญาณแรงพอเพื่อนำไปใช้งานได้ สมบูรณ์มากที่สุด จากนั้นจะส่งสัญญาณไปให้กับภาค Decoding เพื่อทำงาน ถอดรหัสออกมา เป็นสัญญาณ Binary เพื่อเอาคำสั่งที่ถูกกำหนดขึ้นส่งให้กับภาคควบคุมการทำงานของ Motor เพื่อให้ Motor ทำงานตามที่ต้องการ



รูปที่ 2.11 Block Diagram การทำงานของ HT12D



รูปที่ 2.12 แสดง Pin ของ ICเบอร์ HT12D

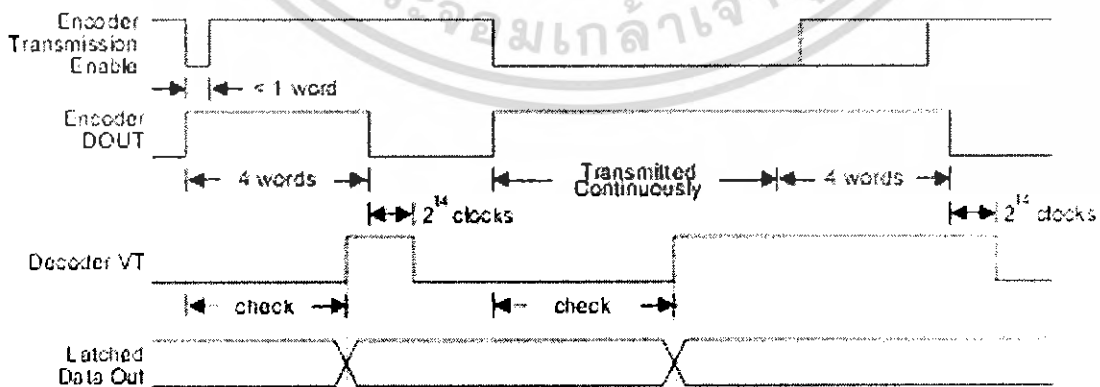
จากรูปที่ 2.11 เป็นการแสดงภาพโครงสร้าง Block Diagram การทำงานภายในของ HT12D F โดยละเอียด ส่วนรูปที่ 2.10 เป็นการแสดงโครงสร้างภายนอกทางกายภาพของ HT12D

การทำงานของ HT12D จะสามารถถอดรหัสได้ต้องอาศัยความสัมพันธ์ของ VDD ที่ป้อนให้กับ HT12D และค่าความต้านทานที่จะนำมาต่อที่ OSC1 กับ OSC2 ต้องมีความสัมพันธ์กับความถี่ที่ภาคส่ง ส่งมาสามารถหาค่าความต้านทานที่ถูกต้อง ได้จากการฟที่จะแสดงต่อไป

ตารางที่ 2.3 ตารางแสดงคำอธิบายของ Pin IC เบอร์ HT12D

Pin Name	I/O	Internal Connection	Description
A0 ~ A11	I	NMOS TRANSMISSION GATE	Input pins for address A0 ~ A11 setting . They can be externally set to VDD or VSS
D8 ~ D11	O	CMOS OUT	Output data pins
Din	I	CMOS IN	Serial data input pin
VT	O	CMOS OUT	Valid Transmission, active high
OSC1	I	OSCILLATOR	Oscillator input pin
OSC2	O	OSCILLATOR	Oscillator output pin
VSS	I	-	GND
VDD	I	-	Positive power supply

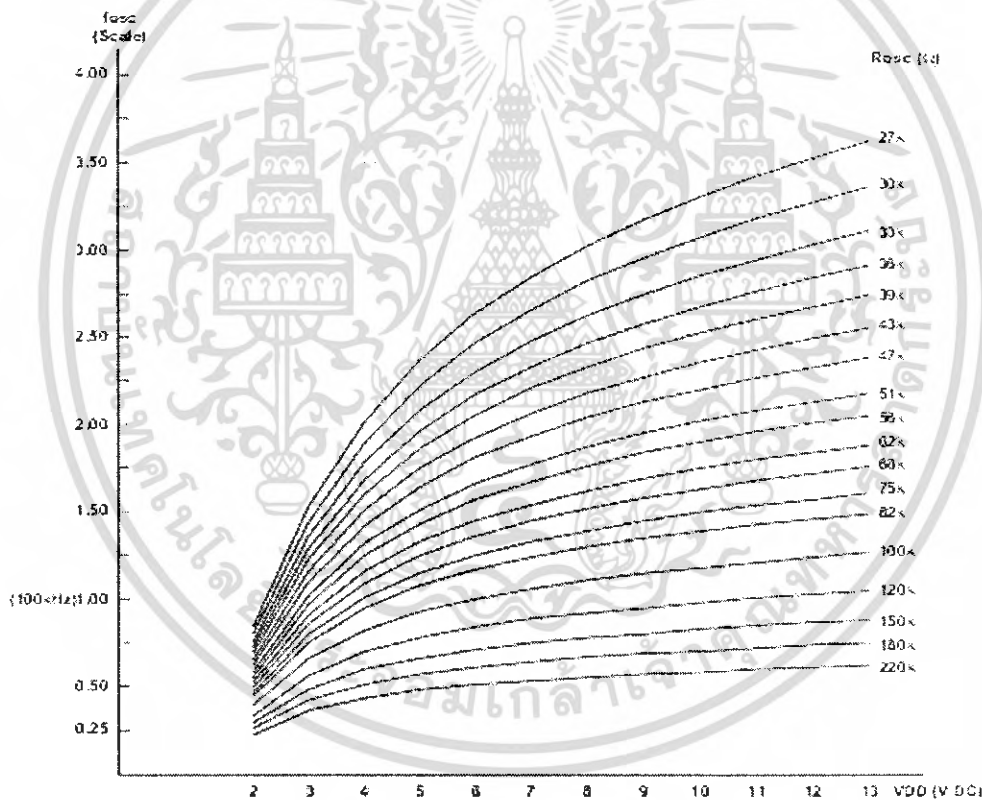
จะเห็นว่า Pin แต่ละ Pin จะมีหน้าที่ที่แตกต่างกัน โดยอาจจะเป็น Input หรือว่า Output ของส่วนต่างๆที่อยู่ใน IC เพื่อรับเอา signal encoding มาแล้ว decoding เอาข้อมูลที่ส่งออกมาตามต้องการซึ่งแสดงหน้าที่โดยรวมของ IC ได้ตาม Block Diagram ดังรูปที่ 2.11



รูปที่ 2.13 แสดงการทำงานของ Decoder timing

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.11 จะเห็นได้ว่าเมื่อทำการกดปุ่มที่ตัวส่งไม่ว่าจะเป็นคำสั่งอะไรก็ตามโดยคำสั่งนั้นจะ active Low เมื่อกดคำสั่งแล้วจะถูกการเข้ารหัสคั่งที่ได้กล่าวมาข้างต้น จะถูกส่งออกมาที่ Output โดยลักษณะของการส่ง Code ออกมานั้นจะส่งมาตลอดอย่างซึ่งหมายความว่าเมื่อในการกดคำสั่งแต่ละครั้งจะเกิด Word เดียวกันขึ้นต่อเนื่องกันจนกว่าเลิกกดคำสั่งนั้นซึ่งในที่นี้เมื่อกดจะเป็น Low เมื่อเลิกกดจะเป็น High จากนั้นในการเข้ารหัสนั้นเมื่อ ลักษณะเป็น High จะมี Function Code ของคำสั่งนั้นออกมาและลักษณะเป็น High จะมี End Code แสดงอย่างต่อเนื่องจนกว่าจะถูกกดคำสั่งอีกครั้ง ส่วนการถอดรหัสหรือ Decoding นั้นจะเริ่มการ Decoding เมื่อ Encoding Output เกิด Word 2 ครั้งดังรูปแล้วก็จะ Delay ไป แล้วก็จะ Decoding เอาคำสั่งไว้สายออกที่ Output โดยที่จะแสดงคำสั่งในสถานะ High จากนั้นจะเลิกการ Decoding เมื่อผ่าน End Code ไป 2 ครั้งดังรูป แล้วก็จะ Delay ไปอีกแล้วก็จะจบการ Decoding ของคำสั่งในแต่ละครั้ง

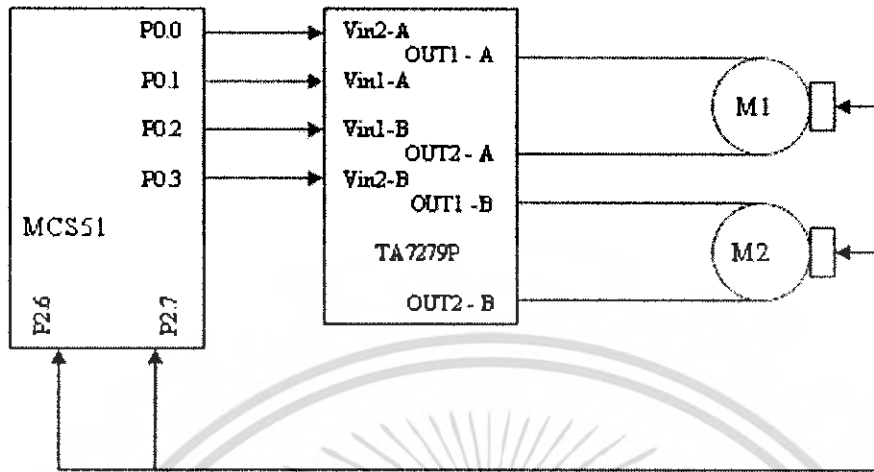


รูปที่ 2.14 กราฟแสดงความสัมพันธ์ระหว่าง VDD และค่าความต้านทาน

จากกราฟ รูปที่ 2.12 สามารถหาความถี่ที่ต้องการได้จากการอ่านกราฟ โดยต้องรู้ค่า VDD และ ความถี่ที่ต้องการ แล้วจึงหาจุดตัดกราฟเพื่อให้ได้ค่า ความต้านทานที่ต้องการเพื่อไปใส่ในวงจร ที่ได้ออกแบบไว้ได้อย่างถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 ไมโครคอนโทรลเลอร์ และ Drive Motor



รูปที่ 2.15 Diagram การเชื่อมต่อ AT89C1051 กับ L293D

จากรูปที่ 2.15 ใช้ไมโครคอนโทรลเลอร์ตระกูล MCS-51 เบอร์ AT89C51 เนื่องจาก MCS-51 ไม่สามารถจ่ายกระแสให้มอเตอร์หมุนได้ ดังนั้นจึงต้องใช้วงจร Drive Motor ช่วยในการจ่ายกระแสให้กับมอเตอร์ในโครงการนี้ในการควบคุมมอเตอร์นั้นจะใช้ IC สำเร็จรูปในการขับมอเตอร์คือ IC เบอร์ AT7279P แสดงการเชื่อมต่อ MCS-51 กับ AT7279P

2.5 ไมโครคอนโทรลเลอร์ MCS-51

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

2.5.1 โครงสร้างภายในของไมโครคอนโทรลเลอร์ MCS-51

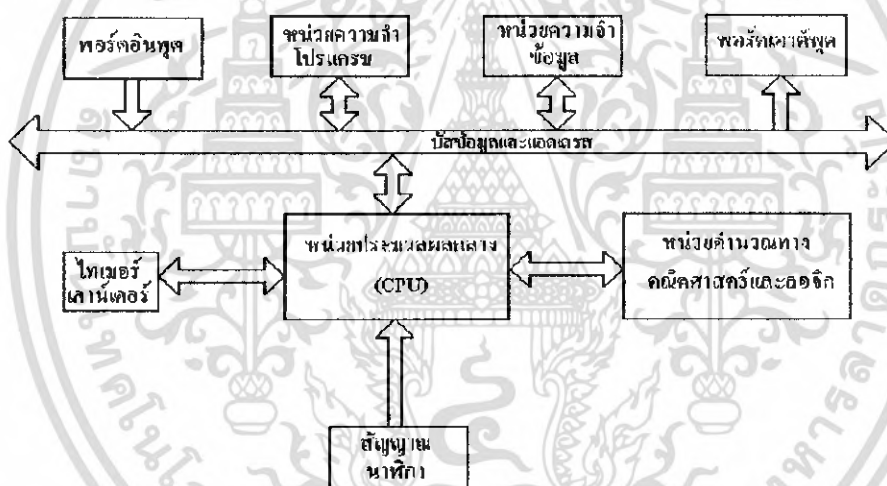
โครงสร้างภายในพื้นฐาน ของไมโครคอนโทรลเลอร์ MCS-51 แสดงในรูปที่ 2.1 ประกอบด้วยอุปกรณ์ต่างๆ ดังนี้

-หน่วยประมวลผลกลางขนาด 8 บิต

-หน่วยประมวลผลสำหรับข้อมูลแบบบิต (BOOLEAN PROCESSOR)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ความสามารถในการอ้างตำแหน่งของหน่วยความจำโปรแกรม 64 กิโลไบต์
- ความสามารถในการอ้างตำแหน่งของหน่วยความจำข้อมูล 64 กิโลไบต์
- หน่วยความจำโปรแกรมภายในขนาด 4 กิโลไบต์ แบบ อีพรอม (เบอร์ 8451)
- หน่วยความจำแบบ แรม ภายในจำนวน 128 ไบต์
- พอร์ตอินพุต/เอาต์พุตแบบขนานจำนวน 32 เส้น ซึ่งสามารถแยกทำงานได้อย่างอิสระ
- วงจรมับ/จับเวลาขนาด 16 บิต จำนวนสองวงจร
- วงจรถ่ายส่งแบบอนุกรมแบบคู่เฟล็กเต็ม(FULL DUPLEX)
- วงจรถวควบคุมการอินเตอร์รัปต์จากแหล่งกำเนิดสัญญาณ 6 ประเภท พร้อมการกำหนดลำดับวงจรถผลิตสัญญาณนาฬิกาภายในซึ่งโครงสร้างการทำงานทั้งหมดของไมโครคอนโทรลเลอร์จะ อาศัยหลักการการทำงานที่เกี่ยวข้องกัน โดยอาศัยหลักการการทำงานที่เป็นไปตามโครงสร้างเสมอ



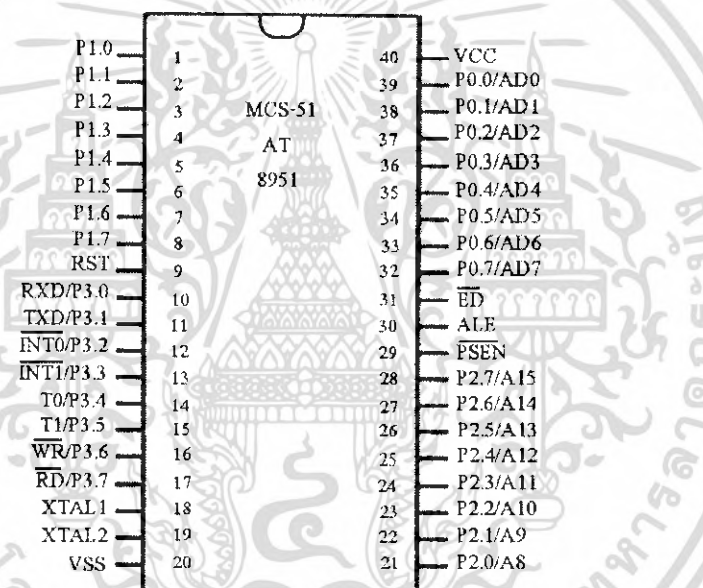
รูปที่ 2.16 แสดง โครงสร้างภายในของไมโครคอนโทรลเลอร์ MCS-51

โดยมากแล้วไมโครคอนโทรลเลอร์ตระกูลนี้มักจะมีรูปร่างของไอซีเป็นแบบขนาด 40 ขาดังแสดงในรูปที่ 2.16 ซึ่งแต่ละขาสัญญาณจะมีหน้าที่ระบุชัดเจนตามสัญลักษณ์ชื่อย่อที่กำกับในแต่ละขาอย่างไรก็ตามจะมีบางขาสัญญาณที่อาจจะทำหน้าที่ได้มากกว่าหนึ่งอย่าง (ซึ่งเขียนกำกับไว้ว่าALTERNATE FUNCTION ในรูปที่ 2.16) ซึ่งจะไม่สามารถใช้งานในเวลาเดียวกันได้ ตัวอย่างเช่นขาสัญญาณบิต 0 ของพอร์ต 3 (ใช้ตัวย่อเป็น P3.0) อาจจะใช้เป็นขาสัญญาณเอาต์พุตหรืออินพุตตาม ปกติ ภายในไมโครคอนโทรลเลอร์ MCS-51 ซึ่งประกอบด้วยหน่วยการทำงานต่างๆ ภายในไอซีMCS-51 จำนวนมาก โดยแต่ละบล็อกซึ่งเป็นวงจรถวควบคุมรีจิสเตอร์ (REGISTER) หรือหน่วยความจำภายในของไอซี MCS-51 จะถูกเชื่อมต่อเข้าด้วยกันผ่านทางเส้นเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณที่เรียกว่าบัสข้อมูลภายใน รีจิสเตอร์และหน่วยความจำเหล่านี้จะถูกนำไปใช้ระหว่างการประมวลผลคำสั่ง หน้าที่ของโปรแกรมที่ผู้ใช้สร้างขึ้นมาก็เป็นการควบคุมการรับหรือส่งข้อมูลระหว่างรีจิสเตอร์เหล่านี้ ซึ่งอาจจะมีการดำเนินการร่วมกับหน่วยการดำเนินงานประมวลผลทางคณิตศาสตร์และลอจิก หรือเรียกว่า ARITHMETIC AND LOGIC UNIT :ALU

2.5.2 โครงสร้างหน่วยความจำภายในไมโครคอนโทรลเลอร์ MCS-51

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

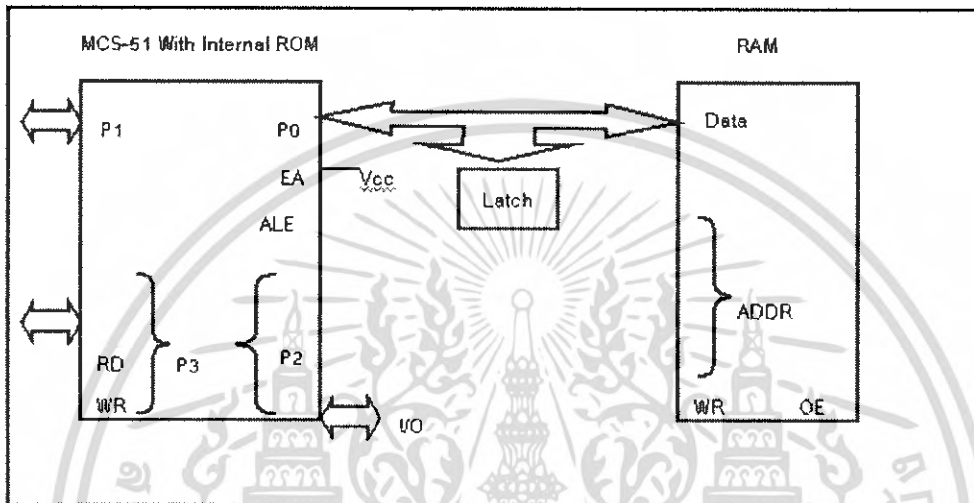


รูปที่ 2.17 แสดงรูปร่างและการจัดวางขาต่างๆ ของไมโครคอนโทรลเลอร์ MCS-51

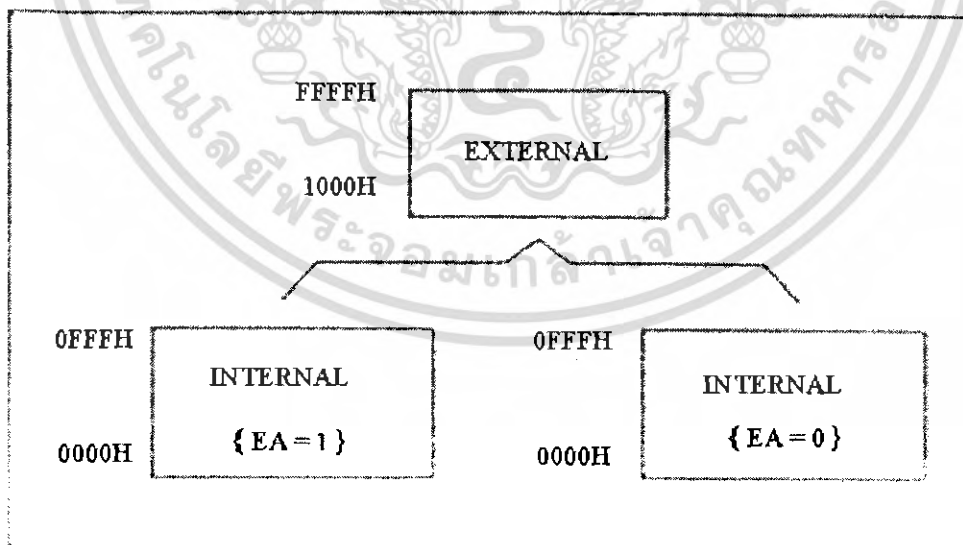
2.5.3 หน่วยความจำโปรแกรม

หน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ MCS-51 เป็นบริเวณหน่วยความจำสำหรับเก็บข้อมูลและคำสั่งใช้งานต่างๆ ซึ่งแม้ว่าจะไม่มีการจ่ายกระแสไฟฟ้าให้กับระบบข้อมูลเหล่านี้ก็ยังคงอยู่ไม่สูญหาย โครงสร้างของหน่วยความจำโปรแกรมมีลักษณะ เช่นเดียวกับหน่วยความจำที่บรรจุอยู่ในไมโครคอนโทรลเลอร์ MCS-51 ของหน่วยความจำประเภทต่างๆ เช่น หน่วยความจำแบบรอม (READ ONLY MEMORY) หรืออีพรอม (ERASABLE PROGRAMABLE READ ONLY MEMORY) ในไมโครคอนโทรลเลอร์ MCS-51 สามารถอ่านข้อมูลหน่วยความจำโปรแกรมนี้ได้สูงสุดไม่เกิน 64 กิโลไบต์ และแยกประเภทของหน่วยความจำโปรแกรมเป็น 2 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะ ตามตำแหน่งของหน่วยความจำนั้นคือ หน่วยความจำโปรแกรมภายใน (INTERNAL PROGRAM MEMORY) ซึ่งเป็นหน่วยความจำรอม หรืออีพรอม ที่อยู่ภายในตัวไอซีของไมโครคอนโทรลเลอร์เอง และหน่วยความจำโปรแกรมภายนอก (EXTERNAL PROGRAM MEMORY) ซึ่งเป็นการใช้ไอซีหน่วยความจำมาทำหน้าที่เป็นหน่วยความจำโปรแกรมของระบบ



รูปที่ 2.18 แสดงการใช้หน่วยความจำสำหรับเก็บโปรแกรม



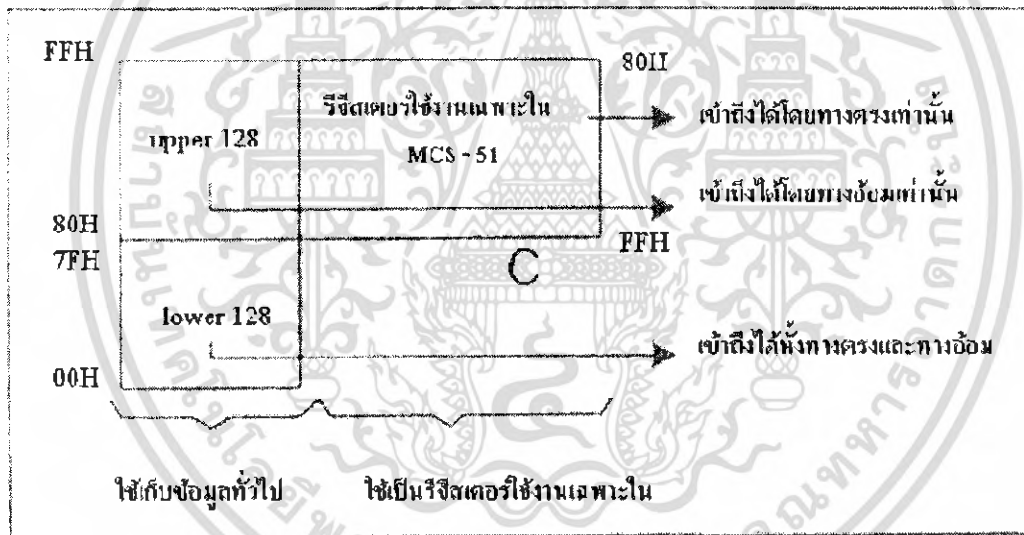
รูปที่ 2.19 แสดงการจัดพื้นที่ของหน่วยความจำโปรแกรมภายในและภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ไมโครคอนโทรลเลอร์เบอร์ต่างๆ ของตระกูล 8051 นี้สามารถขยายให้ใช้งานในหน่วยความจำภายนอกได้ทั้งสิ้น โดยกรณีที่มีหน่วยความจำโปรแกรมภายในอยู่แล้วการอ้างตำแหน่งแอดเดรสที่มีทั้ง ในหน่วยความจำโปรแกรมภายใน และภายนอกนั้นจะต้องทำการควบคุมระดับลอจิกของสัญญาณ ในขณะที่ด้วยขนาดหน่วยความจำโปรแกรมภายในของไมโครคอนโทรลเลอร์เบอร์ต่างๆ ภายในตระกูล 8051จะแตกต่างกันออกไป เพื่อความเหมาะสมกับการนำไปใช้งานลักษณะต่างๆ

- 8051และ8052มีหน่วยความจำแบบรวม4 และ 8 กิโลไบต์
- 8751 มีหน่วยความจำแบบ อีพรอม ขนาด 4 กิโลไบต์ข้อมูลที่จัดเก็บภายในนี้ ซึ่งสามารถใช้แสงอุลตราไวโอเลตลบ และนำกลับไปบรรจุโปรแกรมใหม่ได้อีกครั้งหนึ่ง
- 8031 และ 8032 ไม่มีหน่วยความจำโปรแกรมอยู่ภายในตัวไอซี ดังนั้นในการนำไปใช้งานจึงจำเป็นต้องอาศัยหน่วยความจำโปรแกรมภายนอกเสมอ

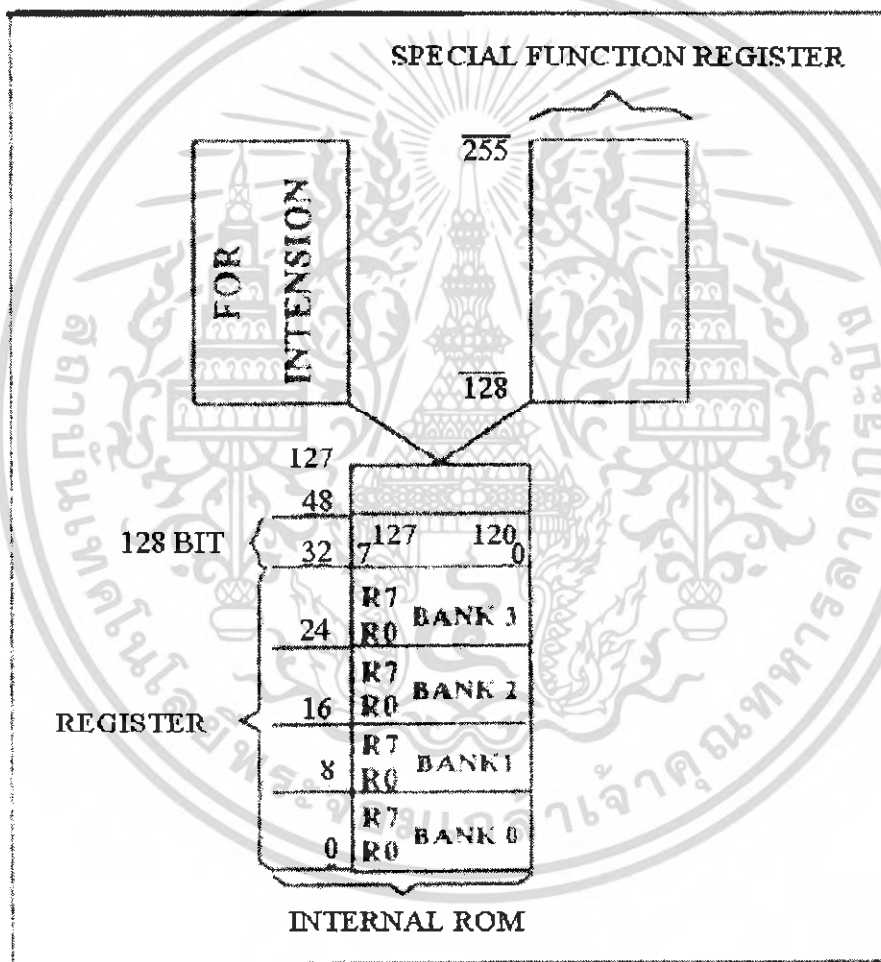


รูปที่ 2.20 แสดงหน่วยความจำสำหรับเก็บข้อมูลภายใน ไมโครคอนโทรลเลอร์ MCS-51

2.5.4 หน่วยความจำข้อมูล

หน่วยความจำข้อมูล (DATA MEMORY) ซึ่งโดยพื้นฐานแล้วเป็นหน่วยความจำแรมสามารถเขียนหรืออ่านข้อมูลได้ (READ OR WRITE MEMORY) ใช้สำหรับเก็บข้อมูลหรือตัวแปรที่เกิดขึ้นในขณะที่กำลังประมวลผลโปรแกรมไว้เป็นการชั่วคราว ซึ่ง โดยพื้นฐานแล้ว หน่วยความจำข้อมูลจัดเป็นหน่วยความจำแรมแบบสแตติก ดังนั้นเมื่อไม่มีการจ่ายไฟฟ้าให้กับระบบก็จะมีผลทำให้ข้อมูลที่จัดเก็บไว้ภายในหน่วยความจำนี้สูญไป พื้นที่ของหน่วยความจำข้อมูลของ

ไมโครคอนโทรลเลอร์ MCS-51 มีได้สูงสุดไม่เกิน 64 กิโลไบต์ และแยกประเภทออกเป็นสองลักษณะตามตำแหน่งที่ตั้งของหน่วยความจำนั้น ตามลักษณะของหน่วยความจำโปรแกรมภายในซึ่งก็เป็นแรมที่อยู่ภายในตัวไอซีในตระกูลของไมโครคอนโทรลเลอร์ และหน่วยความจำข้อมูลภายนอกซึ่งเป็นการใช้ไอซีหน่วยความจำแรมมาเพิ่ม เดิมเข้าไปในวงจรลักษณะเดียวกับ การนำไอซีอีพรอมมาใช้งานเป็นหน่วยความจำโปรแกรมนั่นเอง

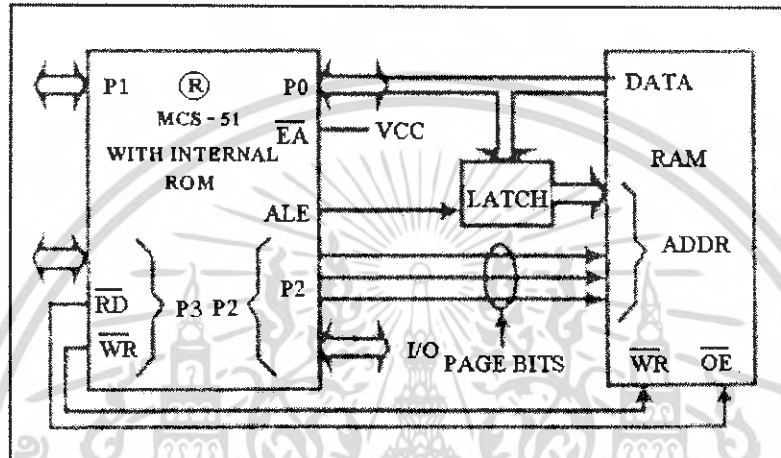


รูปที่ 2.21 แสดงการจัดหน่วยความจำข้อมูล

โดยที่หน่วยความจำสำหรับเก็บข้อมูลของไมโครคอนโทรลเลอร์ MCS-51 นี้สามารถแบ่งออกเป็น 2 ส่วนคือ ในส่วนที่เป็นหน่วยความจำสำหรับเก็บข้อมูลภายในไอซี และหน่วยความจำสำหรับเก็บข้อมูลภายนอกไมโครคอนโทรลเลอร์ MCS-51 ทุก ๆ เบอร์จะมีหน่วยความจำเก็บข้อมูลทั่ว ๆ ไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภายในไอซีอย่างน้อย คือ 128 ไบต์ ไปจนถึง 256 ไบต์ ทั้งนี้ขึ้นกับเบอร์ของไอซี หน่วยความจำสำหรับเก็บข้อมูลภายในไอซีในบริเวณ 128 ไบต์เรียกว่า LOWER 128 และในบริเวณ 128 ไบต์หลังที่มีเพิ่มในบางเบอร์มีชื่อเรียกว่า UPPER

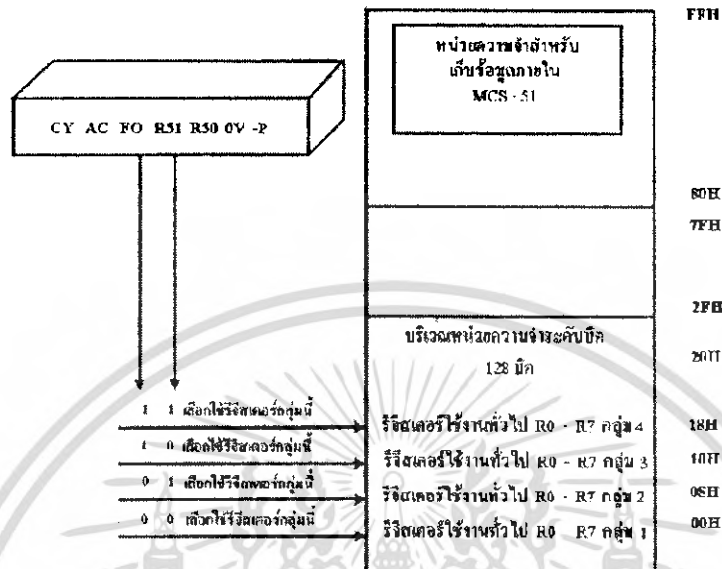


รูปที่ 2.22 แสดงการต่อกับหน่วยความจำข้อมูลภายนอกไอซี

2.5.5 รีจิสเตอร์ที่เกี่ยวข้องกับไมโครคอนโทรลเลอร์ MCS-51

รีจิสเตอร์ในกลุ่มนี้ จะเป็นรีจิสเตอร์ขนาด 16 บิตที่ใช้งาน เพื่อเก็บข้อมูลของตัวแอดเดรส เป็นสำคัญโดยค่าที่อยู่ภายในแอดเดรสนี้จะนำไปเป็นค่าของข้อมูลที่ส่งออกไปทางบัสแอดเดรส ใน ส่วนของไมโครคอนโทรลเลอร์ เพื่อบอกตำแหน่งที่ต้องการติดต่อ รีจิสเตอร์ที่จัดในกลุ่มนี้ ประกอบด้วยรีจิสเตอร์ใช้งานทั่วไป (GENERAL-PURPOSE REGISTERS) รีจิสเตอร์ในกลุ่มนี้ จัดเป็นพื้นที่หน่วยความจำที่ใช้ในการสนับสนุนในการประมวลผล การทำงานจากหน่วยประมวลผลทางคณิตศาสตร์และลอจิก (ALU) เพื่อให้สามารถจัดการข้อมูลให้เร็วที่สุด นอกจากนี้ โปรแกรมที่ไม่ได้ใช้คำสั่งเหล่านี้ก็ยังใช้เป็นการเก็บข้อมูลตัวแปรภายใน โปรแกรม จะเห็นได้ว่าชื่อของรีจิสเตอร์ไม่ว่าจะอยู่ในรีจิสเตอร์แบงก์ใด ก็จะมีชื่อว่า R0 ถึง R7 เหมือนกันทั้งสิ้น ดังนั้นในการใช้งานผู้ใช้งานจะต้องให้ความระมัดระวังว่า ต้องการรีจิสเตอร์นั้นๆ จากแบงก์ใดๆ ซึ่งการกำหนดเลือกแต่ละกลุ่มของรีจิสเตอร์นี้ก็ทำได้ง่าย เพียงการกำหนดค่าของบิตที่อยู่ภายในแฟลช (PSW) เท่านั้นอย่างไรก็ตาม โดยทั่วไปก็มักจะมีการใช้งานรีจิสเตอร์ R0 ถึง R7 เฉพาะในแบงก์ 0 เท่านั้น ดังนั้นพื้นที่ของแบงก์อื่นๆ ที่เหลือก็สามารถนำมาใช้ในลักษณะของหน่วย ความจำแรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.23 แสดงการเลือกใช้รีจิสเตอร์ใช้งานทั่วไป R0-R7 ในแต่ละกลุ่ม

- **รีจิสเตอร์หน้าที่พิเศษ** เป็นรีจิสเตอร์หน้าที่พิเศษ (SFR) เป็นรีจิสเตอร์สำหรับการควบคุมหน้าที่ และการทำงานของอุปกรณ์หรือพอร์ตของไมโครคอนโทรลเลอร์ MCS-51 ทั้งหมด ตำแหน่งของรีจิสเตอร์เหล่านี้จะจัดอยู่ในบริเวณแอดเดรส 80H-FFH การใช้งานรีจิสเตอร์หน้าที่พิเศษเหล่านี้สามารถทำได้ทั้งการระบุชื่อของรีจิสเตอร์ หรือตำแหน่งแอดเดรส ที่เป็นของรีจิสเตอร์นั้นก็ได้การจัดพื้นที่หน่วย ความจำสำหรับรีจิสเตอร์หน้าที่พิเศษเหล่านี้ โดยมีข้อสังเกตว่า รีจิสเตอร์ที่อยู่ในตำแหน่งแอดเดรสที่มีจำนวนเป็นทวีคูณของค่า 8 จะสามารถอ้างถึงในระดับบิตได้ด้วย (นั่นคือแอดเดรส 80H 88H 90H A0H A8H B0H B8H D0H E0H และ F0H)

- **แอกคิวมูลเตอร์ (ACCUMULATOR) หรือ ACC** เป็นรีจิสเตอร์ขนาด 8 บิต ทำหน้าที่ในการเก็บข้อมูลที่จะส่งให้กับหน่วยทำงานภายในหน่วยประมวลผลกลาง และเก็บผลลัพธ์ที่ได้จากการทำงานเท่านั้น การทำงานของรีจิสเตอร์นี้มีลักษณะเช่นเดียวกับตัวแอกคิวมูลเตอร์ของโปรเซสเซอร์ทั่วไป การใช้งานในโปรแกรมซึ่งใช้เรียกเป็น รีจิสเตอร์ A

2.5.6 ชุดคำสั่งของไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์ MCS-51 ประกอบด้วยคำสั่งทั้งหมดจำนวนมาก ซึ่งสามารถจะจัดกลุ่มคำสั่งเหล่านี้ตามลักษณะและหน้าที่การทำงานที่คล้ายคลึงกัน เพื่อความสะดวกต่อการศึกษาทำความเข้าใจและใช้งานดังนี้

2.6.6.1 กลุ่มการถ่ายเทข้อมูล คือ กลุ่มคำสั่งในการโอนย้ายข้อมูล ทำหน้าที่ในโอนย้ายข้อมูลระหว่างรีจิสเตอร์ หรือหน่วยความจำภายในแรม โดยมีรายละเอียดดังนี้ ชุดคำสั่งในการถ่ายเทแรม ภายในนั้น ซึ่งเวลาที่ใช้ในหนึ่งคำสั่งนั้นจะเป็นเวลา เมื่อขณะที่ความถี่ในการทำงานของหน่วยประมวลผลกลางที่ความถี่ 12 เมกะเฮิร์ตซ์ และรายละเอียดของแต่ละคำสั่งมีดังนี้ MOV :จะทำงานในลักษณะเป็นการถ่ายเทข้อมูลที่มีขนาดเป็นไบต์ หรือ บิตก็ได้ จากแหล่งกำเนิดเข้าสู่ตัวรับข้อมูลในฟิลด์ โอเปอร์เรนด์ PUSH:จะทำงานโดยเพิ่มค่ารีจิสเตอร์ SP ก่อนแล้วจึงทำการถ่ายเทข้อมูล 1 ไบต์จากแหล่งกำเนิดไปบริเวณสแต็กตามตำแหน่งที่รีจิสเตอร์ SP กำหนด POP:การถ่ายเทข้อมูลขนาด 1 ไบต์ จากบริเวณตำแหน่งที่รีจิสเตอร์ SP กำหนดไปยังรีจิสเตอร์ที่โอเปอร์เรนด์ กำหนดและหลังจากนั้น รีจิสเตอร์ SP จะลดค่าลง XCH:คำสั่งแลกเปลี่ยนไบต์ระหว่างแหล่งกำเนิดโอเปอร์เรนด์กับรีจิสเตอร์ AXCHD คำสั่งในการแลกเปลี่ยนขนาดนิบเบิลทางอันดับต่ำของแหล่งกำเนิดโอเปอร์เรนด์กับนิบเบิลอันดับต่ำของแอกคิวมูลเตอร์ ตัวอย่างเช่น ทำการเลื่อนข้อมูลไป 2 ไบต์ทางขวามือ ซึ่งจะมี 2 วิธีคือ ใช้คำสั่ง MOV หรือใช้คำสั่ง XCH รายละเอียดการใช้คำสั่งทั้ง 2 แบบ

2.5.6.2 กลุ่มคำสั่งทางคณิตศาสตร์ เช่น การบวก ลบ คูณ และหารข้อมูลภายในตัวรีจิสเตอร์ต่างๆ ช่วงเวลาการทำงานของแต่ละคำสั่งนั้นจะกำหนดที่ความถี่ของสัญญาณนาฬิกาที่ 12 เมกะเฮิร์ตซ์ คำสั่งทางคณิตศาสตร์ส่วนใหญ่ใช้เวลา 1 ms ยกเว้นคำสั่ง INC DPTR ซึ่งใช้เวลา 2 ms โดยที่คำสั่งการคูณและหารใช้เวลา 4 ms โดยมีรายละเอียดการใช้คำสั่งมีดังนี้ INC:เป็นการบวกหนึ่งกับโอเปอร์เรนด์และใส่ค่าใหม่กลับเข้าที่ตัวโอเปอร์เรนด์นั้น ๆ DEC:เป็นการลบออกจากตัวเลขที่อยู่ในแหล่งกำเนิด โอเปอร์เรนด์ และนำผลลัพธ์ที่ได้มาเก็บไว้ที่ตัวโอเปอร์เรนด์นั้น ADD:เป็นการบวกในแอกคิวมูลเตอร์เข้ากับค่าในแหล่งกำเนิดโอเปอร์เรนด์ ADDC:เป็นการบวกค่าต่างๆ ในแอกคิวมูลเตอร์เข้ากับค่าในแหล่งกำเนิดโอเปอร์เรนด์และบวกกับบิตทดด้วย SUBB:เป็นการนำเลขที่แหล่งกำเนิดโอเปอร์เรนด์ ลบออกจากตัวเลขใน A และนำค่าบิตตัวทดมาลบออกอีกและผลลัพธ์ที่ได้นำมาใส่ลงในแอกคิวมูลเตอร์ A MUL:เป็นการคูณแบบไม่คิดเครื่องหมายของตัวเลขที่อยู่ใน แอกคิวมูลเตอร์กับ เลขในรีจิสเตอร์ B แล้วได้ผลลัพธ์ 2 ไบต์ นำมาเก็บไว้ที่ AB โดย A จะรับอันดับต่ำส่วน B จะรับอันดับสูง DIV:เป็นคำสั่งในการหารแบบไม่คิดเครื่องหมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่อยู่ในแอกคิวมูลเตอร์แล้วหารตัวเลขในรีจิสเตอร์ B แล้วนำผลลัพธ์ไปเก็บในแอกคิวมูลเตอร์ และเศษของการหารตัวเลขจะเก็บไว้ในรีจิสเตอร์ B DA: สำหรับการบวกกันทางตัวเลข BCD เป็นการปรับค่ารวมซึ่งเป็นผลมาจากการบวกกันทางไบนารีของระบบตัวเลข BCD ขนาด 2 หลักสอง จำนวนการปรับค่าตัวเลขผลรวมด้วยการใช้คำสั่ง DA จะได้ผลลัพธ์กลับมาที่แอกคิวมูลเตอร์ รายละเอียดการใช้คำสั่ง

2.5.6.3 กลุ่มคำสั่งทางตรรกศาสตร์หรือแมบลอจิก ทำหน้าที่เกี่ยวกับการประมวลผลแบบ ลอจิก ต่างๆ เช่นการ AND OR หรือ EX-OR ระหว่างข้อมูลในรีจิสเตอร์ A นั้นเองโดยมีการใช้ คำสั่งดังนี้ CPL:เป็นการใช้คำสั่งกลับค่าหรือคอมพลิเมนต์ ข้อมูลในแอกคิวมูลเตอร์จะไม่มีผลใดๆ ต่อค่าของแฟล็ก หรือการอ้างถึงตำแหน่งแอดเดรสนั้นตามบิตนั้นๆ RL, RLC, RR, RRC, SWAP: ทั้ง 5 คำสั่งนี้เป็นคำสั่งในการทำงานการวนบิตบนตัวของแอกคิวมูลเตอร์ ซึ่ง RL เป็นการวนบิต ทางขวา, RLC เป็นการทำการวนทางซ้ายผ่านบิตทด, RRC เป็นการวนขวาผ่านบิตทด และ SWAP เป็นการวนซ้ายสี่ครั้ง ANL:เป็นการ ADD กันทางตรรกศาสตร์ ระหว่างแหล่งกำเนิดสองโอเปอร์ แรนด์ ซึ่งจะสั่งให้ทำงานในรูปแบบของตรรกศาสตร์ทางข้อมูลขนาดเป็น ไบต์หรือบิตกลุ่มคำสั่ง แบบบูลีนหรือแบบบิตซึ่งเป็นความสามารถ ของไมโครคอนโทรลเลอร์ MCS-51 ที่จะดำเนินการ ประมวลผลแบบบิตแทนที่จะเป็นข้อมูล ทั้งไบต์เช่นปกติ โดยมีชุดคำสั่งที่จัดการโดยตรง ทุกคำสั่ง จะเข้าถึงข้อมูลโดยตรงในระดับบิต โดยมีการบิตแอดเดรสได้ตั้งแต่ 00H - 7FH ในพื้นที่ 128 บิต หน่วยความจำข้อมูลภายในและบิตแอดเดรส 80H-FFH ในบริเวณกลุ่มรีจิสเตอร์ฟังก์ชันพิเศษ (SFR)

2.5.6.4 กลุ่มคำสั่งในการกระโดดไปยังตำแหน่งต่างๆภายในโปรแกรม ซึ่งจะเปลี่ยนลำดับ ของการประมวลผลภายในโปรแกรมไปยังส่วนต่างๆแทนที่จะดำเนินการไปเป็นลำดับ ต่อเนื่อง โดยที่คำสั่ง JMP จะแบ่งเป็น 3 ลักษณะ คือ SJMP, LJMP, AJMP ซึ่งในแต่ละคำสั่งจะมีข้อแตกต่าง ของการกระโดดไปยังแอดเดรสไกลสุดที่ต่างกันคำสั่ง JMP ซึ่งเป็นแบบโมนิซิกที่สามารถจะใช้ได้ โดยมีรายละเอียดการใช้งานของคำสั่งดังต่อไปนี้ SMP:จะเป็นการกระโดดแบบการย้ายอันดับ ตำแหน่งของแอดเดรสตำแหน่งเดิมซึ่งจะสามารถกระโดดได้ -128 ถึง +127 ไบต์ AJMP:ลักษณะ แบบนี้จะสามารถกระโดดได้ไกลสุดประมาณ 2 กิโลไบต์ ซึ่งจะใช้หน่วยความจำเพียง 2 ไบต์ เท่านั้น ในการกำหนด LJMP:ลักษณะแบบนี้จะสามารถกระโดดได้ไกลสุดประมาณ 64 กิโลไบต์ ซึ่งจะใช้หน่วยความจำเพียง 3 ไบต์เท่านั้นในการกำหนด JMP @A+DPTR:เป็นการควบคุมการ กระโดดไปยังโปรแกรมที่ต้องการเฉพาะภายในส่วนต่างๆ

2.5.7 โครงสร้างการอินเตอร์รัปต์ของไมโครคอนโทรลเลอร์ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากแผนภาพโครงสร้างระบบอินเทอร์รัปต์ของไมโครคอนโทรลเลอร์ MCS-51 สัญญาณที่เข้ามาทำ การอินเทอร์รัปต์ MCS-51 นั้นเกิดขึ้นได้ 5 ลักษณะตามตารางข้อมูล ในรูปที่ 2.6 โดยจะเห็นได้ว่าสามารถที่จะกำหนดเลือกเพื่อยินยอม (หรืออินาเบิล : ENABLE) และห้าม (หรือ ดิสเอเบิล:DISABLE) ไม่ให้มีการอินเทอร์รัปต์แต่ละประเภทได้ โดยการกำหนดบิตของข้อมูลที่ เกี่ยวข้อง ซึ่งมักจะอยู่ภายในรีจิสเตอร์ TCONและSCON นอกจากนี้ยังมีตำแหน่งบิตภายใน รีจิสเตอร์ IE(INTERRUPT ENABLE REGISTER) ซึ่งทำหน้าที่เสมือนกับเป็นสวิตช์หลักที่ เกี่ยวข้องกับสัญญาณอินเทอร์รัปต์ทั้งหมด หากว่ากำหนดไม่ให้เกิดการอินเทอร์รัปต์ แล้วการ กำหนดบิตเพื่อห้ามหรือยินยอมของแต่ละอินเทอร์รัปต์ก็จะมีผลใดๆ เกิดขึ้นยังแสดงให้เห็นว่า สัญญาณอินเทอร์รัปต์ แต่ละประเภทยังสามารถกำหนดระดับความสำคัญ (PRIORITY) ของการ อินเทอร์รัปต์ได้สองลักษณะ ก็ระดับความสำคัญสูงหรือต่ำ (HIGH OR LOW PRIORITY) กล่าวคือ ขณะที่กำลังประมวลผลอยู่ภายในส่วนของโปรแกรมย่อยบริการอินเทอร์รัปต์ ของ สัญญาณที่มีระดับความสำคัญต่ำอยู่ ก็อาจจะถูกขัดจังหวะให้ไปประมวลผลของสัญญาณอินเทอร์-รัปต์ที่มีระดับความสำคัญสูงกว่า แต่หากว่าเป็นสัญญาณอินเทอร์รัปต์ที่มีระดับความสำคัญต่ำ เช่นเดียวกันแล้วก็ต้องรอให้เสร็จสิ้นการประมวลผลที่ ดำเนินการอยู่ก่อน

2.5.8 การรีเซต

โดยความหมายของการรีเซตเป็นการบังคับให้มีการเริ่มต้นใหม่อีกครั้งหนึ่ง ซึ่ง มักจะกระทำ โดยการกำหนดสถานะของสัญญาณที่ขาเรเซตของไอซี MCS-51ให้เป็นระดับลอจิกที่เหมาะสม สม เท่านั้น การรีเซตด้วยวิธีนี้ถือว่าการอินเทอร์รัปต์อย่างหนึ่งได้ แต่จะมีลักษณะต่างออกไปจาก การอินเทอร์รัปต์ของสัญญาณนี้ได้ ซึ่งมีศัพท์เฉพาะเรียกว่า NON-MASKABLE INTERRUPT นอกจากนี้การดำเนินการของ โปรแกรมก็แตกต่างออกไปด้วย โดยจะไม่มี การเก็บค่าของคำสั่งที่ กำลังจะไปทำในลำดับต่อไปภายในรีจิสเตอร์ PC เมื่อมีการรีเซตเกิดขึ้น โปรแกรมจะถูกสั่งให้ กระโดดไปยังแอดเดรส 0000 ทันทัน ซึ่งตำแหน่งนี้จะเป็ตำแหน่งเริ่มต้นของการทำงาน ของ ไมโครคอนโทรลเลอร์ MCS-51 เมื่อเริ่มจ่ายไฟให้กับระบบเมื่อใดก็ตามที่มีการรีเซตเกิดขึ้นค่า สถานะต่างๆ ภายในไมโครคอนโทรลเลอร์จะถูกกำหนดกลับไปเป็นค่าเริ่มต้นใหม่อีกครั้ง

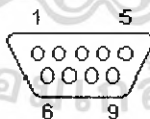
2.6 การใช้งานไมโครคอนโทรลเลอร์กับพอร์ตอนุกรม

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

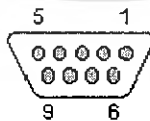
ข้อดีของพอร์ตอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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



(Male at the computer side)



(Female at the cable side)

Pin	Name	RS232	V.24	Dir	Description
1	CD	CF	109	←	Carrier Detect
2	RXD	BB	104	←	Receive Data
3	TXD	BA	103	→	Transmit Data

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4	DTR	CD	108.2	→	Data Terminal Ready
5	GND	AB	102	—	Signal Ground
6	DSR	CC	107	←	Data Set Ready
7	RTS	CA	105	→	Request to Send
8	CTS	CB	106	←	Clear to Send
9	RI	CE	125	←	Ring Indicator

รูปที่ 2.24 ตารางแสดงถึงหน้าที่ของขาต่าง ๆ

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.25 แสดงสัญญาณต่างๆที่ส่งในรูปแบบอนุกรม

2.6.1 หน้าที่ของสัญญาณต่างๆ มีดังนี้

Protective ground เป็นจุดที่ต่อกับตัวเปลือกของอุปกรณ์ และไม่ต่อกับสัญญาณใด ๆ ในระบบเพื่อใช้ต่อลงดิน เป็นการป้องกันอันตรายจากไฟลัดวงจร และใช้เป็น Shield ป้องกันการรบกวน

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

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

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

Request to send เป็นสัญญาณจาก DTE ส่งไปให้ DCE เพื่อขอส่งข้อมูลไปปรกติจะอยู่ในสถานะ "Off" เมื่อต้องการส่งข้อมูลจะเปลี่ยนเป็นสถานะ "On" จนกว่าการส่งเสร็จสิ้นจึงเปลี่ยนกลับมาที่สถานะ "Off" ตามเดิม ทั้งนี้ทาง DTE ต้องได้รับสัญญาณ CTS จึงจะสามารถส่งข้อมูล TD ไปยัง DCE ได้ และสัญญาณ RTS ที่กลับสู่สถานะ "Off" จะไม่สามารถเปลี่ยนเป็น "On" ใหม่ขณะที่สัญญาณ CTS อยู่ในสถานะ "On" ต้องรอจนกว่าสัญญาณ CTS เปลี่ยนมาอยู่ในสถานะ "Off" ก่อน เพื่อป้องกันการเกิด Overrun

Clear to send เป็นสัญญาณที่ DCE ส่งให้ DTE เพื่อแจ้งว่าพร้อมรับการส่งข้อมูลจาก DTE
Data set ready เป็นสัญญาณที่ DCE ส่งให้ DTE เพื่อแจ้งว่า DCE สามารถเชื่อมโยงไปยังปลายทางและพร้อมที่จะติดต่อแล้ว

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

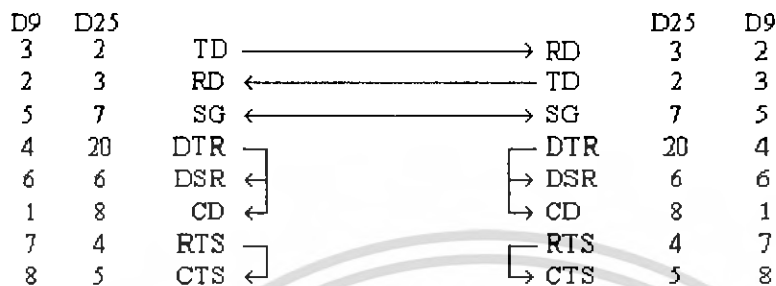
Data carrier detect เป็นสัญญาณที่ DCE ส่งให้ DTE เพื่อแจ้งว่าได้รับสัญญาณพาหะจาก DCE ที่อยู่อีกด้านหนึ่งของการสื่อสาร ซึ่งหมายความว่า ช่องการสื่อสารระหว่าง DCE ทั้ง 2 ไม่ขาดตอน พร้อมทั้งจะทำการสื่อสารได้ ซึ่งอุปกรณ์ DTE หรือโปรแกรมที่ควบคุมการสื่อสารมักจะตรวจสอบ สัญญาณนี้ ถ้าไม่อยู่ที่ on แสดงว่าช่องการสื่อสารขาด ก็จะไม่ทำการรับหรือส่งข้อมูล

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

2.6.2 Null Modems

Null Modem ใช้สำหรับเชื่อมโยงระหว่าง DTE 2 ตัวเข้าด้วยกันโดยตรง ซึ่งโดยมากใช้ในการถ่ายข้อมูลระหว่างคอมพิวเตอร์ หรือใช้ในการพัฒนาระบบไมโครโปรเซสเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.26 แสดงการเชื่อมต่อของสายสัญญาณ

2.6.3 Flow Control

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

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.4 รูปคลื่น สัญญาณ RS-232

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



รูปที่ 2.27 รูปคลื่นของสัญญาณที่ส่ง

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

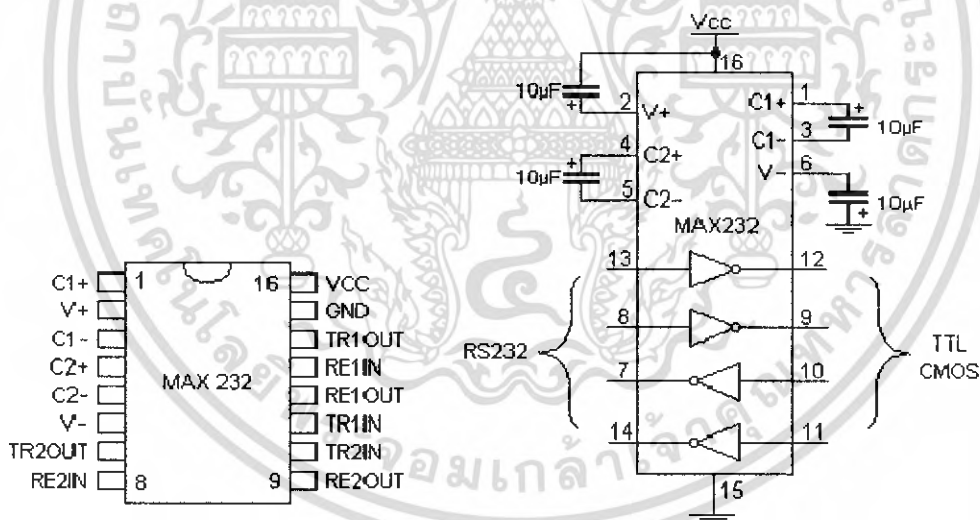
2.6.5 ตัวแปลงสัญญาณ RS-232

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

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



รูปที่ 2.28 แสดง โครงสร้างภายในและตำแหน่งขาต่างๆของ Max232

ส่วนการที่เราจะนำข้อมูลมาใช้งานก็ต้องแปลงเป็น Parallel ก่อนซึ่งเป็นหน้าที่ของ UART ซึ่งปัจจุบันไมโครคอนโทรลเลอร์มักจะมี serial communication interface (SCI) อยู่ในตัว แต่อาจจะมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

งานบางอย่างที่ไม่ได้ใช้ไมโครคอนโทรลเลอร์และต้องการประมวลผลข้อมูลกับพอร์ตอนุกรม เช่น ต่อ ADC เข้ากับ UART หรือต่อ LCD display เข้ากับ Serial comm. ก็ต้องใช้ UART ช่วย เช่น เบอร์ 8250 หรือ 16550A หรือเบอร์อื่น ๆ ที่ได้กล่าวมาแล้ว แต่มี UART อีกพวกหนึ่งที่แยก Tx bus กับ Rx bus ออกจากกันทำให้มีความคล่องตัวมากขึ้น

2.7 การสื่อสารข้อมูล

การสื่อสารข้อมูลมีจุดมุ่งหมายในการส่ง หรือการถ่ายทอดข่าวสารจากจุดหนึ่งไปยังอีกจุดหนึ่งอย่างถูกต้องเหมือนกับการที่ด้านส่งออกมา ในการสื่อสารรูปแบบใดๆ จะต้องประกอบไปด้วยส่วน ประกอบหลักเบื้องต้น 3 ส่วนดังนี้

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

2.7.1 ประเภทของการสื่อสารข้อมูล

ข้อมูลในระบบของการสื่อสารเขียนแสดงได้ด้วยค่าในระบบเลขฐานสอง โดยใช้ค่าตัวเลข 0 หรือ 1 มาประกอบกันเป็นรหัส แต่ในการส่งเราอาศัยการส่งทางไฟฟ้า ข้อมูลจะถูกเปลี่ยน ให้อยู่ในรูปแบบทางไฟฟ้าโดยใช้ค่าสัญญาณไฟฟ้า 2 ระดับคือสูงและต่ำ ในการวัดอัตราเร็วในการส่ง ได้จากจำนวนบิตที่ส่งไปในหน่วยเวลาโดยทั่วไปใช้หน่วย Bit per second (bps) ซึ่งในระบบการสื่อสาร ข้อมูลนั้นอาจจะมีย่านใดก็ได้แต่ในการส่งนั้นจะต้องมีการกำหนดจุดเริ่มต้น และจุดสิ้นสุดของการแต่ละส่ง ตามปกติจะจัดแบ่งข่าวสารที่จะส่งเป็นบล็อกๆ ซึ่งหนึ่งบล็อกคือกลุ่มของบิตจำนวนหนึ่งที่ถูกส่งออกไปเป็นหน่วยเดียวกัน โดยมีการนำกลุ่มบิตนั้นไปผ่านกระบวนการบางอย่าง เพื่อใช้ในการควบคุมข้อผิดพลาดที่อาจจะเกิดขึ้น

เราสามารถจำแนกวิธีการส่งข้อมูลได้หลายแบบตามคุณสมบัติต่างๆดังนี้

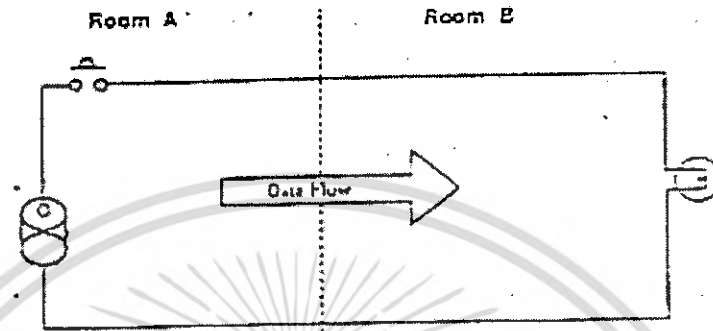
2.7.1.1 การจำแนกตามทิศทางการส่งข้อมูล

สามารถแบ่งการส่งข้อมูลออกได้เป็น 3 ชนิด ดังนี้

2.7.1.1.1 การรับส่งข้อมูลทางเดียว (Simplex)

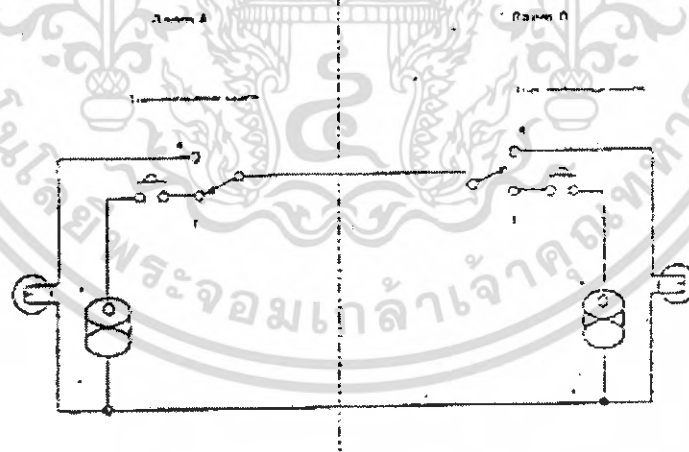
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นการสื่อสารทางเดียว ที่เห็นได้ชัดก็คือ การรับส่งโทรทัศน์ และวิทยุกระจายเสียงนั่นเอง สถานีโทรทัศน์จะเป็นตัวส่งและเครื่องรับทำหน้าที่รับเพียงอย่างเดียว จะส่งข่าวสารมายังสถานีส่งไม่ได้



รูปที่ 2.29 แสดงการรับส่งข้อมูลแบบทางเดียว (Simplex)

2.7.1.1.2 การรับส่งแบบผลัดกันส่ง (Half Duplex) มีคุณสมบัติสามารถรับและส่งข้อมูลได้ แต่จะต้องสลับการส่ง โดยจะส่งพร้อมกันทั้งสองด้านไม่ได้ อุปกรณ์ที่ใช้ในการติดต่อในแบบนี้ ได้แก่ วิทยุสื่อสาร และอินเตอร์คอม



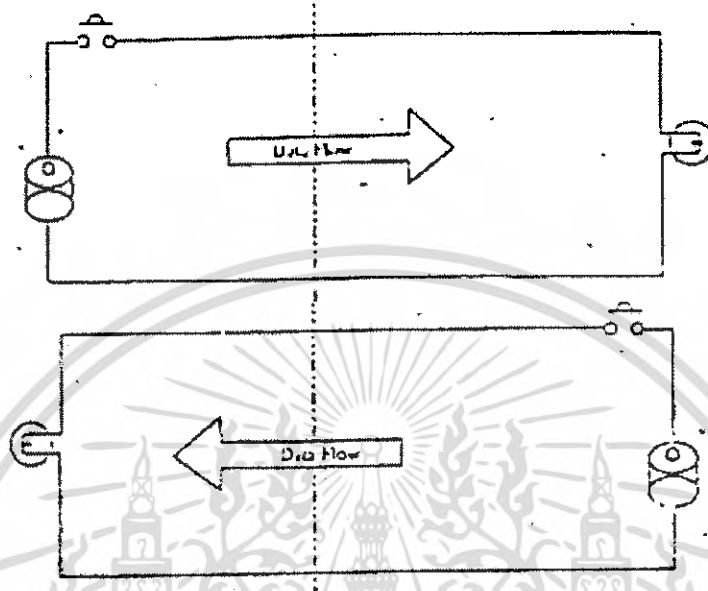
Half-duplex communication systems

รูปที่ 2.30 แสดงการรับส่งข้อมูลสวนทางกันได้แบบผลัดการส่ง(Half Duplex)

2.7.1.1.3 การรับส่งสวนทางได้พร้อมกัน (Full Duplex)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การรับส่งแบบนี้ ผู้รับและผู้ส่ง สามารถรับ และส่งพร้อมๆกันได้ ในเวลาเดียวกันโดยจำเป็นต้องรอให้อีกฝ่ายหนึ่งส่งจบเสียก่อน เช่น การพูดโทรศัพท์ของเรา



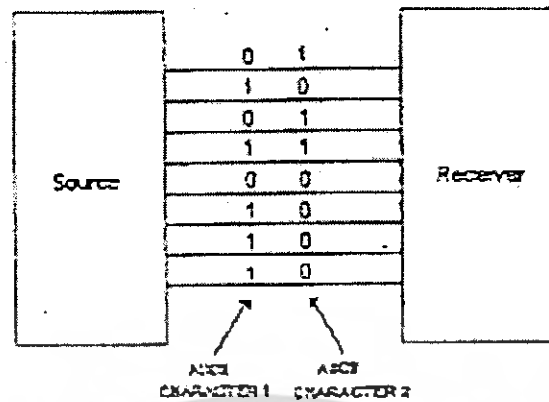
รูปที่ 2.31 แสดงการรับส่งข้อมูลแบบสวนทางกันได้พร้อมกัน (Full Duplex)

2.7.1.2 การจำแนกตามลักษณะการจัดข้อมูล

สามารถแบ่งการส่งข้อมูลออกได้เป็น 2 ชนิดดังนี้

2.7.1.2.1 การส่งข้อมูลแบบขนาน (Parallel Transmission)

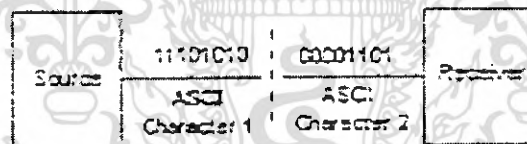
เป็นการส่งข้อมูลที่ละไบนารี คือส่งทุกบิตของรหัสที่ประกอบขึ้นเป็นอักขระไปพร้อมๆ กันในเวลาเดียวกัน มีข้อดีคือ ความสามารถในการส่งข้อมูลที่สูงขึ้น แต่จำนวนช่องทางการสื่อสารที่จำเป็นต้องมีจำนวนเท่ากับจำนวนที่บิตที่ประกอบขึ้นเป็นอักขระ ซึ่งต้อง ใช้การมัลติเพล็กซ์ข้อมูลต่างๆ



รูปที่ 2.32 แสดงการส่งข้อมูลแบบขนาน

2.7.1.2.2 การส่งข้อมูลแบบอนุกรม (Series Transmission)

การส่งข้อมูลแบบนี้จะกระทำทีละบิตด้วยแกนแนลเพียงแกนแนลเดียว ทางด้านรับจะมีอุปกรณ์สำหรับจัดการจัดข้อมูลดังกล่าวเป็นชุดอักขระ ความขัดข้องระหว่างอุปกรณ์ปลายทางทั้งสองที่สื่อสารกัน



รูปที่ 2.33 แสดงการส่งข้อมูลแบบอนุกรม

2.7.1.3 การจำแนกตามความสัมพันธ์ของข้อมูล

ในการสื่อสารข้อมูลนั้นจะต้องพิจารณาถึงความสัมพันธ์ระหว่างข้อมูล 2 ชนิด คือ

- ความสัมพันธ์ระหว่างบิต (Bit Synchronization)

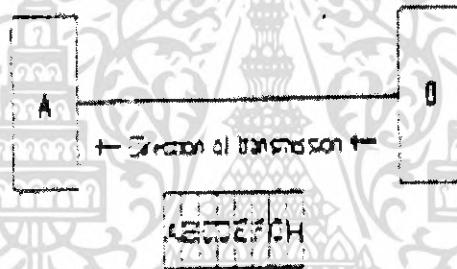
- ความสัมพันธ์ของอักขระ (Character Synchronization)

เอกสารนี้เป็นเอกสารที่ สงวนลิขสิทธิ์ไว้สำหรับนักเรียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการสื่อสารข้อมูลเราจำแนกวิธีการส่งข้อมูลตามการกำหนดความสัมพันธ์ของข้อมูลได้ 2 แบบ คือ

2.7.1.3.1 การส่งแบบสัมพันธ์ (Synchronous Transmission)

การส่งแบบนี้ใช้สำหรับการส่งข้อมูลทั้งหมดไปครั้งเดียวดังรูป 2.33 เทคนิคการส่งแบบนี้ ช่วงเวลาระหว่างบิตต่อบิตจะมีค่าเท่ากัน การส่งมีลักษณะคล้ายกับการส่งข่าวสารในรูปของเลขฐานสองที่มีจำนวนติดต่อกันไป โดยไม่ได้แยกว่าความยาวใดเป็นช่วงอักขระใดในระบบเช่นนี้ บิตแต่ละบิตจะมีความยาวเท่ากัน ตัวอักษรแต่ละตัวมีช่วงเวลาห่างกันเท่ากับศูนย์ ทางด้านรับนั้น เพียงหาว่าบิตแรกของตัวอักษรตัวแรกคือบิตใด และทราบขนาดหรือจำนวนบิตในหนึ่งตัวอักษร พร้อมทั้งความเร็วในการส่งก็สามารถแยกข่าวสารของแต่ละอักษรออกมาได้

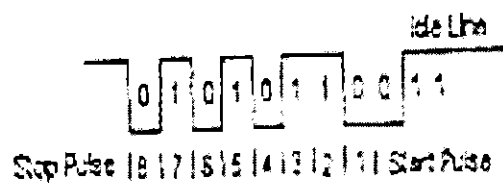


รูปที่ 2.34 แสดงการส่งข้อมูลแบบสัมพันธ์

2.7.1.3.2 การส่งแบบไม่สัมพันธ์ (Asynchronous Transmission)

การส่งแบบนี้ตัวอักขระจะถูกส่งออกไปที่เวลาใดๆ ก็ได้โดยไม่จำเป็นต้องมีความสัมพันธ์ระหว่างตัวอักษรว่าต้องมีเวลาที่แน่นอนอย่างไร ทางด้านรับจะต้องทราบถึงบิตเริ่มต้นของรหัสแต่ละตัวว่าเริ่มต้นเมื่อใดซึ่งเมื่อใดซึ่งสามารถกระทำได้โดยการเพิ่มบิตที่เรียกว่า พัลส์เริ่มต้น (Start Pulse) โดยเติมเข้าไปที่ข้างหน้าสุดของทุกอักขระ และเมื่อการส่งส่งครบทุกบิตของตัวอักษรแล้ว จะต้อง มีบิตสำหรับบอกถึงการสิ้นสุดที่เรียกว่า พัลส์การสิ้นสุด (Stop Pulse) ส่งมาให้ทางด้านรับมีเวลาสำหรับ การเตรียมข้อมูลของตัวอักษรตัวต่อไปดังรูป 2.20 บางครั้งจึงเรียกระบบการส่งแบบนี้ว่า ระบบการส่งแบบเริ่มหยุด (Start-stop Transmission)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.35 แสดงการส่งข้อมูลแบบ ไม่สัมพันธ์

2.8 ดีซีมอเตอร์ (DC Motor)

2.8.1 หลักการทำงานของดีซีมอเตอร์

ดีซีมอเตอร์เป็นทรานส์ดิวเซอร์แรงบิดซึ่งมีการออกแบบให้มีคุณลักษณะพิเศษ คือ แรงบิดของเพลลาของมอเตอร์จะเป็นสัดส่วนโดยตรงกับกระแสอาร์เมเจอร์ ความสัมพันธ์ระหว่างแรงบิดของเพลลาและกระแสคือ

$$T = K \Phi I$$

เมื่อ T คือแรงบิดของเพลลา มีหน่วยเป็น นิวตัน-เมตร

Φ คือเส้นแรงแม่เหล็ก มีหน่วยเป็น เวเบอร์

I คือกระแส มีหน่วยเป็น แอมแปร์

K คือค่าคงตัว

จะเห็นได้ว่าแรงบิดของเพลลาจะเป็นสัดส่วน โดยตรงกับผลคูณของเส้นแรงแม่เหล็กและกระแส การที่ขดลวดตัวนำเคลื่อนที่ในสนามแม่เหล็กจะทำให้เกิดแรงดันตกคร่อมตัวมันเองโดยแรงดันนี้จะเป็นสัดส่วนกับความเร็วของเพลลาของมอเตอร์และเส้นแรงแม่เหล็ก โดยมีความสัมพันธ์ดังนี้

$$E = K \Phi \omega$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เมื่อ E คือแรงดันย้อนกลับมีหน่วยเป็น โวลท์
 Φ คือเส้นแรงแม่เหล็กมีหน่วยเป็น เวเบอร์
 ω คือความเร็วของมอเตอร์มีหน่วยเป็น เรเดียน/วินาที
 K คือค่าคงตัว

2.8.2 ดีซีมอเตอร์แบบเส้นแรงแม่เหล็กคงที่

ฟิลด์ (Field) ของมอเตอร์ชนิดนี้นิยมใช้แม่เหล็กถาวร มอเตอร์ชนิดนี้จึงอาจจะเรียกว่า มอเตอร์แบบแม่เหล็กถาวร (Permanent-Magnet Motors) มอเตอร์แบบแม่เหล็กถาวรนี้ไม่ต้องใช้ การกระตุ้นเพื่อให้เกิดสนามแม่เหล็ก เนื่องจากมีแกนขั้ว (Core) ทำด้วยอัลลอยซึ่งสามารถรักษา คุณสมบัติความเป็นแม่เหล็กหรือขดลวดฟิลด์ มอเตอร์ชนิดนี้จึงมีประสิทธิภาพสูงกว่ามอเตอร์ ชนิดซันท์หรือมอเตอร์คอมปาวด์ที่มีพิคัดเท่ากันนอกจากนั้นยังควบคุมความเร็วและทิศทางการ หมุนได้ง่ายโดยการกำหนด ขนาดและขั้วแรงดันที่ป้อนให้แก่มอเตอร์ ซึ่งหมายความว่าถ้ากลับขั้ว ของแรงดันจะทำให้ทิศทางของกระแสที่อาร์เมเจอร์เปลี่ยนแปลงไปเพียงอย่างเดียวเท่านั้น แต่ไม่ ทำให้ฟลักซ์เปลี่ยนแปลงจึงได้อัตรา ส่วนระหว่างกระแสอาร์เมเจอร์และแรงบิดมีค่าคงที่ ดังมี ความสำคัญดังนี้

$$T=K\Phi I$$

$$E=K_e\omega$$

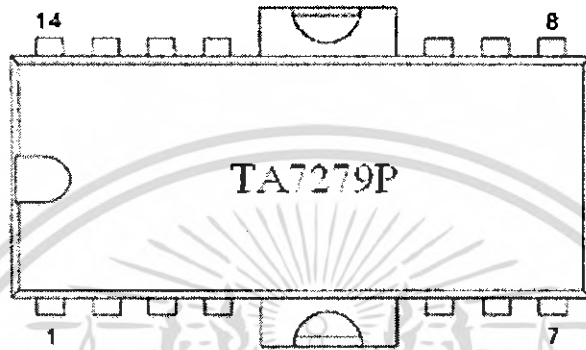
ระบบนี้จะให้ความสัมพันธ์ระหว่างกระแสอาร์เมเจอร์ , แรงบิด และความเร็วอยู่ใน ลักษณะเชิงเส้น สมการทางไฟฟ้าของดีซีมอเตอร์แบบนี้เขียนได้เป็น

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

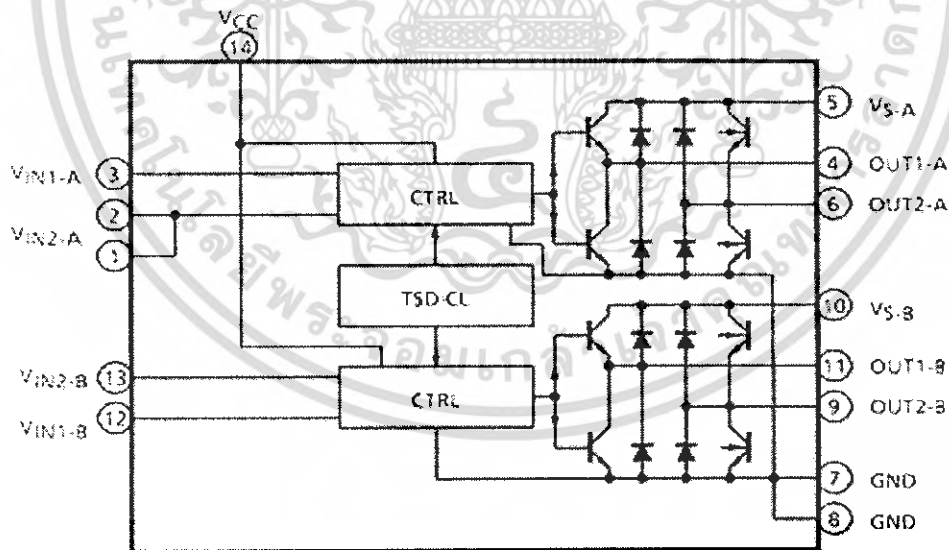
- เมื่อ V คือแรงดันที่ป้อนให้มอเตอร์
 K_e คือค่าคงที่ของแรงดันย้อนกลับ
 L คือค่าความเหนี่ยวนำของขดอาร์เมเจอร์
 R คือความต้านทานที่ขั้วของมอเตอร์

2.9 วงจร Drive Motor

IC เบอร์ TA7279P ซึ่งเป็น IC Drive Motor แบบ DUAL BRIDGE DRIVER ซึ่งมีทั้งหมด 16Pin



รูปที่ 2.36 แสดงลักษณะ Pin ของ IC เบอร์ TA7279P



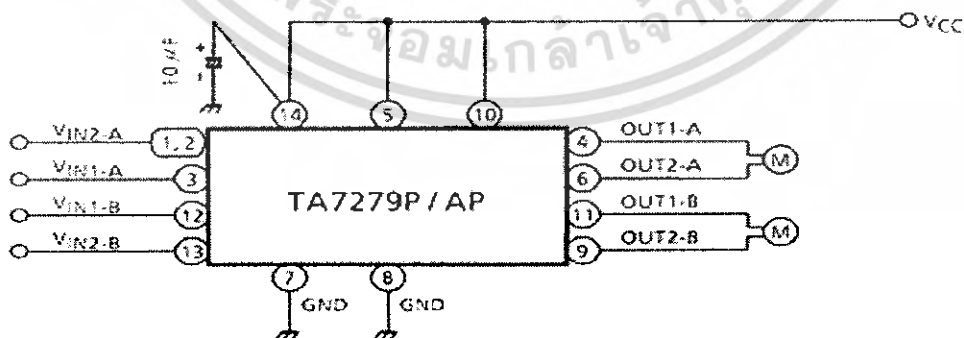
รูปที่ 2.37 แสดง Block Diagram ของ IC TA7279P

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.4 ตารางแสดงคำอธิบายของ Pin IC เบอร์ TA9297P

PIN No.	SYMBOL	FUNTIONAL DESCRIPTION
1	Vin2 - A	A - ch input terminal
2	Vin2 - A	
3	Vin1 - A	A - ch input terminal
4	OUT1 - A	A - ch Motor drive power
5	Vs - A	A - ch input terminal
6	OUT2 - A	A - ch input terminal
7	GND	GND terminal
8	GND	
9	OUT - B	B - ch input terminal
10	Vs-B	B - ch Motor drive power
11	OUT1 - B	B - ch input terminal
12	Vin1 - B	B - ch input terminal
13	Vin2 - B	B - ch input terminal
14	VCC	Logic power supply

จากตารางที่ 2.4 และ รูปที่ 2.37 ทำให้เห็น โครงสร้างและการทำงานทั้งหมดของ IC เบอร์ TA7279P ได้ จะทำให้สามารถเข้าใจ การนำไปใช้งาน



รูปที่ 2.38 แสดงการทำงานของ IC เบอร์ TA7279P กับมอเตอร์ชนิด DC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.20 เป็นการแสดงการทำงานของ IC โดยนำเอาต์พุตของแต่ละ Chanel ไปต่อกับมอเตอร์ที่จะควบคุมการหมุนคือ M1 และ M2 โดยมี Supply จ่ายกระแสเพื่อควบคุมการทำงาน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

หลักการประมวลผลภาพดิจิทัล

3.1 รูปภาพดิจิทัล

รูปภาพนั้นสามารถนิยามได้ด้วย Two dimension function(x, y) โดยที่ x และ y เป็นจุดใดๆ บนระนาบ และค่าของ F ที่จุด (x, y) ใดๆ ก็คือความเข้มสี (intensity) หรือระดับเทาของจุดนั้นๆ นั้นเอง ถ้าค่า x, y และ f ทั้งหมดนี้เป็นค่าที่จำกัด เราเรียกภาพนี้ว่า digital image โดยจะเรียกจุด (x, y) ใดๆ ว่า picture element, image element, pals และ pixel ซึ่งส่วนใหญ่แล้วจะเรียกว่า pixel นั้นเอง

3.2 การปรับปรุงภาพ

วัตถุประสงค์หลักในการปรับปรุงภาพก็คือ การประมวลผลภาพให้ภาพที่ได้ออกมาใหม่นั้นเหมาะสมกับการนำไปใช้มากขึ้น ซึ่งการปรับปรุงภาพนี้ไม่มีรูปแบบที่แน่นอนว่าแบบไหนดีที่สุด ทั้งนี้การตัดสินใจว่าควรใช้วิธีใดจึงขึ้นอยู่กับว่า จะนำภาพไปใช้ทำอะไร ในแอปพลิเคชันนั้นๆ

วิธีการปรับปรุงภาพแบ่งออกเป็น 2 ประเภทคือ

3.2.1 วิธีสเปเชียลโดเมน (Spatial Domain Method) คือ การประมวลผลค่าที่อยู่ในแต่ละพิกเซลนั้น โดยตรง

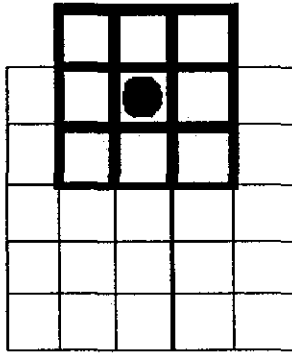
3.2.2 วิธีฟริควเอนซีโดเมน (Frequency Domain Method) คือ การประมวลผลกับภาพที่ถูกแปลงโดยวิธี ฟูเรียทรานส์ฟอร์ม (Fourier Transform) ก่อน

3.2.1.1 วิธีสเปเชียลโดเมน

กระบวนการที่ทำ Spatial Domain สามารถแทนด้วยสมการทางคณิตศาสตร์ดังนี้

$$g(x,y)=T(f(x,y))$$

โดยที่ $f(x,y)$ คือรูปภาพที่รับเข้ามา $g(x,y)$ เป็นภาพที่ได้ประมวลผลแล้ว และ T คือ กระบวนการที่ทำกับ f โดยทำกับจุดข้างเคียงของ (x,y) ด้วยการที่จะนำจุดข้างเคียงมาประมวลผลด้วย สามารถทำได้โดยใช้ภาพสี่เหลี่ยม หรืออาจจะเรียกว่า หน้ากาก(mark) หรือหน้าต่าง(window) มาครอบกับภาพที่รับเข้ามา และให้จุดกึ่งกลางของหน้าต่างอยู่ที่จุด (x,y) ดังรูป



รูปที่ 3.1 แสดงการใช้หน้าต่างครอบภาพที่รับเข้ามา

การนำหน้าต่างมาครอบที่ละจุดนี้ จะมีการคำนวณค่าทั้งหมดที่อยู่ในหน้าต่างแล้ว แทนที่ลงไป ณ จุดนั้น จากนั้นจะเลื่อนหน้าต่างแล้วคำนวณแบบเดิมไปเรื่อยๆ วิธีการที่ง่ายที่สุดในการสร้าง T คือใช้หน้าต่างขนาด IXI นั่นก็คือหนึ่งพิกเซลนั่นเอง ในกรณีนี้ g จะขึ้นกับค่าของภาพที่รับมาที่จุด (x,y) นั่นเอง

3.2.1.2 Histogram Equalization

Histogram ของภาพดิจิทัลที่เป็นระดับเทา (gray level) ในช่วง $[0, L-1]$ คือฟังก์ชันที่เป็น discrete ซึ่งสามารถแทนสมการทางคณิตศาสตร์ดังนี้

$$h(r_k) = n_k \quad \text{โดยที่ } k=[0,L-1]$$

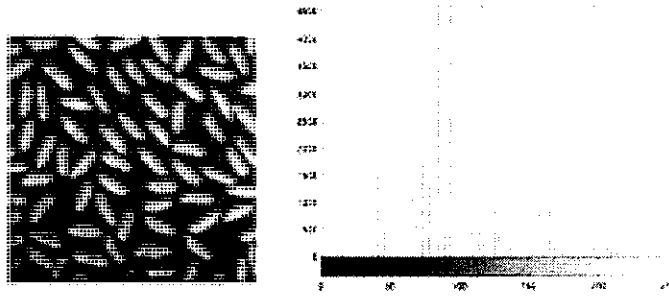
ค่า r_k คือระดับเทาที่ k

ค่า n_k คือจำนวนพิกเซลในภาพที่มีระดับเทาเท่ากับ r_k

ปกติจะมีการ normalized histogram ด้วยการหารแต่ละค่าด้วยจำนวนพิกเซลทั้งหมดในภาพ ดังนั้น histogram ที่ถูก normalized จะเขียนเป็นสมการได้ดังนี้

$$p(r_k) = n_k/n \quad \text{โดยที่ } k=[0,L-1]$$

หรืออาจกล่าวได้ว่า $p(r_k)$ คือประมาณของความน่าจะเป็นของระดับเทาที่ r_k ที่ในภาพสังเกตว่าผลรวมของทุกค่าใน histogram ที่ถูก normalized จะมีค่าเท่ากับ 1



รูปที่ 3.2 แสดงภาพต้นฉบับและ Histogram ที่ไม่ถูก normalized

Histogram Equalization

สามารถอธิบายได้ดังสมการ

$$S_k = \sum_{j=0}^k p(r_j) \quad \text{ตั้งแต่ } j=0 \text{ ถึง } k \text{ โดยที่ } k=0,1,\dots,L-1$$

$$= \sum_{j=0}^k (n_j/n) \quad \text{ตั้งแต่ } j=0 \text{ ถึง } k \text{ โดยที่ } k=0,1,\dots,L-1$$

วิธี Histogram equalization นี้จะช่วยกระจาย histogram ของภาพ input ทำให้ภาพผลลัพธ์ที่ได้ออกมานั้นกระจายระดับเทาในช่วงที่กว้างขึ้น

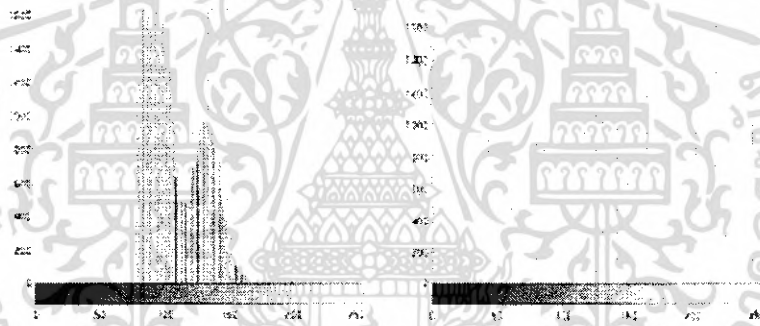


รูปที่ 3.3 ภาพต้นฉบับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 ผ่านการทำ Histogram equalization



รูปที่ 3.5 แสดง Histogram ของภาพที่3.3และภาพที่3.4

3.2.1.3 Averaging Filters

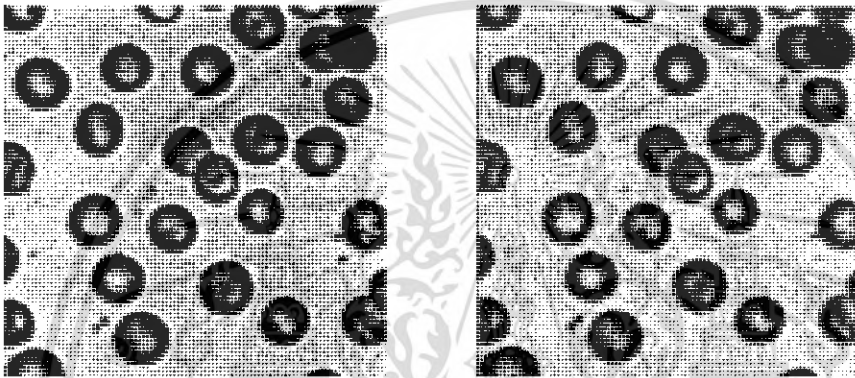
ใช้สำหรับทำให้ภาพเบลอและสำหรับลบสัญญาณรบกวนออก โดยผลลัพธ์ที่ได้จากการใช้ Averaging filters ในแต่ละจุดก็คือ ค่าเฉลี่ยของพิกเซลทุกพิกเซลที่อยู่ในหน้าต่างของ filters นั้นๆ เช่น การใช้ filters ขนาด 3x3 ดังภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4	7	2		
9	5	3		
8	12	46		

รูปที่ 3.6 แสดงการทำ averaging filters

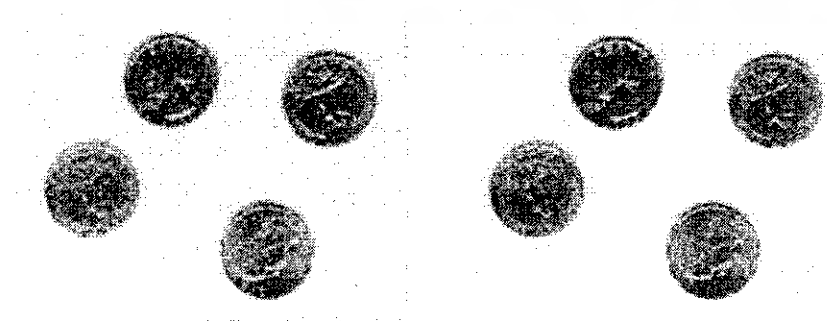
จะได้ค่าที่จุดกึ่งกลางเป็น $(4+7+2+9+5+3+8+12+46)/9 = 10.67$



รูปที่ 3.7 แสดงรูปต้นฉบับและรูปที่ผ่าน Averaging filters

3.2.1.4 Median filters

3.2.1.5 ผลลัพธ์ที่ได้จากการทำ Median filters ก็คือ การแทนที่ค่าในพิกเซลทั้งหมดที่หน้าตาครอบอยู่ด้วยค่ามัธยฐานนั่นเอง จากรูปที่ 3.8 ถ้าจะประมวลผลกับภาพด้วย median filters จะทำได้โดย เรียงค่าน้อยไปมากนั้นก็คือ 2,3,4,5,7,8,9,12 และ 46 เพราะฉะนั้น ค่ามัธยฐานก็คือ 7 ภาพที่ใช้ median filters แสดงดังตัวอย่างข้างล่างนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.8 แสดงภาพต้นฉบับและภาพหลังจากการทำ Median filters

3.2.2.1 วิธีฟรีแควนซีโดเมน

การปรับปรุงภาพด้วยวิธีนี้จะต้องมีการแปลงโดย Fourier Transform ซึ่งใช้เวลานานมากในการประมวลผล

3.3 พิกเซล (Pixel)

ภาพจะถูกเปลี่ยนไปเป็นสัญญาณทางไฟฟ้าแล้วไปเก็บไว้ในรูปของจุดเล็กๆ ที่เรียงต่อเนื่องกันทำให้เกิดการเรียงตัวกันจนเป็นภาพให้เราได้มองเห็น จุดเล็กๆนี้ ถูกเรียกว่า พิกเซล มีการจัดเรียงเป็นแถวและคอลัมน์ ซึ่งตำแหน่งแต่ละตำแหน่งจะแทนด้วย $P(i, j)$ โดยที่ i และ j จะเป็นเลขจำนวนเต็มและเรียกการจัดเรียงนี้ว่า พิกเซลเมตริกซ์ (pixel matrix)

เมื่อรู้ตำแหน่งแล้วก็ต้องรู้ว่าตำแหน่งนั้นมีค่าสีเท่าไรในแต่ละตำแหน่งพิกเซล ค่าสีนี้คือค่าที่เรียงกันจนทำให้สามารถมองเห็นเป็นภาพได้

3.4 รูปแบบข้อมูล

ภาพ 2 ระดับ คือ ภาพที่มีแต่ขาว กับ ดำ โดยในแต่ละจุดของภาพนั้น จะมีข้อมูลเป็น 1บิต ภาพ 16 ระดับ คือ ภาพที่แต่ละจุดจะมีข้อมูล เป็น 4บิต สามารถแสดงภาพได้ 16ระดับความแตกต่างของสี หรือ 16ระดับสีเทา ขึ้นอยู่กับว่าภาพนั้นเป็นภาพสี หรือ ภาพขาว-ดำ

ภาพ 256 ระดับ คือ ภาพที่มีข้อมูลขนาด 8 บิต ซึ่งก็ทำให้สามารถแสดงผลความแตกต่างของภาพได้ถึง 256ระดับสี หรือ ก็คือ 256ของระดับเทา

ภาพ True color คือ ภาพที่มีข้อมูลขนาด 24 บิต ทำให้สามารถแสดงผลของภาพได้เหมือนจริงมากที่สุด เพราะความสามารถในการแสดงสีถึง 16,777,216 สี ได้เฉพาะภาพสีเท่านั้น

3.5 เกรย์สเกล

เกรย์สเกล หมายถึง ความแตกต่างของระดับความเข้มของแสง โดยเกรย์สเกลหนึ่งๆอาจแบ่ง 13, 20 หรือ 9 ระดับ โดยระดับที่ว่านี้ก็คือ ระดับสีเทา ในภาพหนึ่งๆถ้าต้องการแบ่งระดับความเข้มแสงหรือระดับสีเทาให้มีหลายๆค่า นั้น จำเป็นอย่างยิ่งที่จะต้องเพิ่มจำนวนบิตที่แสดงค่ากับพิกเซล ตัวอย่างเช่นถ้าต้องการภาพที่มีระดับสีเทา 4 ระดับ ต้องแทนด้วยเลขฐาน 2 บิตถ้าต้องการภาพที่มีระดับสีเทา 16 ระดับ ต้องแทนด้วยเลขฐานจำนวน 4บิต และถ้าต้องการภาพที่มีระดับสีเทา 256ระดับ ต้องแทนด้วยเลขฐานสองจำนวน 8 บิต เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำนวนระดับสีเทาที่ต้องการนี้ก็คือค่ายกกำลังของ 2 นั้นเอง ซึ่งค่าต่ำสุดหรือ 0 จะแทนสีดำ หรือ ไม่มีความสว่างเลยและค่าที่มากที่สุดก็คือค่าที่น้อยกว่าจำนวนระดับสีเทาอยู่เช่นค่า 15 ในที่มีระดับสีเทา 16 ระดับก็จะเป็นสีขาวหรือสว่างมากที่สุดเป็นต้น

ในยุคแรกๆระบบการมองเห็น(Vision system) จะใช้ระบบเลขฐานสองเพราะสะดวกต่อการนำเซนเซอร์มาใช้ นอกจากนี้การรวบรวมข้อมูล การเก็บรักษาข้อมูลยังสามารถทำได้ง่ายอีกด้วย

ในปัจจุบัน ไมโครโปรเซสเซอร์ที่ใช้กันโดยทั่วไป ขนาดที่เล็กที่สุดก็คือ 8 บิต ดังนั้นเกรย์สเกลขนาด 8 ค่า, 16 ค่า และ 256 ค่า จึงไม่เป็นปัญหาในการประมวลผล

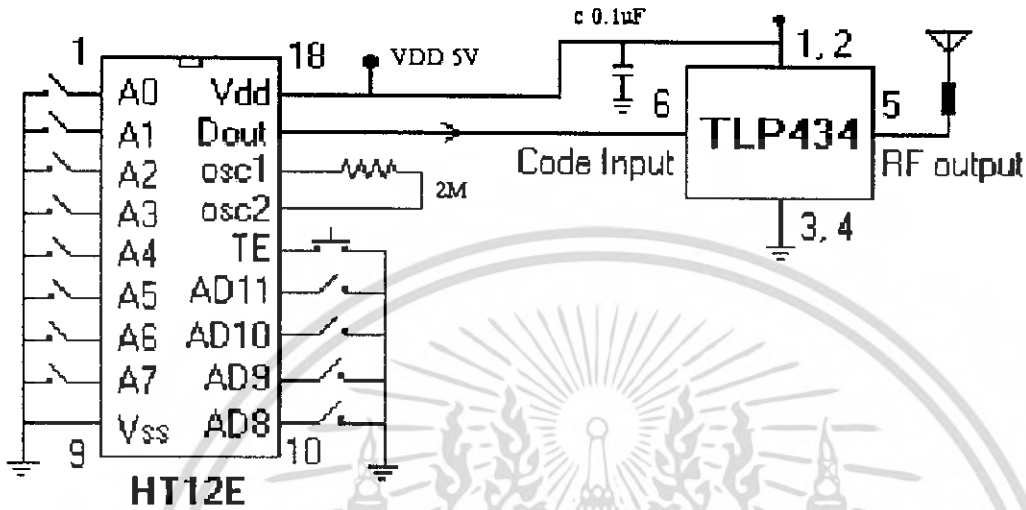
ความสามารถในการแบ่งแยกระดับความแตกต่างของสายตามนุษย์ โดยทั่วไปจะอยู่ระหว่าง 10 ถึง 15 ระดับ เกรย์สเกลขนาด 16 ระดับจึงถือได้ว่าใกล้เคียงกับสายตามนุษย์หรือ อาจจะละเอียดน้อยกว่าสายตามนุษย์เล็กน้อย(ในบางคน) ในขณะที่เกรย์สเกลขนาด 64 หรือ 256 นั้นละเอียดเกินไปสำหรับตามนุษย์การแสดงผลนี้ใช้วิธีตั้งค่าของแม่สีในตารางสี โดยอาจเลือกสีเป็นแบบ 16 สี จาก 64 สี หรือ 16 สี จาก 262,144 สี หรือ 256 สี จาก 262,144 สีขึ้นอยู่กับโหมดการแสดงผลสำหรับทรูคัลเลอร์จะไม่มีการเลือกสี แสดงผลโดยการส่งค่าสี RGB ผ่าน D/A สีละ 8 บิต ออกไปความแตกต่างของงานแสดงภาพสีและขาวดำคือ ภาพขาวดำจะต้องตั้งให้แม่สีทั้งสามสีมีค่าเท่ากันเนื่องจาก VGA กำหนดให้แม่สีแต่ละสีใช้ได้เพียง 64 ระดับสีเท่านั้นหากต้องการให้เห็นทั้ง 256 ระดับ ต้องแสดงในโหมดทรูคัลเลอร์ แล้วให้ RGB มีค่าเท่ากัน ซึ่ง จะสามารถใช้จีบิตเตอร์ 8 บิต สำหรับแต่ละแม่สี

โดยทั่วไปวิธีการประมวลผลภาพเชิงตัวเลขที่ทำให้คอมพิวเตอร์ สามารถรู้จักวัตถุภาพในภาพได้นั้นแบ่งได้สองระดับด้วยกันคือ การประมวลผลภาพในระดับต่ำ(Low level image processing) และการประมวลผลเชิงตัวเลขเกือบทั้งหมด เพื่อหาค่าตัวแปรต่างๆมาอธิบายข้อมูลภาพ โดยมีจุดประสงค์เพื่อนำตัวแปรเหล่านั้นไปใช้ในการประมวลผลระดับสูงต่อไป

การประมวลผลภาพในระดับสูง เป็นการนำผลลัพธ์ที่ได้จากการประมวลผลระดับต่ำมาตีความหรือเพื่อให้คอมพิวเตอร์ สามารถรู้จักและเข้าใจภาพได้สำหรับความแตกต่างของการประมวลผลภาพระดับต่ำและสูง คือ ข้อมูลที่นำมาใช้ในการประมวลผลโดยการประมวลผลภาพระดับต่ำจะใช้ค่าความสว่างของ จุดโดยตรงส่วนการประมวลผลภาพระดับสูงนั้นข้อมูลภาพที่นำมาประมวลผล ผลจะถูกแสดงในรูปลักษณะซึ่งจะแสดงถึงสิ่งต่างๆที่อยู่ในภาพ เช่น ขนาด หรือ รูปร่างของวัตถุในภาพ

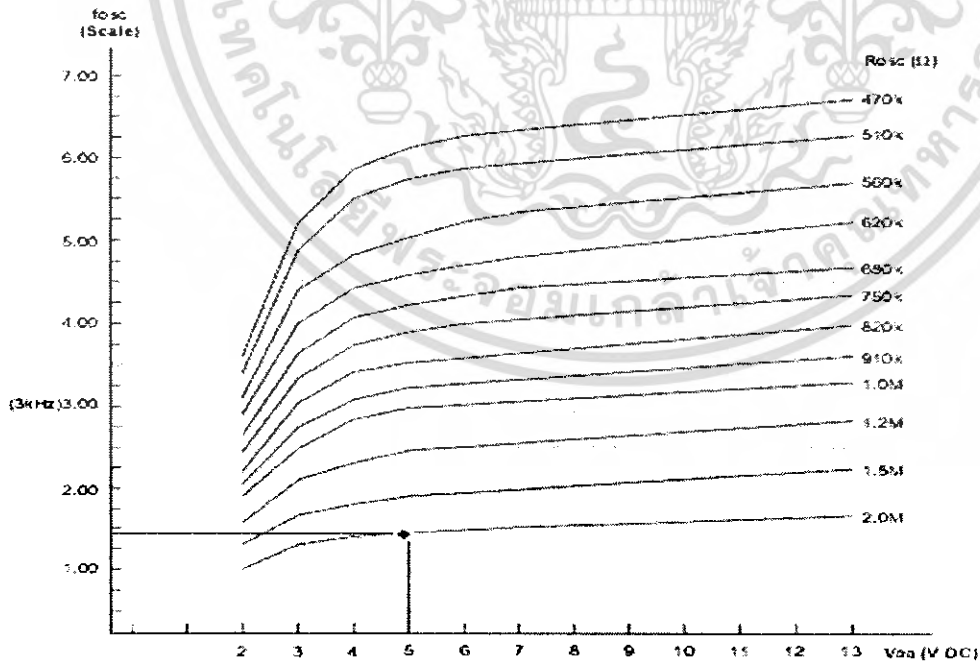
บทที่ 4

ผลการทดลอง



รูปที่ 4.1 ภาคล่งในการใช้งานจริง

จากรูปที่ 4.1 ได้ทดลองชุดส่งสัญญาณความถี่วิทยุเพื่อนำไปใช้งานจริง จากกราฟหาค่าความต้านทานเพื่อให้ HT12E ซึ่งเป็น IC เข้ารหัส เพื่อให้ได้ความถี่ที่เราจะส่ง



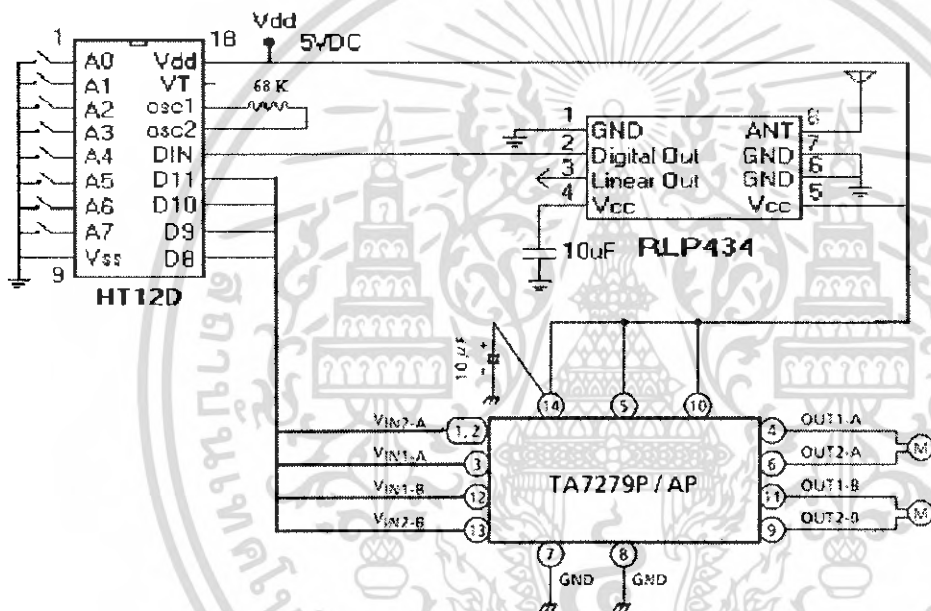
รูปที่ 4.2 เป็นการแสดงการหาค่าความต้านทาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลองที่ 1

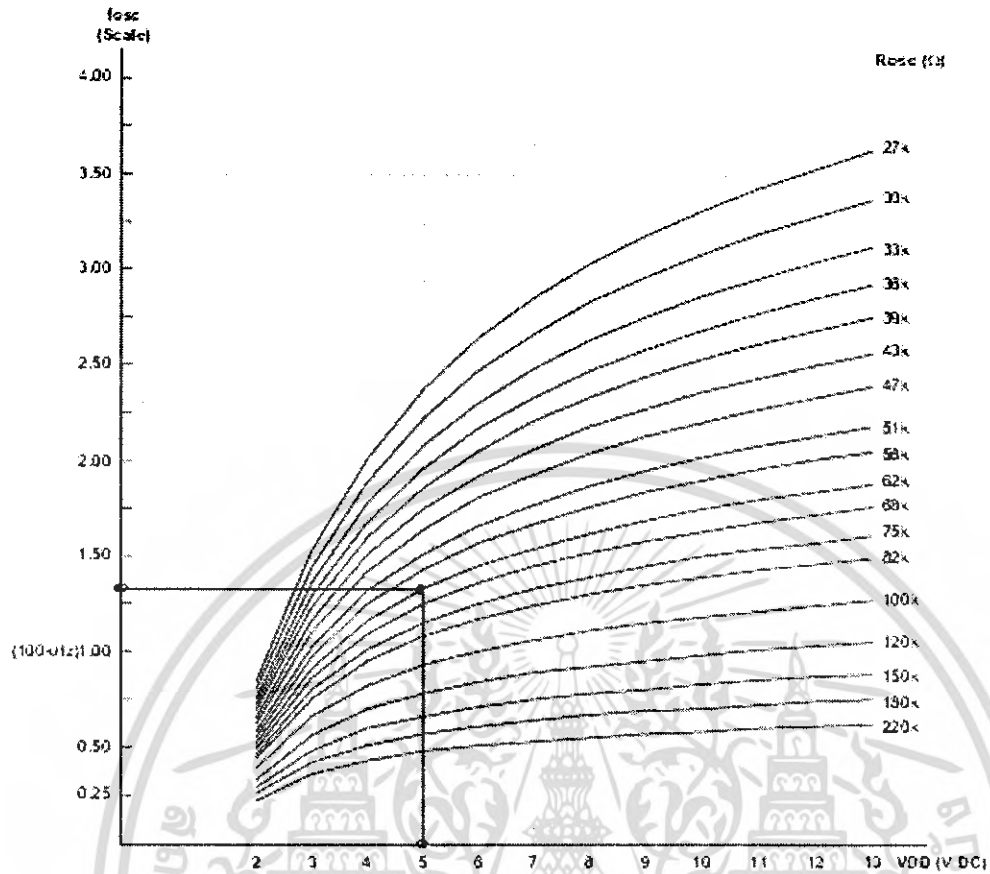
หาความถี่และค่าความต้านทาน จากรูปที่ 4.2 เป็นการแสดงการทดลองเพื่อหาค่าความต้านทานโดยการอ่านค่าจากกราฟ วงจรที่ออกแบบไว้ใช้แหล่งจ่าย 5 V จึงต้องเริ่มจากการลากเส้นตรงจากกราฟ VDD ที่ 5 V ขึ้นไปชนเส้นกราฟไปยังเส้นความต้านทานเมื่อตัดกับเส้นความต้านทานก็ลากจากเส้นความต้านทานผลคือ ได้ค่าความต้านทานที่ 2M โอห์ม และค่าความต้านทานที่ 2M จะได้ความถี่ประมาณ 1.48 KHz และค่าความถี่นี้จะนำไปหาความถี่และค่าความต้านทานที่ภาครับเพื่อให้ได้ค่าความถี่เข้าระหัดเดียวกัน

หลังจากนั้นจะได้ความถี่มาเข้าระหัดแล้วส่งสัญญาณแบบไร้สายผ่าน TLP434 ซึ่งเป็นตัวส่งสัญญาณแบบ Wireless



รูปที่ 4.3 ชุดควบคุม Robot ทั้งหมดเมื่อใช้งานจริง

จากรูปที่ 4.3 เป็นชุดควบคุมการทำงานของหุ่นยนต์ทั้งหมด เมื่อนำไปใช้งานจริง ต้องหา ค่าความต้านให้ตรงกับค่าความถี่เข้าระหัดที่ส่งมาให้ถูกต้อง จากกราฟ เพื่อทำการหาจุดตัดกราฟ ลักษณะเดียวกับภาคส่ง เพื่อให้ IC HT12D สามารถถอดรหัสที่ส่งมาได้



รูปที่ 4.4 เป็นการแสดงการหาค่าความต้านทาน

ผลการทดลองที่ 2

จากหาความถี่และค่าความต้านทาน จากรูปที่ 4.4 เป็นการแสดงการทดลองเพื่อหาค่าความต้านทานโดยการอ่านค่าจากกราฟ วงจรที่ออกแบบไว้ใช้แหล่งจ่าย 5 V จึงต้องเริ่มจากการลากเส้นตรงจากกราฟ VDD ที่ 5 V ขึ้นไปชนเส้นกราฟไปยังเส้นความถี่ความต้านทานเมื่อตัดกับเส้นความถี่ความต้านทานก็ลากจากเส้นความถี่ความต้านทานผลคือ ได้ค่าความต้านทานที่ 68K โอห์ม และความถี่ ประมาณ 1.40 KHz ซึ่งจากการทดลองพบว่าใกล้เคียงที่สุดและสามารถนำไปใช้งานจริงได้และHT12D สามารถถอดรหัสออกมาได้

จากรูปที่ 4.1 ได้ทำการทดลองส่งข้อมูลแบบการเข้ารหัสบิตจากในแต่ละบิตเพื่อทดลองทำการควบคุมการทำงานของหุ่นยนต์ โดยบิตที่เข้ารหัสและส่งจะไปทำการควบคุมการทำงานของมอเตอร์ให้ทำงานเพื่อควบคุมหุ่นยนต์เดินหน้าถอยหลังเลี้ยวซ้ายและเลี้ยวขวาซึ่งผลที่ได้เป็นไปตามตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 แสดงการทำงานของหุ่นยนต์แบบ Binary

ทิศทาง \ ข้อมูลบิต	Data bit 0	Data bit 1	Data bit 2	Data bit 3
หยุด	0	0	0	0
เดินหน้า	1	0	0	1
ถอยหลัง	0	1	1	0
หมุนรอบซ้าย	1	0	0	0
หมุนรอบขวา	0	0	0	1

ผลการทดลองที่ 3

จากการทดลองรถสามารถทำงานได้จริง ซึ่งรถสามารถเดินได้ตามตารางที่ได้บันทึกค่าการทำงานไว้ ซึ่งการทดลองนี้เป็นการป้อนค่าการทำงานให้กับหุ่นยนต์โดยตรง และผลที่ได้เป็นไปตามตารางที่ 4.1

ตารางที่ 4.2 แสดงคำสั่งต่างๆที่สั่งให้หุ่นยนต์ทำงาน

รหัสคำสั่งฐาน 16	ลักษณะการทำงานของคำสั่ง
01H	สั่งให้หุ่นยนต์เดินหน้า
02H	สั่งให้หุ่นยนต์เลี้ยวซ้าย
03H	สั่งให้หุ่นยนต์เลี้ยวขวา
04H	สั่งให้หุ่นยนต์ถอยหลัง
05H	สั่งให้หุ่นยนต์ยิงลูกบอล

ความเร็วในการส่งข้อมูลผ่านระบบไร้สาย

4 บิต ต่อ 200 m sec

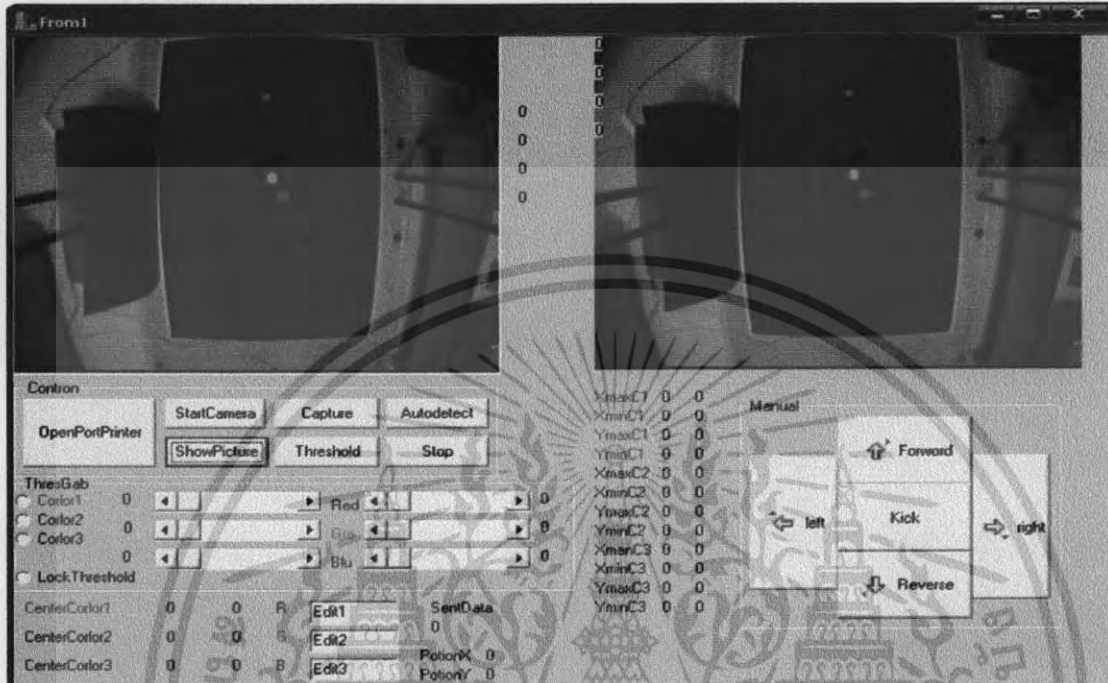
ผลการทดลองที่ 4

เป็นการนำโปรแกรม C++Builder5 ประมวลผลภาพเชิงตัวเลขแล้ว โดยการแยกสีต่างๆที่ต้องการออกจากภาพ โดยวิธีการทำ Threshold เพื่อแยกระดับสีภาพที่เราต้องการสีที่อยู่นอกจากสีที่เราต้องการจะถูกแทนที่ด้วยสีดำ ส่วนสีที่สนใจจะถูกเฉลี่ยแล้วแทนค่านั้นกลับไปยังตำแหน่งสีเดิมที่อ่านออกมา โดยสีที่เราสนใจจะมีทั้งหมด 3 สี สีแรกคือสีเขียวสีที่สองคือสีแดงและสีที่สามคือสีน้ำเงิน จากนั้นเราจะนำเอาค่าสีที่ได้มาอ่านพิกัดของสีเพื่อให้ได้ค่าสูงสุด ค่าต่ำสุดทั้งในแนวนอน xy

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

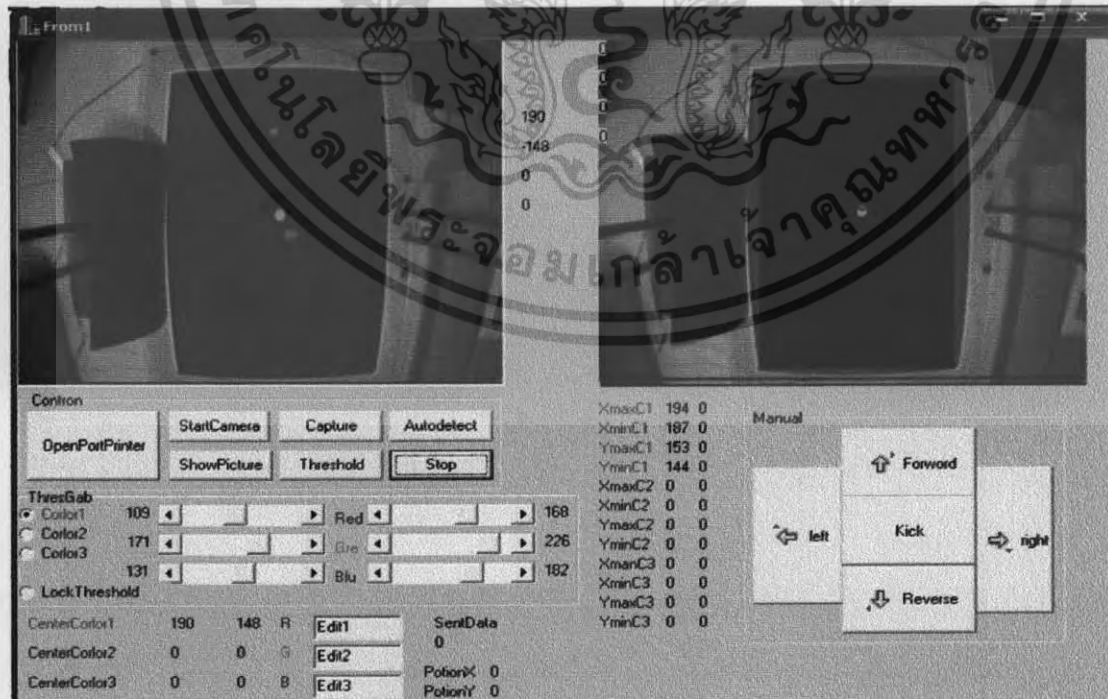
ของแต่ละสี แล้วนำมาคำนวณหาจุดศูนย์กลางของสี แต่ละสี เมื่อได้จุดศูนย์กลางของแต่ละสีแล้วจะทำการหาค่ามุมต่างของสี เขียว สีน้ำเงินโดยเทียบกับสีแดง สีแดงจะถูกกำหนดให้เป็นแกนอ้างอิง ตลอดเมื่อคำนวณหามุมและระยะทาง

4.1 การแสดงภาพทดลองผ่านหน้าจอคอมพิวเตอร์



รูปที่ 4.5 แสดงภาพหน้าต่างโปรแกรม

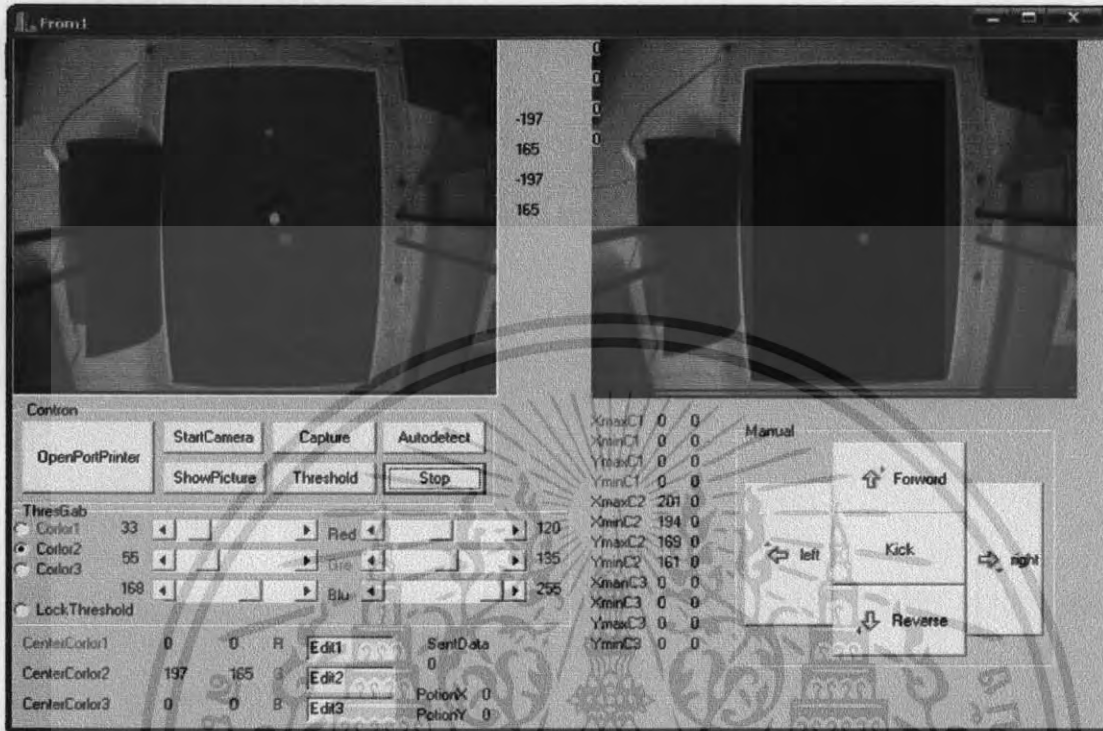
4.2 แสดงการทดลองทำ Threshold ของสีเขียว



รูปที่ 4.6 แสดงการทำ Threshold สีเขียว

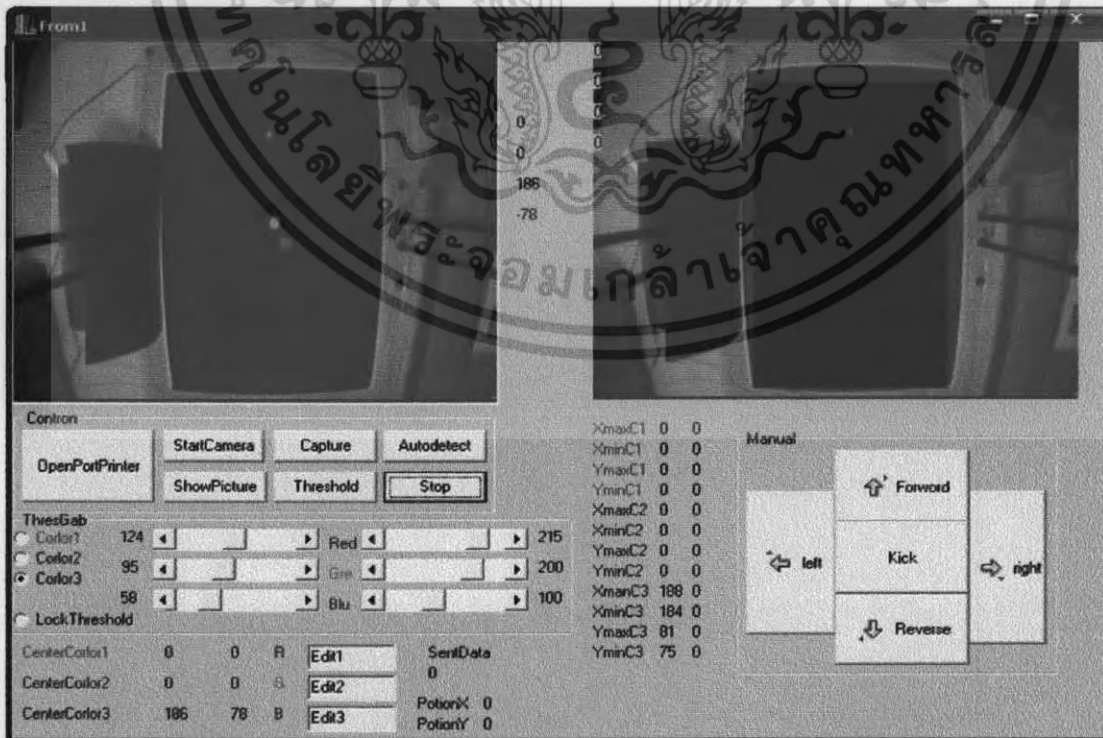
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 แสดงการทดลองทำ Threshold สีแดง



รูปที่ 4.7 แสดงการทำ Threshold สีแดง

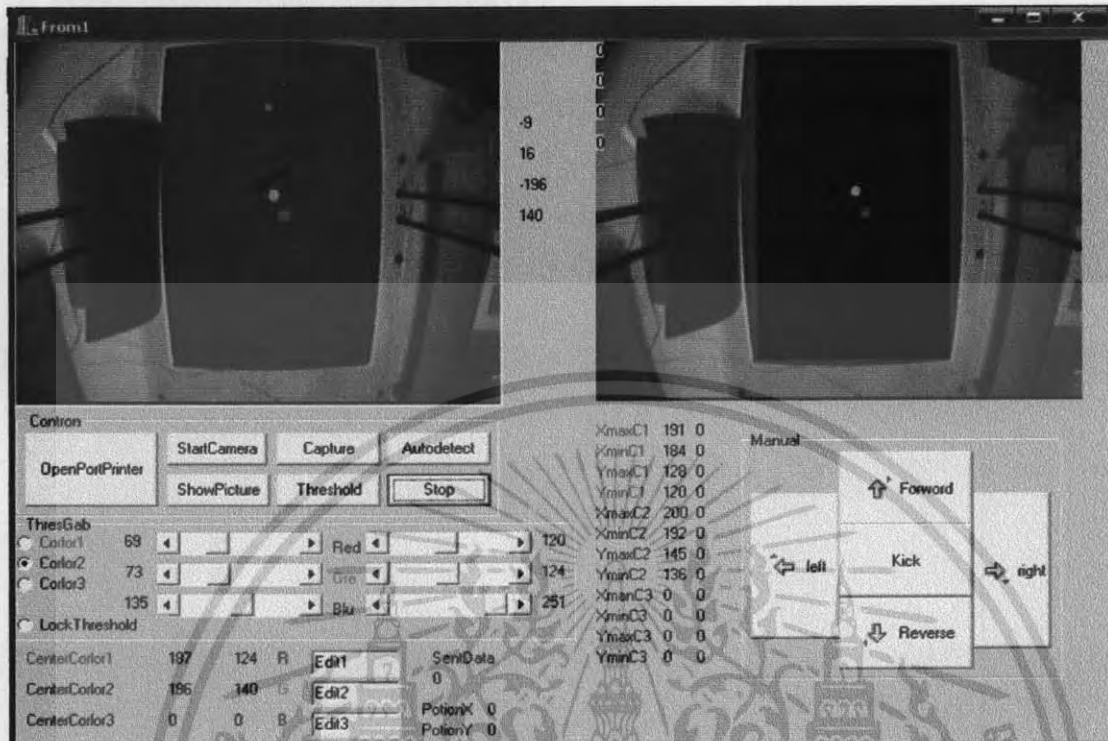
4.4 แสดงการทดลองทำ Threshold สีฟ้า



รูปที่ 4.8 แสดงการทำ Threshold สีฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 แสดงการทดลองทำ Threshold พร้อมกัน 2 สี



รูปที่ 4.9 แสดงการทำ Threshold พร้อมกัน 2 สี

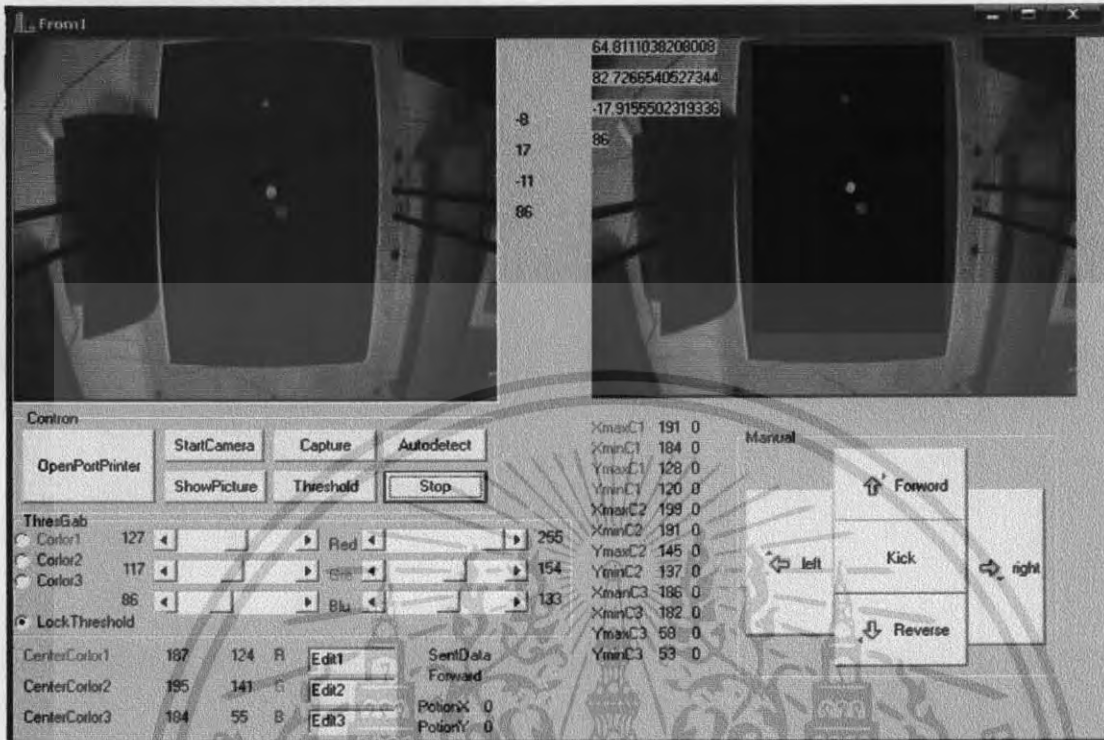
4.6 แสดงการทดลองทำ Threshold พร้อมกัน 3 สี



รูปที่ 4.10 แสดงทำ Threshold พร้อมกัน 3 สี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 แสดงการทดลองทำ Auto พร้อมกัน3สี



รูปที่4.11แสดงการทำ Auto พร้อมกัน3สี

หลังจากทำการจับสีทั้งสามสีแล้ว โดยที่สีเขียวจะเป็นสีแรกที่จะต้องจับก่อน สีเขียวเป็นสีที่บอกถึงว่าเป็นตำแหน่งของหัวหุ่นยนต์ สีแดงจะเป็นสีที่สองในการจับตำแหน่งของสี สีแดงจะบอกถึงว่าเป็นตำแหน่งท้ายของหุ่นยนต์ สีฟ้าเป็นสีที่สามที่จะตรวจจับตำแหน่ง สีที่สามจะบอกให้รู้ว่าเป็นลูกบอกลที่ต้องการ เมื่อทำการจับสีทั้ง สามสี พร้อมกัน โปรแกรมจะคำนวณหาจุดสูงสุด จุดต่ำสุด จุดซ้ายสุด จุดขวาสุด เพื่อมาคำนวณหาจุดกึ่งกลางของสีแต่ละสีที่ต้องการจับตำแหน่ง จุดกึ่งกลางของแต่ละสีจะมีค่าผิดพลาดไม่เกิน ± 1 หลังจากได้ตำแหน่งแล้ว โปรแกรมจะเอาจุดกึ่งกลางไปคำนวณหา เพื่อ สั่งให้หุ่นยนต์เดินไปตามตำแหน่งได้ จากการทดลอง หุ่นยนต์ สามารถวิ่งไปหาลูกบอลได้ เมื่อลูกบอลเปลี่ยนตำแหน่งไป หุ่นยนต์ก็สามารถวิ่ง ไปหาลูกบอลได้ตามที่ตำแหน่งที่ลูกบอลอยู่ เมื่อหุ่นยนต์วิ่งไปถึงลูกบอลได้แล้วมุมระหว่างลูกบอลอยู่ในช่วง ± 5 Pixel ของภาพ และระยะทางน้อยกว่า 35 Pixel ของภาพ หุ่นยนต์จะทำการเตะลูกบอล จากการทดลองมีความผิดพลาดเกิดจากที่ระยะของกล้องในแต่ละมุมไม่เท่ากันอันเนื่องมาจาก ความโค้งงอของเลน ทำให้หุ่นยนต์เข้าใจผิดว่า ได้ระยะแล้วจึงสั่งยิงลูกบอล โดยที่ระยะที่แท้จริงยังไม่ได้ แต่หุ่นยนต์ถูกออกแบบมาให้สามารถแก้ไขการเดินได้ตลอดเวลาเพราะเป็นแบบ Real Time จึงสามารถหาตำแหน่งของลูกบอลได้ตลอด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

ปัญหาและวิธีการแก้ปัญหา

- ปัญหาอันเนื่องมาจากการออกแบบวงจรควบคุมหุ่นยนต์ เนื่องจากต้องทำวงจรใหม่หมด ทำให้เกิดข้อผิดพลาดในการออกแบบอยู่หลายประการ
วิธีการแก้ปัญหาต้องศึกษาอุปกรณ์ที่นำมาใช้งานอย่างละเอียดจึงสามารถแก้ปัญหาได้บางส่วน
- ปัญหาจากการทำ Sensor นับรอบการหมุนของล้อ ที่ไม่ สามารถทำให้นับได้ตามต้องการเกิดจากการเข้าใจโครงสร้างที่ผิด
วิธีการแก้ปัญหา โดยการวัดและทดลองการทำงานอย่างละเอียดอีกครั้งเพื่อความแน่ใจ
- ปัญหาอันเนื่องมาจากการทำงานของ คอลโทรลเลอร์ไม่สามารถจัดการเปลี่ยนแปลงของ Sensor ที่ลို့ได้
วิธีการแก้ปัญหา เนื่องจากตอนแรกมีแสงจากภายนอกรบกวน Sensor ตัวที่ใช้จึงทำการเปลี่ยน Sensor ใหม่ที่ตัวป้องกันแสงเข้าเพื่อลดการรบกวนจากแสงภายนอก
- ปัญหาจากตัวส่งไร้สาย ที่ไม่สามารถส่งข้อมูลได้อย่างแม่นยำ
วิธีการแก้ปัญหา โดยการทดลองหาเวลาที่ดียที่สุดที่ตัวส่งจะส่งข้อมูลแล้วให้ตัวรับถอดรหัสได้ทัน เมื่อได้เวลาที่ดียที่สุดก็จะไปเขียนควบคุมการส่งด้วยคอลโทรลเลอร์
- ปัญหาเกิดจาก Bug ที่เกิดขึ้นจากการเขียนโปรแกรม ทำให้คอลโทรลเลอร์ทำงานผิดพลาด ไม่สามารถควบคุมรถและส่งข้อมูลตามต้องการได้
วิธีการแก้ปัญหา คือการตรวจสอบการเขียน โปรแกรมแล้วทดลองทำการ Simulate เพื่อหาข้อผิดพลาดแล้วทำการแก้ไข
- ปัญหาที่เกิดจาก การใช้งานโปรแกรม C++Builder 5 เนื่องจาก C++Builder 5 ไม่สามารถ ดัด Printer Port ได้ จึงไม่สามารถส่งข้อการควบคุมด้วยคอมพิวเตอร์ได้
วิธีการแก้ปัญหา ได้เปลี่ยนมาใช้ โปรแกรม C++Builder6 แทน ก็สามารถติดต่อ Printer Port ได้
- ปัญหาที่เกิดจาก ไม่สามารถ ติดต่อกลို့ได้ เนื่องจาก Component ที่ใช้ไม่สามารถใช้บน C++Builder6 ได้
วิธีการแก้ปัญหา เนื่องจาก Component ที่ใช้สามารถใช้ได้บน C++Builder 5 เท่านั้น จึงเปลี่ยนมาใช้ C++Builder 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ปัญหาสืบเนื่อง จากการ ที่ C++Builder 5 ไม่สามารถ ติดต่อ Printer Port ได้ทำให้ ติดต่อกลับได้ แต่ไม่สามารถติดต่อ Printer Port ได้

วิธีการแก้ปัญหา เมื่อติดต่อ Printer Port ด้วย C++Builder 5 โปรแกรม C++Builder5 จะเรียกไฟล์ที่ต้องการ คัดลอกไฟล์ที่ต้องการจาก C++Builder 6 เขา ไปลงไว้ที่ C++Builder 5 ก็จะสามารถติดต่อ Printer Port ได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

สรุปและวิจารณ์ผลการทดลอง

ในโครงการนี้เป็นรูปแบบการทำงานของระบบการควบคุมหุ่นยนต์ ด้วยคอมพิวเตอร์ระยะไกลแบบอัตโนมัติผ่านเครื่องรับส่งแบบไร้สาย ซึ่งใช้หลักการของการประมวลผลภาพที่ได้จากกล้องดิจิทัล จะพบว่าจากโครงการนี้จะใช้เทคนิคของอิมเมจ โพรเซสซึ่งมาประมวลผลเพื่อให้รู้ตำแหน่งของหุ่นยนต์และลูกบอลว่าอยู่ในทิศทางเดียวกันหรือไม่ถ้าไม่อยู่ในทิศทางเดียวกัน อิมเมจจะทำหน้าที่ประมวลผลและควบคุมให้หุ่นยนต์เดินไปในตำแหน่งที่ถูกต้อง และสามารถเตะลูกบอลให้เข้าประตูได้แต่เนื่องจากการใช้อิมเมจโพรเซสซึ่งมาประมวลผล ในเทอมนี้สามารถแยกสีต่างๆ ออกจากสิ่งแวดล้อมได้ตามที่ต้องการ โดยการแยกสีสามารถแยกได้พร้อมกันทีเดียว 3 สี และสามารถแยกเพิ่มได้อีก แล้วแต่ว่ามีสีที่ต้องการแยกทั้งหมดกี่สี เพียงเพิ่มส่วนของโปรแกรมเข้าไปเล็กน้อย เพราะ โปรแกรมที่เขียนขึ้นสามารถทำความเข้าใจได้ง่าย

เมื่อสามารถแยกสีออกจากสิ่งแวดล้อมได้แล้วก็สามารถหาพิกัดต่างๆ และ จุดศูนย์กลางของสีได้ แล้ว สามารถนำพิกัดของสีมาคำนวณหาระยะห่างและมุมที่ต้อง การแล้วทำการส่งข้อมูลที่คำนวณได้ส่งออกไปที่ตัวส่งไร้สายเพื่อไปควบคุมหุ่นยนต์ได้ตามที่คำนวณ

หุ่นยนต์สามารถรับคำสั่งได้โดยการส่งจะค่อยๆส่งไปที่ละคำสั่ง 1 คำสั่งจะถูกแยกออกเป็น 4 บิตบน ตัวรถจะนำคำสั่งที่ได้มา แล้วนำไปประมวลผล

สำหรับแนวทางที่จะพัฒนาโครงการนี้ต้องมีต้นแบบหุ่นยนต์ที่ดีและสามารถนำไปใช้งานได้ทันทีหรือเพิ่มเติมอะไรไปเพียงเล็กน้อยเท่านั้นเพื่อไม่ให้เสียเวลาในการพัฒนามาก แล้วหันมาพัฒนาด้านซอฟต์แวร์เพียงอย่างเดียว จะทำให้ได้หุ่นยนต์ที่มีประสิทธิภาพสูงกว่าที่ ต้องมาทำใหม่ทั้งหมดมันเป็นการสิ้นเปลืองเวลาสิ้นเปลือง ทรัพยากรเป็นอย่างมากทำให้ความรู้ทางด้านอิมเมจเป็นไปได้ช้า เพราะต้องมาทำความเข้าใจกับโปรแกรมที่จะนำมาเขียนต้องเรียนรู้ทฤษฎีทางด้านอิมเมจจึงทำให้มองว่าเป็นการยุ่งยากมากที่จะพัฒนาต่อไป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Features

- Operating voltage
 - 2.4V~5V for the HT12A
 - 2.4V~12V for the HT12E
- Low power and high noise immunity CMOS technology
- Low standby current: 0.1μA (typ.) at V_{DD}=5V
- HT12A with a 38kHz carrier for infrared transmission medium
- Minimum transmission word
 - Four words for the HT12E
 - One word for the HT12A
- Built-in oscillator needs only 5% resistor
- Data code has positive polarity
- Minimal external components
- HT12A/E: 18-pin DIP/20-pin SOP package

Applications

- Burglar alarm system
- Smoke and fire alarm system
- Garage door controllers
- Car door controllers
- Car alarm system
- Security system
- Cordless telephones
- Other remote control systems

General Description

The 2¹² encoders are a series of CMOS LSIs for remote control system applications. They are capable of encoding information which consists of N address bits and 12-N data bits. Each address/data input can be set to one of the two logic states. The programmed addresses/data are transmitted together with the header bits

via an RF or an infrared transmission medium upon receipt of a trigger signal. The capability to select a \overline{TE} trigger on the HT12E or a DATA trigger on the HT12A further enhances the application flexibility of the 2¹² series of encoders. The HT12A additionally provides a 38kHz carrier for infrared systems.

Selection Table

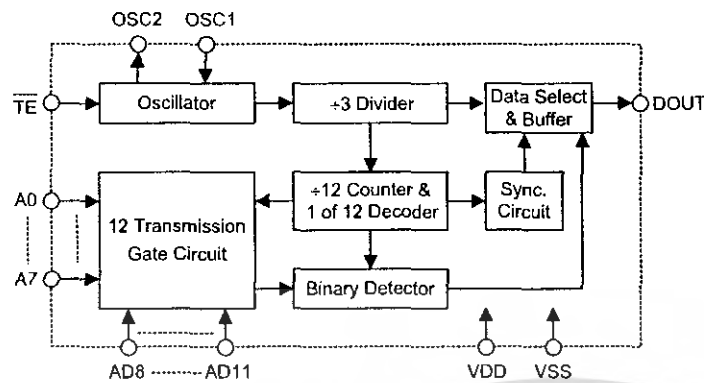
Function Part No.	Address No.	Address/ Data No.	Data No.	Oscillator	Trigger	Package	Carrier Output	Negative Polarity
HT12A	8	0	4	455kHz resonator	D8~D11	18 DIP 20 SOP	38kHz	No
HT12E	8	4	0	RC oscillator	\overline{TE}	18 DIP 20 SOP	No	No

Note: Address/Data represents pins that can be address or data according to the decoder requirement.

Block Diagram

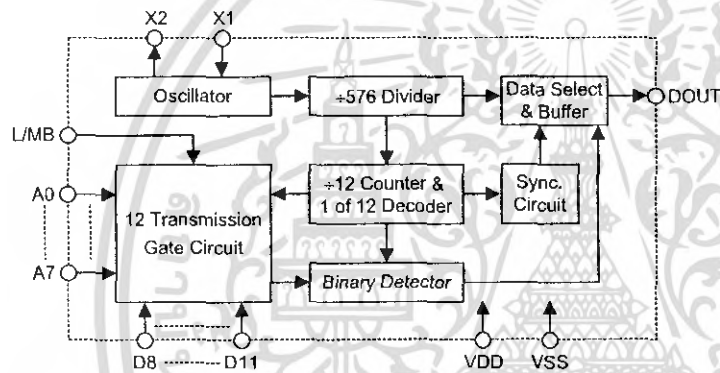
\overline{TE} trigger

HT12E



DATA trigger

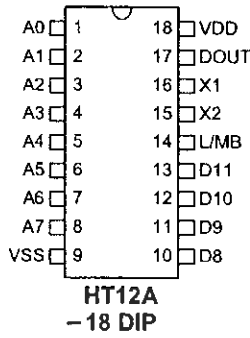
HT12A



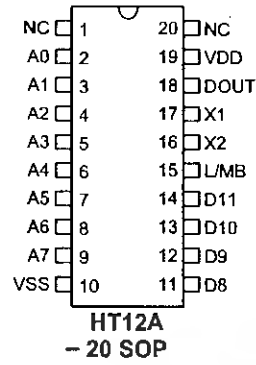
Note: The address data pins are available in various combinations (refer to the address/data table).

Pin Assignment

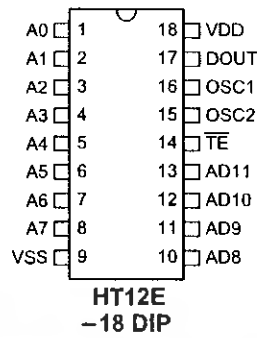
8-Address
4-Data



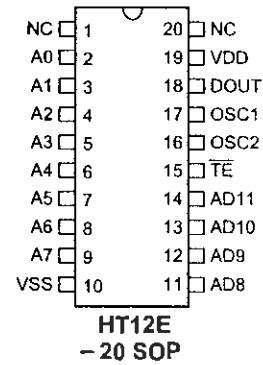
8-Address
4-Data



8-Address
4-Address/Data



8-Address
4-Address/Data



Pin Description

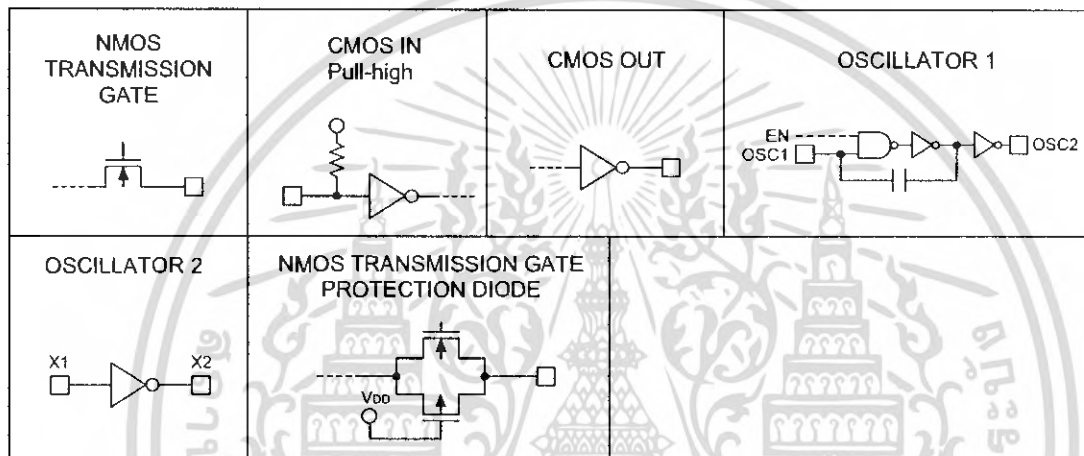
Pin Name	I/O	Internal Connection	Description
A0~A7	I	CMOS IN Pull-high (HT12A) NMOS TRANSMISSION GATE PROTECTION DIODE (HT12E)	Input pins for address A0~A7 setting These pins can be externally set to VSS or left open
AD8~AD11	I	NMOS TRANSMISSION GATE PROTECTION DIODE (HT12E)	Input pins for address/data AD8~AD11 setting These pins can be externally set to VSS or left open
D8~D11	I	CMOS IN Pull-high	Input pins for data D8~D11 setting and transmission enable, active low These pins should be externally set to VSS or left open (see Note)
DOUT	O	CMOS OUT	Encoder data serial transmission output
L/MB	I	CMOS IN Pull-high	Latch/Momentary transmission format selection pin: Latch: Floating or VDD Momentary: VSS

Pin Name	I/O	Internal Connection	Description
\overline{TE}	I	CMOS IN Pull-high	Transmission enable, active low (see Note)
OSC1	I	OSCILLATOR 1	Oscillator input pin
OSC2	O	OSCILLATOR 1	Oscillator output pin
X1	I	OSCILLATOR 2	455kHz resonator oscillator input
X2	O	OSCILLATOR 2	455kHz resonator oscillator output
VSS	I	—	Negative power supply, grounds
VDD	I	—	Positive power supply

Note: D8~D11 are all data input and transmission enable pins of the HT12A.

\overline{TE} is a transmission enable pin of the HT12E.

Approximate internal connections



Absolute Maximum Ratings

Supply Voltage (HT12A)	-0.3V to 5.5V	Supply Voltage (HT12E)	-0.3V to 13V
Input Voltage.....	$V_{SS}-0.3$ to $V_{DD}+0.3V$	Storage Temperature.....	-50°C to 125°C
Operating Temperature.....	-20°C to 75°C		

Note: These are stress ratings only. Stresses exceeding the range specified under "Absolute Maximum Ratings" may cause substantial damage to the device. Functional operation of this device at other conditions beyond those listed in the specification is not implied and prolonged exposure to extreme conditions may affect device reliability.

Electrical Characteristics

HT12A

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V _{DD}	Conditions				
V _{DD}	Operating Voltage	—	—	2.4	3	5	V
I _{STB}	Standby Current	3V	Oscillator stops	—	0.1	1	μA
		5V		—	0.1	1	μA
I _{DD}	Operating Current	3V	No load f _{OSC} =455kHz	—	200	400	μA
		5V		—	400	800	μA
I _{DOUT}	Output Drive Current	5V	V _{OH} =0.9V _{DD} (Source)	-1	-1.6	—	mA
			V _{OL} =0.1V _{DD} (Sink)	2	3.2	—	mA
V _{IH}	"H" Input Voltage	—	—	0.8V _{DD}	—	V _{DD}	V
V _{IL}	"L" Input Voltage	—	—	0	—	0.2V _{DD}	V
R _{DATA}	D8~D11 Pull-high Resistance	5V	V _{DATA} =0V	—	150	300	kΩ

HT12E

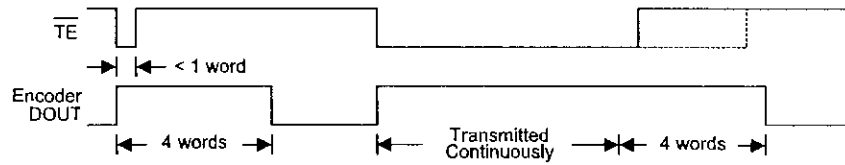
Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V _{DD}	Conditions				
V _{DD}	Operating Voltage	—	—	2.4	5	12	V
I _{STB}	Standby Current	3V	Oscillator stops	—	0.1	1	μA
		12V		—	2	4	μA
I _{DD}	Operating Current	3V	No load f _{OSC} =3kHz	—	40	80	μA
		12V		—	150	300	μA
I _{DOUT}	Output Drive Current	5V	V _{OH} =0.9V _{DD} (Source)	-1	-1.6	—	mA
			V _{OL} =0.1V _{DD} (Sink)	1	1.6	—	mA
V _{IH}	"H" Input Voltage	—	—	0.8V _{DD}	—	V _{DD}	V
V _{IL}	"L" Input Voltage	—	—	0	—	0.2V _{DD}	V
f _{OSC}	Oscillator Frequency	5V	R _{OSC} =1.1MΩ	—	3	—	kHz
R _{TE}	TE Pull-high Resistance	5V	V _{TE} =0V	—	1.5	3	MΩ

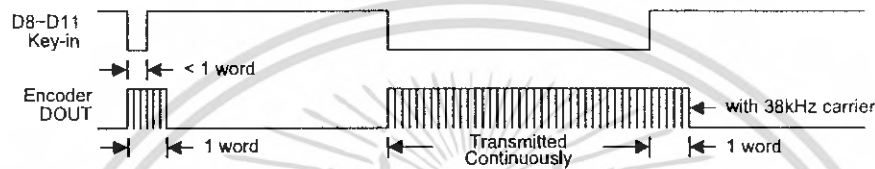
Functional Description

Operation

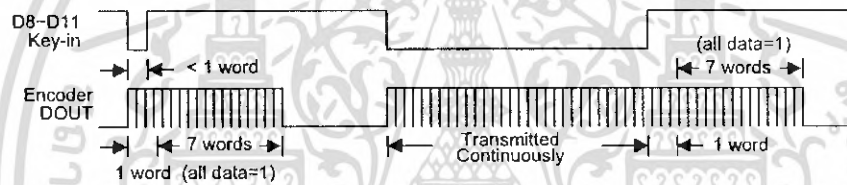
The 2¹² series of encoders begin a 4-word transmission cycle upon receipt of a transmission enable (\overline{TE} for the HT12E or D8~D11 for the HT12A, active low). This cycle will repeat itself as long as the transmission enable (\overline{TE} or D8~D11) is held low. Once the transmission enable returns high the encoder output completes its final cycle and then stops as shown below.



Transmission timing for the HT12E



Transmission timing for the HT12A (L/MB=Floating or VDD)

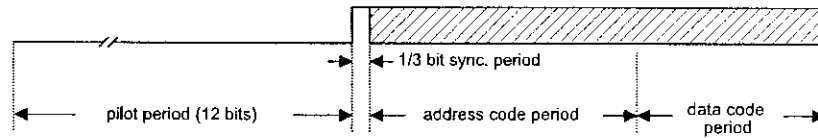


Transmission timing for the HT12A (L/MB=VSS)

Information word

If L/MB=1 the device is in the latch mode (for use with the latch type of data decoders). When the transmission enable is removed during a transmission, the DOUT pin outputs a complete word and then stops. On the other hand, if L/MB=0 the device is in the momentary mode (for use with the momentary type of data decoders). When the transmission enable is removed during a transmission, the DOUT outputs a complete word and then adds 7 words all with the "1" data code.

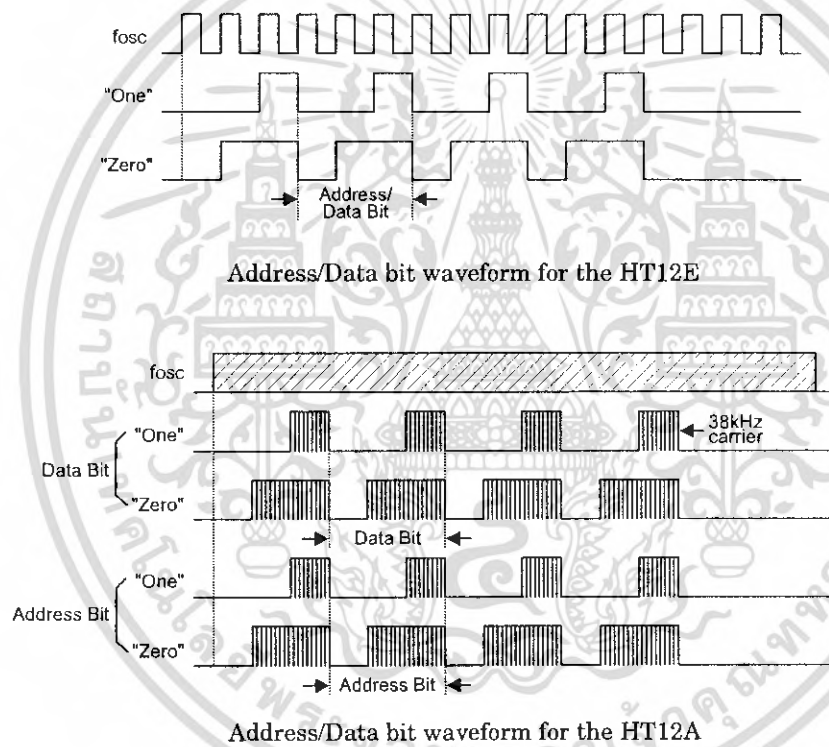
An information word consists of 4 periods as illustrated below.



Composition of information

Address/data waveform

Each programmable address/data pin can be externally set to one of the following two logic states as shown below.



The address/data bits of the HT12A are transmitted with a 38kHz carrier for infrared remote controller flexibility.

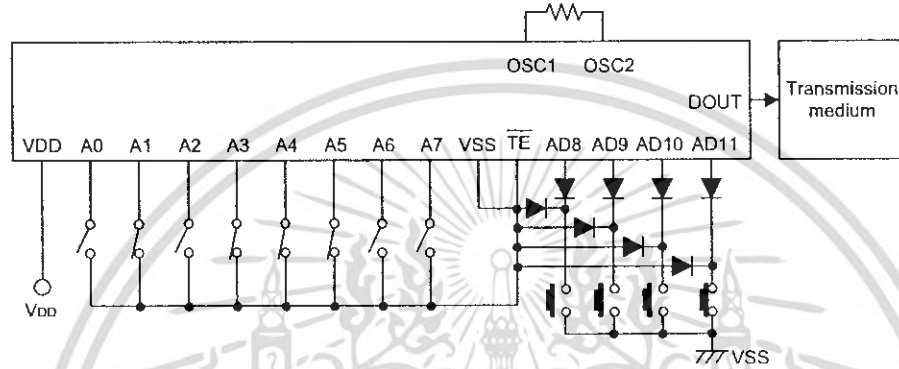
Address/data programming (preset)

The status of each address/data pin can be individually pre-set to logic "high" or "low". If a transmission-enable signal is applied, the encoder scans and transmits the status of the 12 bits of address/data serially in the order A0 to AD11 for the HT12E encoder and A0 to D11 for the HT12A encoder.

During information transmission these bits are transmitted with a preceding synchronization bit. If the trigger signal is not applied, the chip enters the standby mode and consumes a reduced current of less than 1µA for a supply voltage of 5V.

Usual applications preset the address pins with individual security codes using DIP switches or PCB wiring, while the data is selected by push buttons or electronic switches.

The following figure shows an application using the HT12E:



The transmitted information is as shown:

Pilot & Sync.	A0	A1	A2	A3	A4	A5	A6	A7	AD8	AD9	AD10	AD11
1	0	1	0	0	0	1	1	1	1	1	1	0

Address/Data sequence

The following provides the address/data sequence table for various models of the 2¹² series of encoders. The correct device should be selected according to the individual address and data requirements.

Part No.	Address/Data Bits											
	0	1	2	3	4	5	6	7	8	9	10	11
HT12A	A0	A1	A2	A3	A4	A5	A6	A7	D8	D9	D10	D11
HT12E	A0	A1	A2	A3	A4	A5	A6	A7	AD8	AD9	AD10	AD11

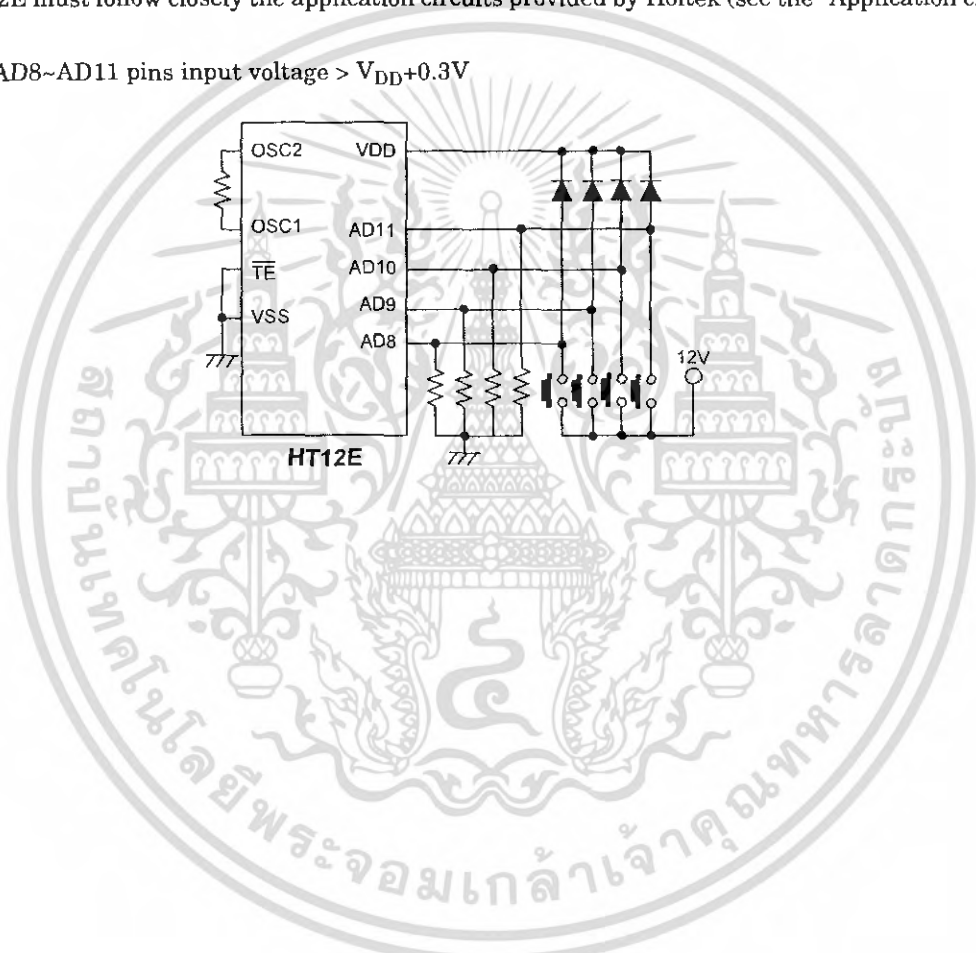
Transmission enable

For the HT12E encoders, transmission is enabled by applying a low signal to the \overline{TE} pin. For the HT12A encoders, transmission is enabled by applying a low signal to one of the data pins D8~D11.

Two erroneous HT12E application circuits

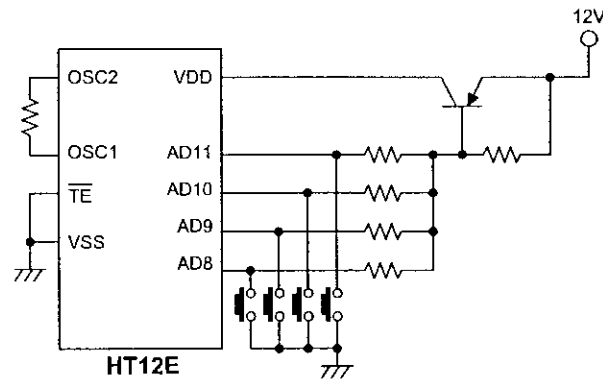
The HT12E must follow closely the application circuits provided by Holtek (see the "Application circuits").

- Error: AD8~AD11 pins input voltage > V_{DD}+0.3V



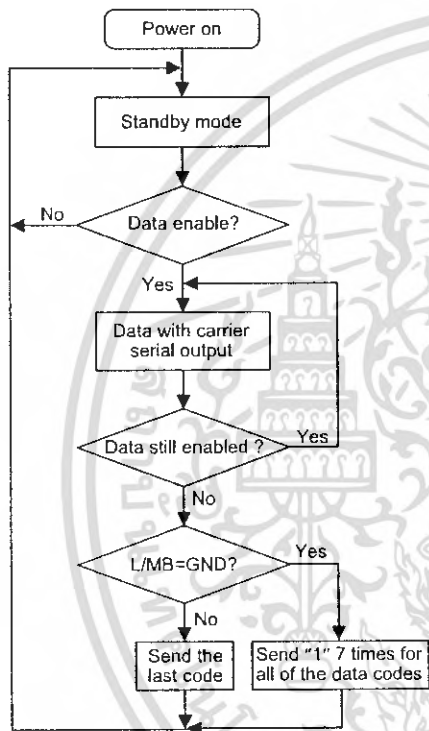
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Error: The IC's power source is activated by pins AD8~AD11

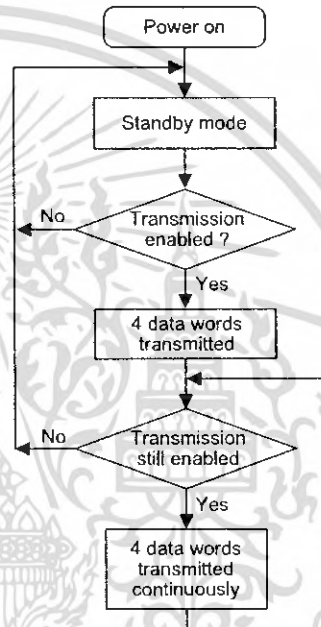


Flowchart

- HT12A



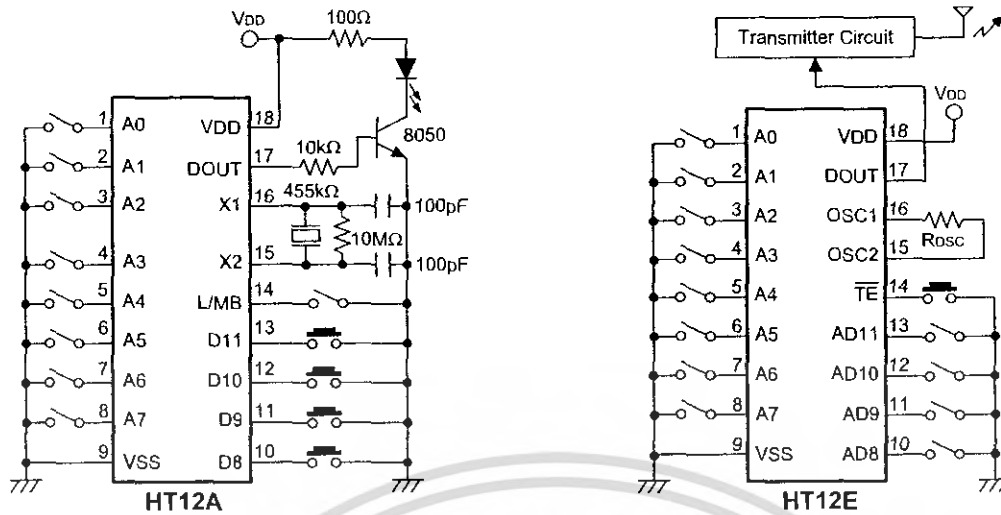
- HT12E



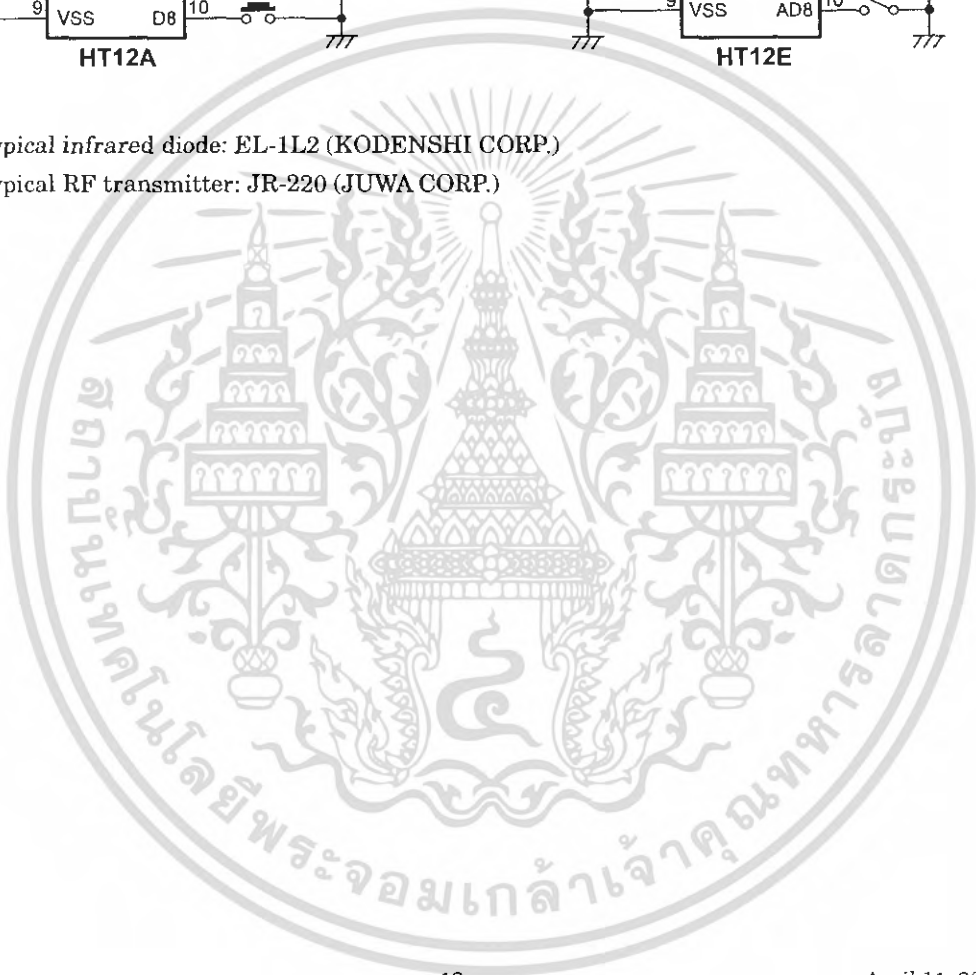
Note: D8~D11 are transmission enables of the HT12A.

\overline{TE} is the transmission enable of the HT12E.

Application Circuits



Note: Typical infrared diode: EL-1L2 (KODENSHI CORP.)
 Typical RF transmitter: JR-220 (JUWA CORP.)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Holtek Semiconductor Inc. (Headquarters)

No.3, Creation Rd. II, Science-based Industrial Park, Hsinchu, Taiwan, R.O.C.
Tel: 886-3-563-1999
Fax: 886-3-563-1189

Holtek Semiconductor Inc. (Taipei Office)

11F, No.576, Sec.7 Chung Hsiao E. Rd., Taipei, Taiwan, R.O.C.
Tel: 886-2-2782-9635
Fax: 886-2-2782-9636
Fax: 886-2-2782-7128 (International sales hotline)

Holtek Semiconductor (Hong Kong) Ltd.

RM.711, Tower 2, Cheung Sha Wan Plaza, 833 Cheung Sha Wan Rd., Kowloon, Hong Kong
Tel: 852-2-745-8288
Fax: 852-2-742-8657

Holtek Semiconductor (Shanghai) Ltd.

7th Floor, Building 2, No.889, Yi Shan Rd., Shanghai, China
Tel: 021-6485-5560
Fax: 021-6485-0313

Holmate Technology Corp.

48531 Warm Springs Boulevard, Suite 413, Fremont, CA 94539
Tel: 510-252-9880
Fax: 510-252-9885

Copyright © 2000 by HOLTEK SEMICONDUCTOR INC.

The information appearing in this Data Sheet is believed to be accurate at the time of publication. However, Holtek assumes no responsibility arising from the use of the specifications described. The applications mentioned herein are used solely for the purpose of illustration and Holtek makes no warranty or representation that such applications will be suitable without further modification, nor recommends the use of its products for application that may present a risk to human life due to malfunction or otherwise. Holtek reserves the right to alter its products without prior notification. For the most up-to-date information, please visit our web site at <http://www.holtek.com.tw>.

Features

- Operating voltage: 2.4V~12V
- Low power and high noise immunity CMOS technology
- Low standby current
- Capable of decoding 12 bits of information
- Pair with Holtek's 2¹² series of encoders
- Binary address setting
- Received codes are checked 3 times
- Address/Data number combination
 - HT12D: 8 address bits and 4 data bits
 - HT12F: 12 address bits only
- Built-in oscillator needs only 5% resistor
- Valid transmission indicator
- Easy interface with an RF or an infrared transmission medium
- Minimal external components

Applications

- Burglar alarm system
- Smoke and fire alarm system
- Garage door controllers
- Car door controllers
- Car alarm system
- Security system
- Cordless telephones
- Other remote control systems

General Description

The 2¹² decoders are a series of CMOS LSIs for remote control system applications. They are paired with Holtek's 2¹² series of encoders (refer to the encoder/decoder cross reference table). For proper operation, a pair of encoder/decoder with the same number of addresses and data format should be chosen.

The decoders receive serial addresses and data from a programmed 2¹² series of encoders that are transmitted by a carrier using an RF or an IR transmission medium. They compare the serial input data three times continuously with

their local addresses. If no error or unmatched codes are found, the input data codes are decoded and then transferred to the output pins. The VT pin also goes high to indicate a valid transmission.

The 2¹² series of decoders are capable of decoding informations that consist of N bits of address and 12-N bits of data. Of this series, the HT12D is arranged to provide 8 address bits and 4 data bits, and HT12F is used to decode 12 bits of address information.

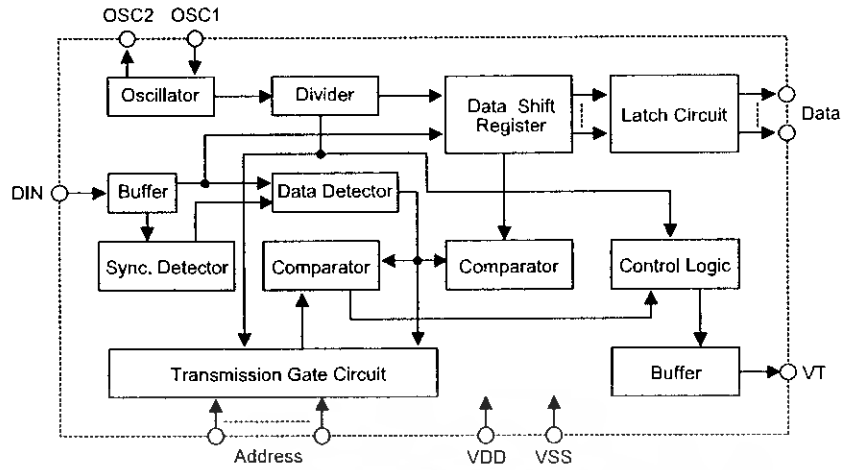
Selection Table

Function Part No.	Address No.	Data		VT	Oscillator	Trigger	Package
		No.	Type				
HT12D	8	4	L	√	RC oscillator	DIN active "Hi"	18 DIP/20 SOP
HT12F	12	0	—	√	RC oscillator	DIN active "Hi"	18 DIP/20 SOP

Notes: Data type: L stands for latch type data output.

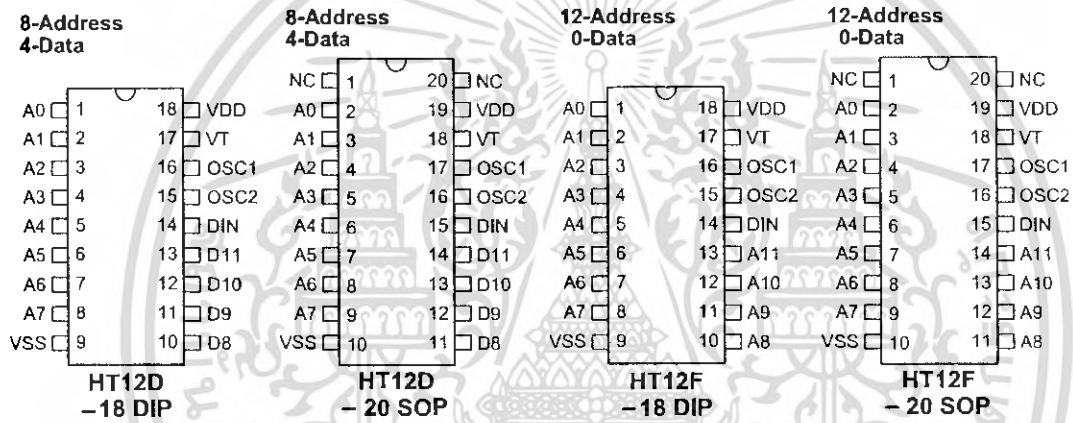
VT can be used as a momentary data output.

Block Diagram



Note: The address/data pins are available in various combinations (see the address/data table).

Pin Assignment

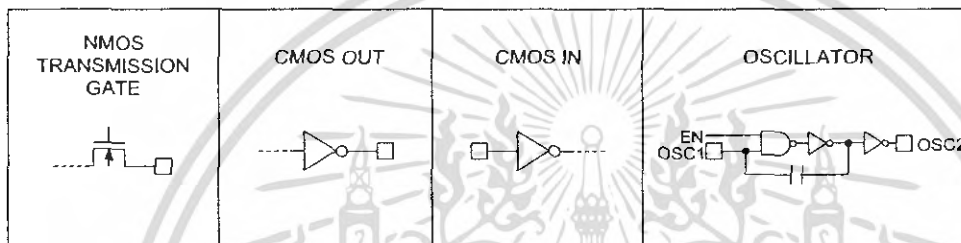


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin Description

Pin Name	I/O	Internal Connection	Description
A0-A11	I	NMOS TRANSMISSION GATE	Input pins for address A0~A11 setting They can be externally set to VDD or VSS.
D8-D11	O	CMOS OUT	Output data pins
DIN	I	CMOS IN	Serial data input pin
VT	O	CMOS OUT	Valid transmission, active high
OSC1	I	OSCILLATOR	Oscillator input pin
OSC2	O	OSCILLATOR	Oscillator output pin
VSS	I	—	Negative power supply (GND)
VDD	I	—	Positive power supply

Approximate internal connection circuits



Absolute Maximum Ratings

Supply Voltage.....-0.3V to 13V Storage Temperature.....-50°C to 125°C
 Input Voltage.....V_{SS}-0.3 to V_{DD}+0.3V Operating Temperature-20°C to 75°C

Note: These are stress ratings only. Stresses exceeding the range specified under "Absolute Maximum Ratings" may cause substantial damage to the device. Functional operation of this device at other conditions beyond those listed in the specification is not implied and prolonged exposure to extreme conditions may affect device reliability.

Electrical Characteristics

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V _{DD}	Conditions				
V _{DD}	Operating Voltage	—	—	2.4	5	12	V
I _{STB}	Standby Current	5V	Oscillator stops	—	0.1	1	μA
		12V		—	2	4	μA
I _{DD}	Operating Current	5V	No load f _{OSC} =150kHz	—	200	400	μA
I _O	Data Output Source Current (D8~D11)	5V	V _{OH} =4.5V	-1	-1.6	—	mA
	Data Output Sink Current (D8~D11)	5V	V _{OL} =0.5V	1	1.6	—	mA
I _{VT}	VT Output Source Current	5V	V _{OH} =4.5V	-1	-1.6	—	mA
	VT Output Sink Current		V _{OL} =0.5V	1	1.6	—	mA
V _{IH}	"H" Input Voltage	5V	—	3.5	—	5	V
V _{IL}	"L" Input Voltage	5V	—	0	—	1	V
f _{OSC}	Oscillator Frequency	5V	R _{OSC} =51kΩ	—	150	—	kHz



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Functional Description

Operation

The 2¹² series of decoders provides various combinations of addresses and data pins in different packages so as to pair with the 2¹² series of encoders.

The decoders receive data that are transmitted by an encoder and interpret the first N bits of code period as addresses and the last 12-N bits as data, where N is the address code number. A signal on the DIN pin activates the oscillator which in turn decodes the incoming address and data. The decoders will then check the received address three times continuously. If the received address codes all match the contents of the decoder's local address, the 12-N bits of data are decoded to activate the output pins and the VT pin is set high to indicate a valid transmission. This will last unless the address code is incorrect or no signal is received.

The output of the VT pin is high only when the transmission is valid. Otherwise it is always low.

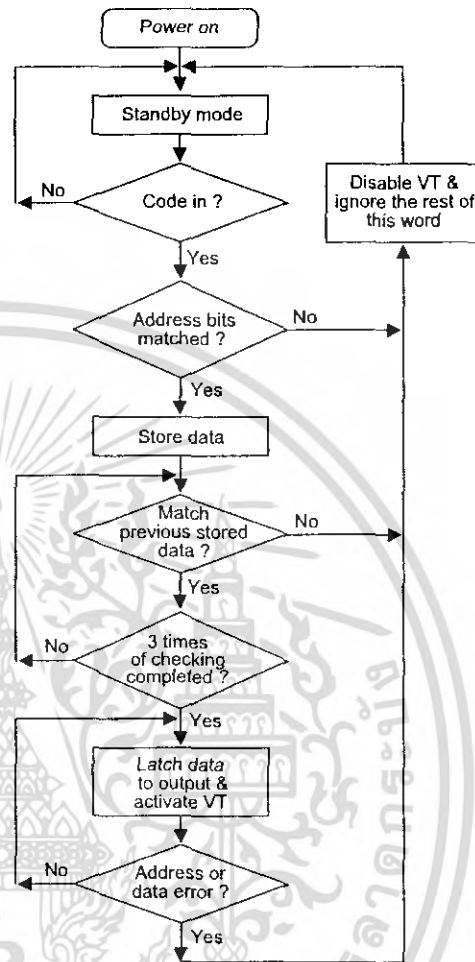
Output type

Of the 2¹² series of decoders, the HT12F has no data output pin but its VT pin can be used as a momentary data output. The HT12D, on the other hand, provides 4 latch type data pins whose data remain unchanged until new data are received.

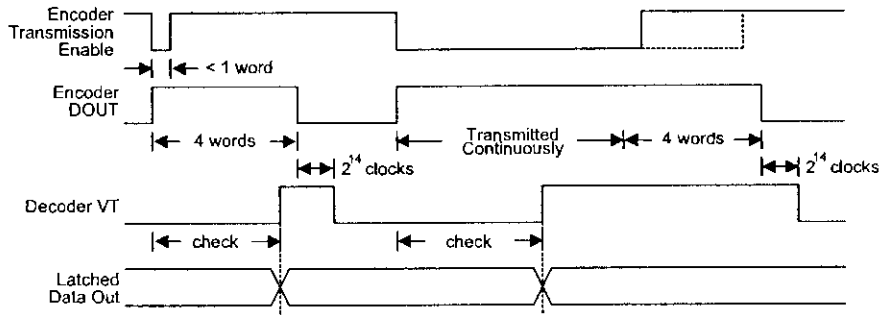
Part No.	Data Pins	Address Pins	Output Type	Operating Voltage
HT12D	4	8	Latch	2.4V~12V
HT12F	0	12	—	2.4V~12V

Flowchart

The oscillator is disabled in the standby state and activated when a logic "high" signal applies to the DIN pin. That is to say, the DIN should be kept low if there is no signal input.



Decoder timing



Encoder/Decoder cross reference table

Decoders Part No.	Data Pins	Address Pins	VT	Pair Encoder	Package			
					Encoder		Decoder	
					DIP	SOP	DIP	SOP
HT12D	4	8	√	HT12A	18	20	18	20
				HT12E	18	20		
HT12F	0	12	√	HT12A	18	20	18	20
				HT12E	18	20		

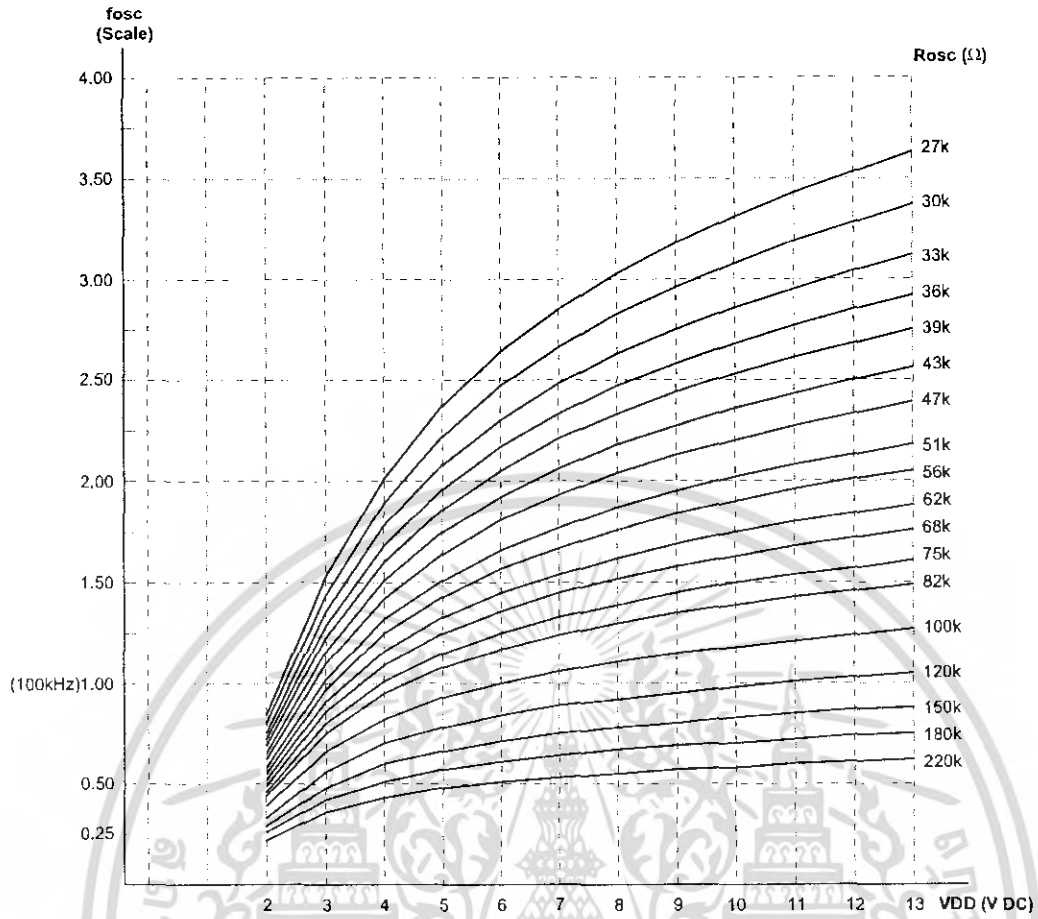
Address/Data sequence

The following table provides address/data sequence for various models of the 2¹² series of decoders. A correct device should be chosen according to the requirements of the individual addresses and data.

Part No.	Address/Data Bits											
	0	1	2	3	4	5	6	7	8	9	10	11
HT12D	A0	A1	A2	A3	A4	A5	A6	A7	D8	D9	D10	D11
HT12F	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

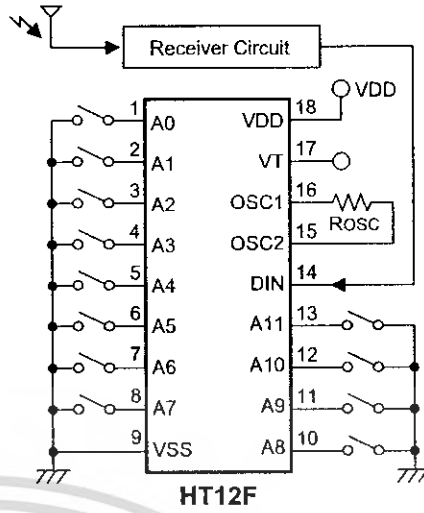
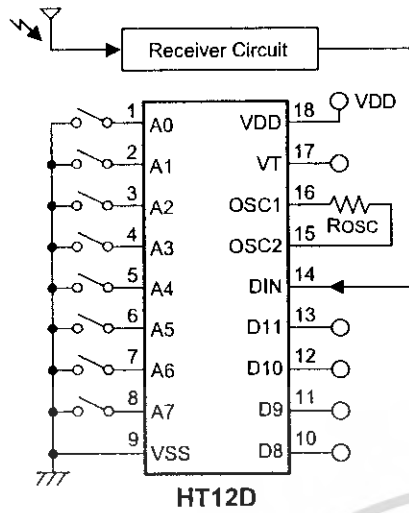
Oscillator frequency vs supply voltage



The recommended oscillator frequency is $f_{OSCD}(\text{decoder}) \cong 50 f_{OSCE}(\text{HT12E encoder})$
 $\cong \frac{1}{3} f_{OSCE}(\text{HT12A encoder})$.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Application Circuits



Notes: Typical infrared receiver: PIC-12043T/PIC-12043S (KODESHI CORP.)
or LTM9052 (LITEON CORP.)

Typical RF receiver: JR-200 (JUWA CORP.)
RE-99 (MING MICROSYSTEM, U.S.A.)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Holtek Semiconductor Inc. (Headquarters)

No.3 Creation Rd. II, Science-based Industrial Park, Hsinchu, Taiwan, R.O.C.
 Tel: 886-3-563-1999
 Fax: 886-3-563-1189

Holtek Semiconductor Inc. (Taipei Office)

11F, No.576, Sec.7 Chung Hsiao E. Rd., Taipei, Taiwan, R.O.C.
 Tel: 886-2-2782-9635
 Fax: 886-2-2782-9636
 Fax: 886-2-2782-7128 (International sales hotline)

Holtek Semiconductor (Hong Kong) Ltd.

RM.711, Tower 2, Cheung Sha Wan Plaza, 333 Cheung Sha Wan Rd., Kowloon, Hong Kong
 Tel: 852-2-745-8288
 Fax: 852-2-742-8657

Holtek Semiconductor (Shanghai) Ltd.

7th Floor, Building 2, No.889, Yi Shan Road, Shanghai, China
 Tel:021-6485-5560
 Fax:021-6485-0313

Holmate Technology Corp.

48531 Warm Springs Boulevard, Suite 413, Fremont, CA 94539
 Tel: 510-252-9880
 Fax: 510-252-9885

Copyright © 1999 by HOLTEK SEMICONDUCTOR INC.

The information appearing in this Data Sheet is believed to be accurate at the time of publication. However, Holtek assumes no responsibility arising from the use of the specifications described. The applications mentioned herein are used solely for the purpose of illustration and Holtek makes no warranty or representation that such applications will be suitable without further modification, nor recommends the use of its products for application that may present a risk to human life due to malfunction or otherwise. Holtek reserves the right to alter its products without prior notification. For the most up-to-date information, please visit our web site at <http://www.holtek.com.tw>.

TA7279P, TA7279AP

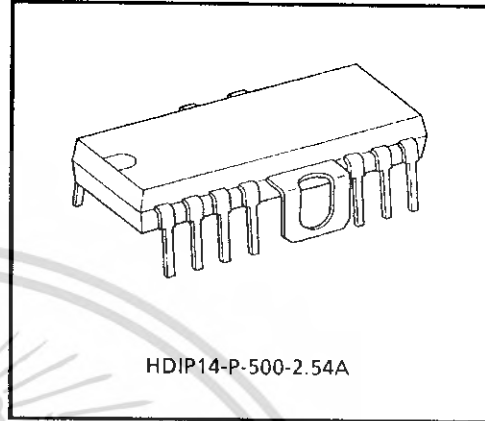
DUAL BRIDGE DRIVER

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

FEATURES

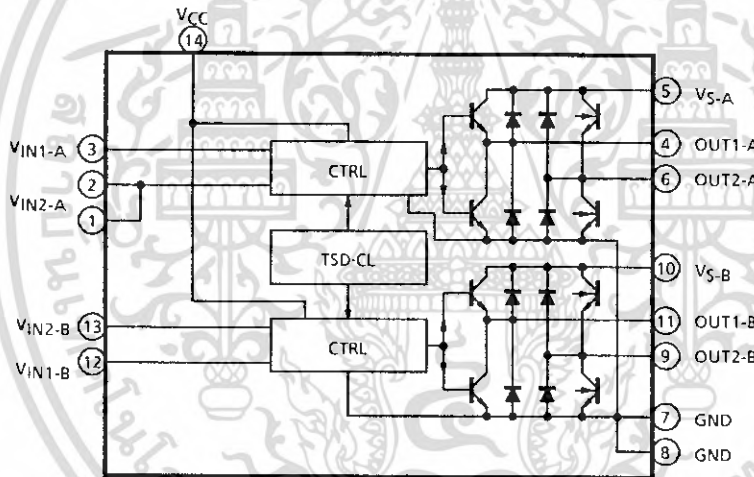
- Wide Range of Operating Voltage
: $V_{CC (opr.)} = 6\sim 18\text{ V (P, AP)}$,
 $V_S (opr.) = 0\sim 16\text{ V (P)} / = 0\sim 18\text{ V (AP)}$
- Output Current Up to 1.0 A (AVE.), 3.0 A (PEAK)
- Built-in Thermal Shut Down and Current Limiter
- Input Hysteresis for Stable Operation

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.

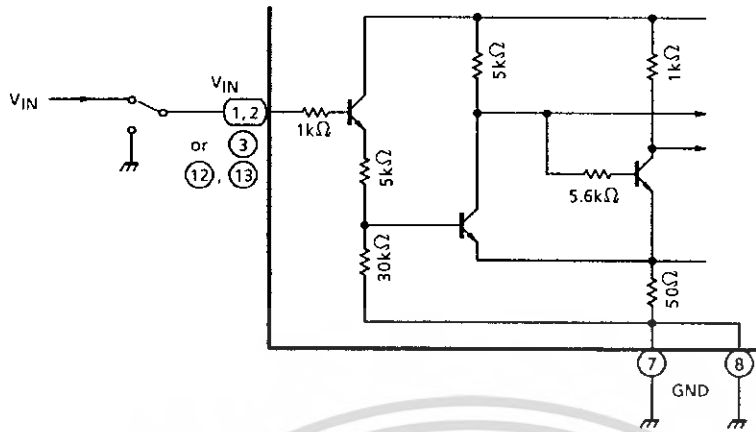
• The information contained herein is subject to change without notice.

PIN FUNCTION

PIN No.	SYMBOL	FUNCTIONAL DESCRIPTION
1	V _{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 Motor 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

APPLICATION NOTE

(1) Input circuit

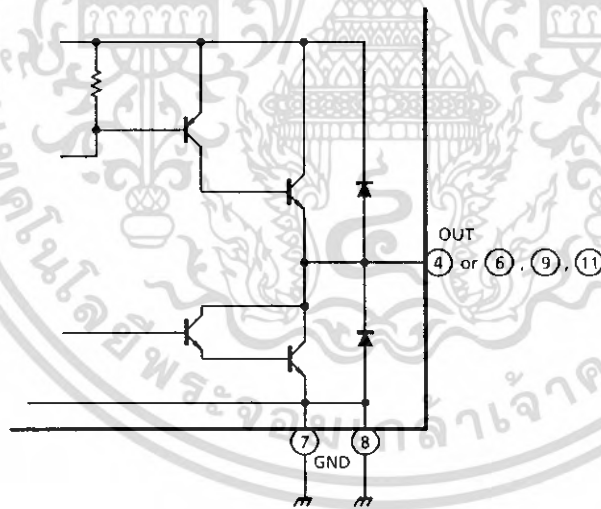


Input terminals of (2), (3), (12) and (13) Pin are all high active type and have a hysteresis.

3 μ 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 _{CC} (MAX.)	25
	P		20
Motor Drive Voltage	AP	V _S (MAX.)	25
	P		18
Output Current	PEAK	I _O (PEAK)	3.0
	AVE.	I _O (AVE.)	1.0
Power Dissipation	P _D (Note)	2.3	W
Operating Temperature	T _{opr}	-30~75	°C
Storage Temperature	T _{stg}	-55~150	°C

Note: No heat sink.

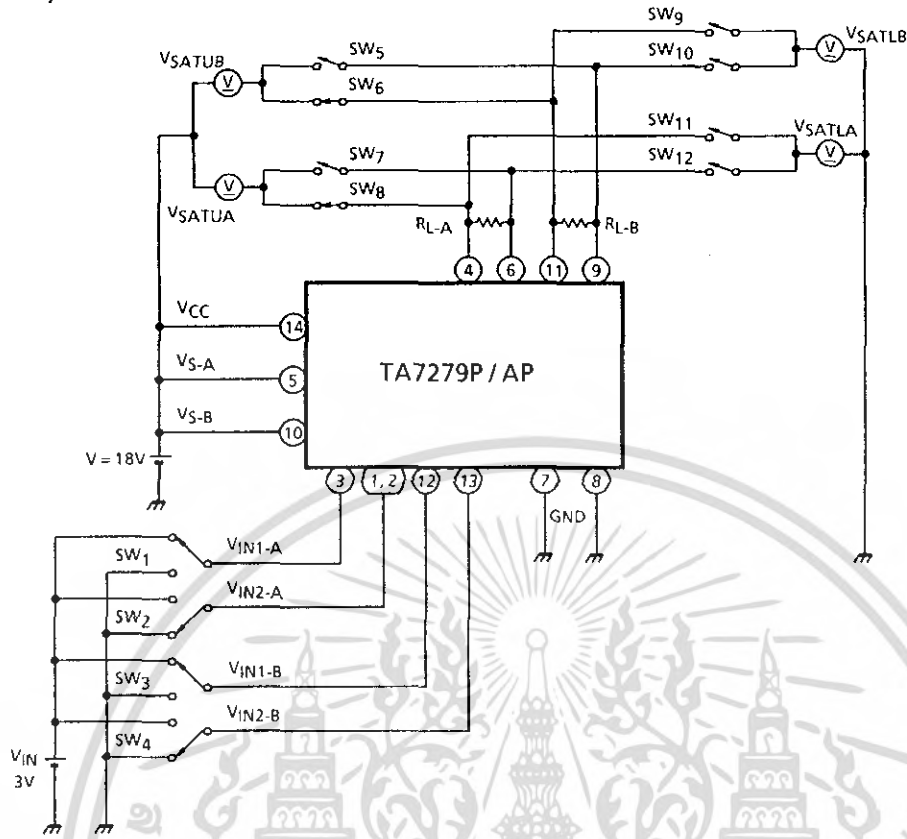
ELECTRICAL CHARACTERISTICS (Ta = 25°C)

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

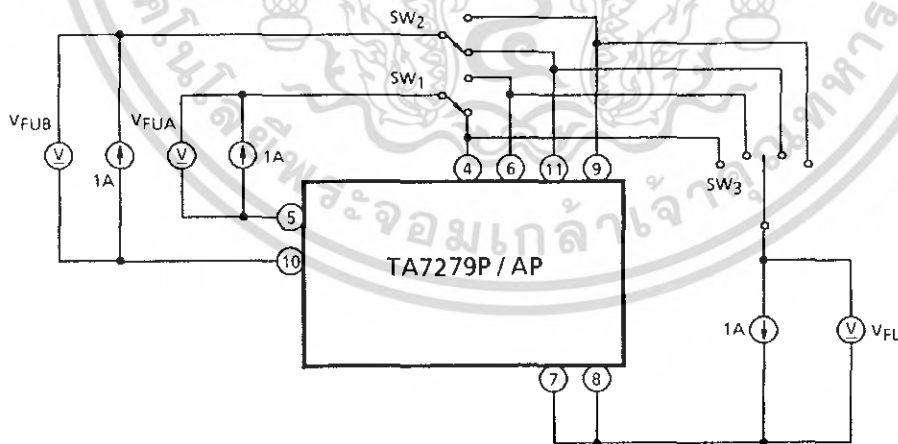
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า

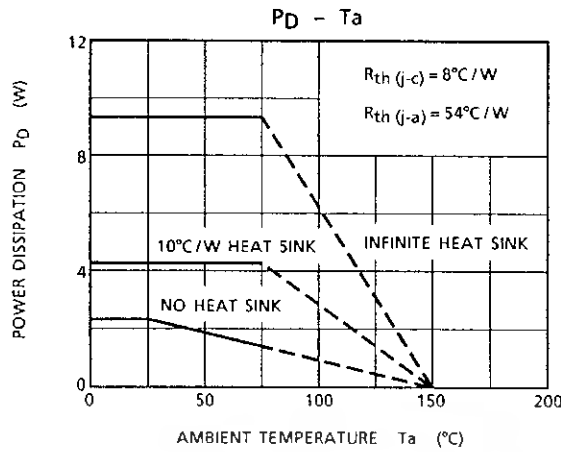
ไม่มี การแก้ไข ที่ดิน อีกที่นี้ ไม่มีให้ที่แสดงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้

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

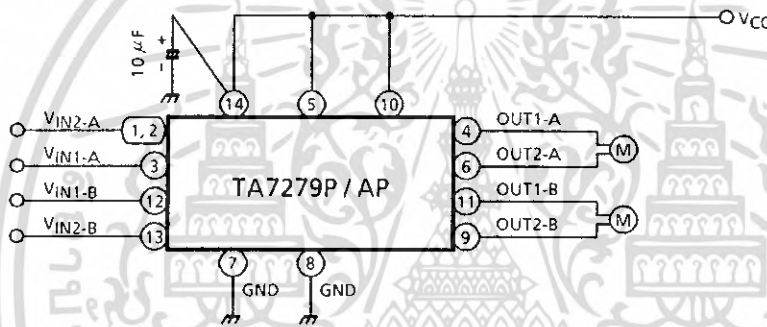


TEST CIRCUIT 4.
 $V_{FU, L}$

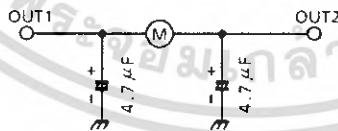




APPLICATION CIRCUIT



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.)

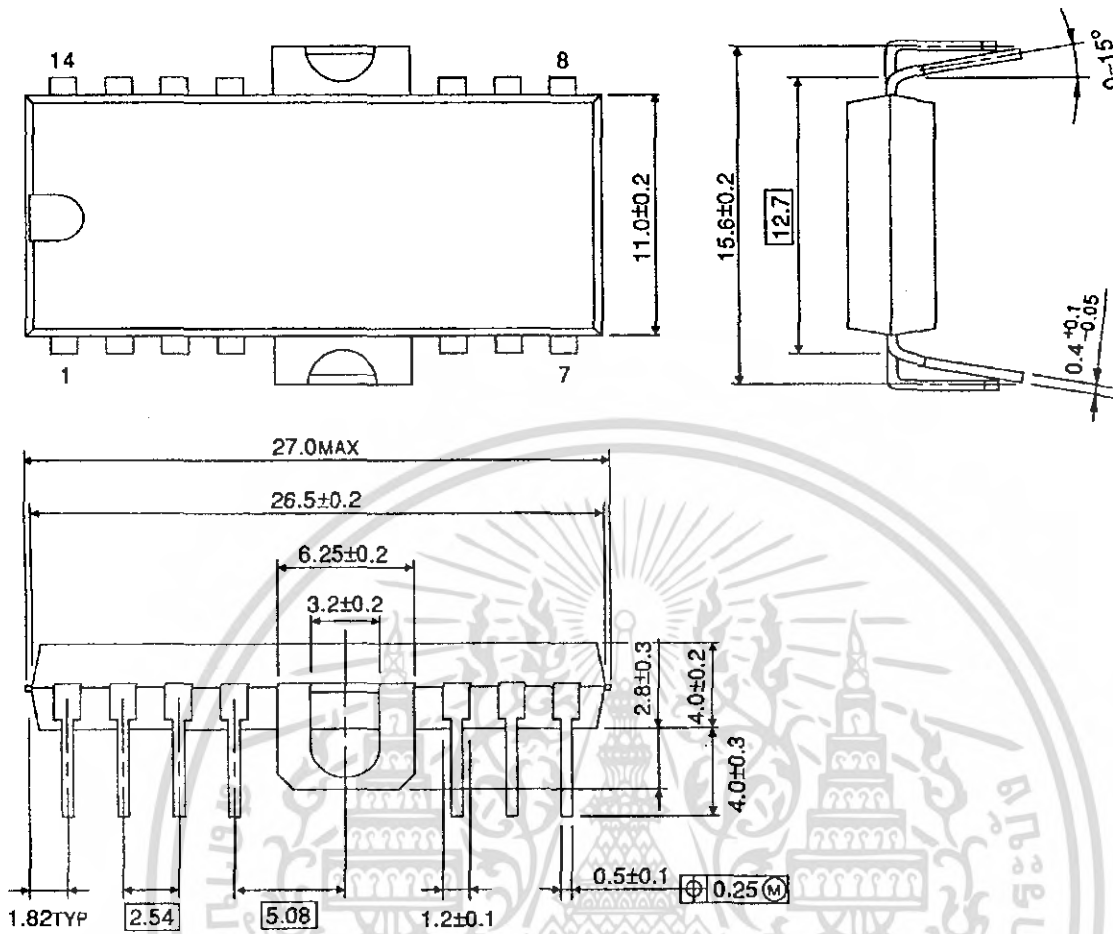


Note: Utmost care is necessary in the design of the output line, V_S and GND line since IC may be destroyed due to short-circuit between outputs, air contamination fault, or fault by improper grounding.

OUTLINE DRAWING

HDIP14-P-500-2.54A

Unit: mm



Weight: 3.00 g (Typ.)

NTPort Library

Version 2.7

User's manual

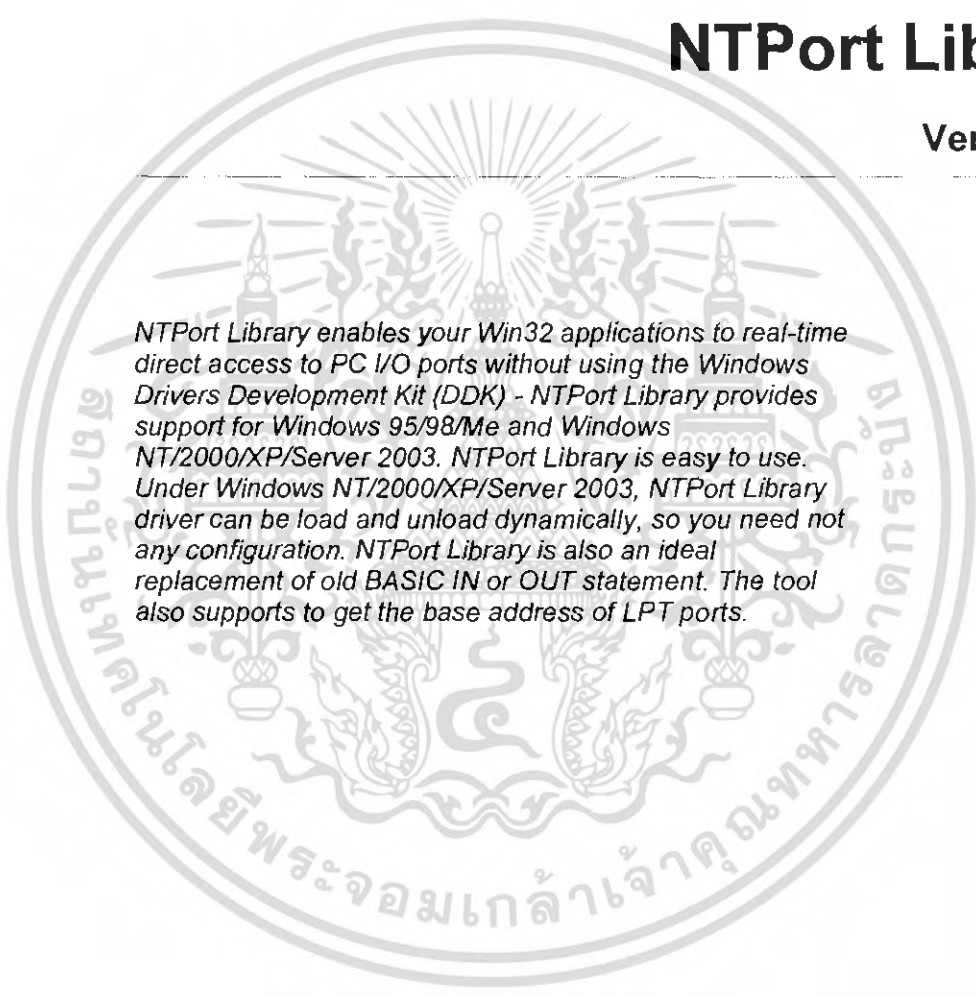
Copyright (c) 1997-2006 Hai Li, Zeal SoftStudio

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NTPort Library

Version 2.7

by LI, Hai



NTPort Library enables your Win32 applications to real-time direct access to PC I/O ports without using the Windows Drivers Development Kit (DDK) - NTPort Library provides support for Windows 95/98/Me and Windows NT/2000/XP/Server 2003. NTPort Library is easy to use. Under Windows NT/2000/XP/Server 2003, NTPort Library driver can be load and unload dynamically, so you need not any configuration. NTPort Library is also an ideal replacement of old BASIC IN or OUT statement. The tool also supports to get the base address of LPT ports.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NTPort Library Reference

Copyright (c) 1997-2006 Hai Li, Zeal SoftStudio

All rights reserved. No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Printed: April 2006 in Beijing, China

Publisher

Zeal SoftStudio

Managing Editor

Li, Hai

Technical Editors

NTPort Library Team

Special thanks to:

All the people who contributed to this document, to all of our valued users.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table of Contents

Foreword	0
Part I Getting Started	4
1 Overview	4
2 Use with Visual Basic	4
3 Use with Visual Basic.NET	4
4 Use with Visual C++	5
5 Use with Visual C#	5
6 Use with Delphi	6
7 Use with Delphi for .NET	6
8 Use with Borland C++ Builder	7
9 Use with Borland C#Builder	7
10 Use with Borland JBuilder	8
11 Contact Information	8
Part II Shipping Your Application	10
1 Distribution Notes	10
2 Using Stand-alone Driver Setup Program	11
3 Using Merge Module for Windows Installer 2.0	11
4 How to Remotely Install Driver	12
5 How to Make Installation Disk	13
Dacris ActiSetup 2005	13
Installer2Go	13
Visual Basic 6.0 Package and Deployment Wizard	13
Visual Studio .NET	14
Visual Studio Installer 1.1	14
Wise for Windows Installer	14
Wise InstallBuilder, InstallMaster and InstallManager 8.x	15
Part III Registration	17
1 Registration Information	17
2 What If I Register	17
Part IV Programming Reference	20
1 Procedures and Functions	20
2 EnablePorts, DisablePorts procedure	20
3 GetFastMode function	21
4 GetLastState function	21
5 GetLPTPortAddress function	22
6 GetNTPortVersion function	23

7	IsWin64 function	23
8	IsWinNT function	23
9	Inp, Inpw, Inpd function	24
10	Inport, InportD, InportW function	25
11	LicenseInfo procedure	25
12	Outp, outpw, outpd procedure	26
13	Outport, OutportW, OutportD procedure	27
14	SetFastMode procedure	27
Part V Frequently Asked Questions		30
1	Frequently Asked Questions	30
Part VI History		33
1	Version History	33
2	Upgrade Issue	33
Part VII Other Products		36
1	CheckListBox ActiveX Control	36
2	Color ComboBox ActiveX Control	36
3	Font List & Combo ActiveX Control	37
4	MemAccess Library	38
Index		39

Getting Started



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1 Getting Started

1.1 Overview

Description

NTPort Library enables your Win32 applications to real-time direct access to PC I/O ports without using the Windows Drivers Development Kit (DDK) - NTPort Library provides support for Windows 95/98/Me and Windows NT/2000/XP/Server 2003 (x86 and x64). NTPort Library is easy to use. Under Windows NT/2000/XP/Server 2003, NTPort Library driver can be load and unload dynamically, so you need not any configuration. NTPort Library is also an ideal replacement of old BASIC IN or OUT statement. The tool also supports to get the base address of LPT ports.

Home Page

Please visit the product homepage at <http://www.zealsoft.com/ntport/>.

We also welcome you to our homepage at <http://www.zealsoft.com> to download other software products.

Registration Information

Order Form

Contact Information

Procedures and Functions

Distribution Notes

Frequently Asked Questions

Upgrade Issue

History

1.2 Use with Visual Basic

NTPort Library can be used with all versions of 32-bit Visual Basic. Please refer to Using with Visual Basic.NET if you are programming for Microsoft .NET platform.

Visual Basic 5.0/6.0 samples are located in NTPort\VB directory

Samples

Purpose

FastMode Illustrates how to use SetFastMode, EnablePorts and DisablePorts procedures to increase the performance under Windows NT/2000/XP/Server 2003.

LPTPort Illustrates how to get the base address of LPT ports.

PortTest Illustrates how to read or write PC I/O ports.

Speaker Illustrates how to control PC speaker.

If you want to use NTPort Library in your Visual Basic project, you should add NTPort.bas in NTPort directory to your Visual Basic project by selecting the menu "Project | Add File".

Please read Distribution Notes if you want to distribute NTPort Library with your Visual Basic project.

1.3 Use with Visual Basic.NET

NTPort Library can be used with Visual Basic.NET 2002/2003/2005. Please refer to Using with

Visual Basic if you are using Visual Basic 4.0 to 6.0.

Visual Basic.NET samples are located in NTPort\VB.NET directory.

Samples	Purpose
FastMode	Illustrates how to use SetFastMode, EnablePorts and DisablePorts procedures to increase the performance under Windows NT/2000/XP/Server 2003.
LPTPort	Illustrates how to get the base address of LPT ports.
PortTest	Illustrates how to read or write PC I/O ports.
Speaker	Illustrates how to control PC speaker.

If you want to use NTPort Library in a new Visual Basic.NET project, you should add NTPort.vb into your projects by selecting the menu "Project | Add Exist Items".

Please read Distribution Notes if you want to distribute NTPort Library with your Visual Basic.NET project.

1.4 Use with Visual C++

NTPort Library can be used with all versions of 32-bit Visual C++.

Visual C++ 5.0/6.0/2002/2003/2005 samples are located in NTPort\VC directory

Samples	Purpose
Console	Illustrates how to use NTPort Library in a Win32 console application.
FastMode	Illustrates how to use SetFastMode, EnablePorts and DisablePorts procedures to increase the performance under Windows NT/2000/XP/Server 2003.
LPTPort	Illustrates how to get the base address of LPT ports.
PortTest	Illustrates how to read or write PC I/O ports.
Speaker	Illustrates how to control PC speaker.

To use NTPort Library in your Visual C++ project, you should do as following steps:

- 1) Add ntport.lib to your project by selecting the menu "Project | Add To Project | Files";
- 2) Copy ntport.h to your project directory or your Visual C++ include directory;
- 3) Add following line to your C/C++ source code:

```
#include "ntport.h"
```

Please read Distribution Notes if you want to distribute NTPort Library with your Visual C++ project.

1.5 Use with Visual C#

NTPort Library can be used with Visual C# 2002/2003/2005.

Visual C# samples are located in NTPort\VCSharp directory.

<u>Samples</u>	<u>Purpose</u>
FastMode	Illustrates how to use SetFastMode, EnablePorts and DisablePorts procedures to increase the performance under Windows NT/2000/XP/Server 2003.
LPTPort	Illustrates how to get the base address of LPT ports.
PortTest	Illustrates how to read or write PC I/O ports.
Speaker	Illustrates how to control PC speaker.

If you want to use NTPort Library in a new Visual C# project, you should add NTPort.cs into your projects by selecting the menu "Project | Add Exist Items".

Please read Distribution Notes if you want to distribute NTPort Library with Visual C# project.

1.6 Use with Delphi

NTPort Library can be used with all versions of 32-bit Delphi.

Delphi samples are located in NTPort\Delphi directory.

<u>Samples</u>	<u>Purpose</u>
Class	Illustrates how to use TIO_Port class to simulate the Port array in Delphi 1.0 and Turbo Pascal.
FastMode	Illustrates how to use SetFastMode, EnablePorts and DisablePorts procedures to increase the performance under Windows NT/2000/XP/Server 2003.
LPTPort	Illustrates how to get the base address of LPT ports.
PortTest	Illustrates how to read or write PC I/O ports.
Speaker	Illustrates how to control PC speaker.

If you want to use NTPort Library in your Delphi project, you should add NTPort.pas in NTPort directory to your Delphi project by selecting the menu "Project | Add to Project".

Please read Distribution Notes if you want to distribute NTPort Library with your Delphi project.

1.7 Use with Delphi for .NET

NTPort Library can be used with all versions of Borland Delphi for .NET, including Delphi 8.0 and Delphi 2005/2006.

Borland Delphi for .NET samples are located in NTPort\Delphi.NET directory.

<u>Samples</u>	<u>Purpose</u>
FastMode	Illustrates how to use SetFastMode, EnablePorts and DisablePorts procedures to increase the performance under Windows NT/2000/XP/Server 2003.
LPTPort	Illustrates how to get the base address of LPT ports.
PortTest	Illustrates how to read or write PC I/O ports.
Speaker	Illustrates how to control PC speaker.

If you want to use NTPort Library in a new Borland Delphi for .NET project, you should add

Zeal.NTPort.pas into your projects by selecting the menu "Project | Add to Project".

Please read Distribution Notes if you want to distribute NTPort Library with Borland Delphi for .NET project.

1.8 Use with Borland C++ Builder

NTPort Library can be used with all versions of 32-bit Borland C++ Builder.

Borland C++ Builder 6.0 samples are located in NTPort\BCB directory.

<u>Samples</u>	<u>Purpose</u>
FastMode	Illustrates how to use SetFastMode, EnablePorts and DisablePorts procedures to increase the performance under Windows NT/2000/XP/Server 2003.
LPTPort	Illustrates how to get the base address of LPT ports.
PortTest	Illustrates how to read or write PC I/O ports.
Speaker	Illustrates how to control PC speaker.

If you want to use NTPort Library in your Borland C++ Builder project, you should do as following steps,

- 1) Add bcbport.lib to your project by selecting the menu "Project | Add to Project";
- 2) Copy ntport.h to your project directory or your Borland C++ Builder include directory;
- 3) Add following line to your C/C++ source code:

```
#include "ntport.h"
```

Please read Distribution Notes if you want to distribute NTPort Library with your Borland C++ Builder project.

1.9 Use with Borland C#Builder

NTPort Library can be used with all versions of Borland C#Builder, including C#Builder 1.0 and Delphi 2005/2006.

Borland C#Builder samples are located in NTPort\CSBuilder directory.

<u>Samples</u>	<u>Purpose</u>
FastMode	Illustrates how to use SetFastMode, EnablePorts and DisablePorts procedures to increase the performance under Windows NT/2000/XP/Server 2003.
LPTPort	Illustrates how to get the base address of LPT ports.
PortTest	Illustrates how to read or write PC I/O ports.
Speaker	Illustrates how to control PC speaker.

If you want to use NTPort Library in a new Borland C#Builder project, you should add NTPort.cs into your projects by selecting the menu "Project | Add to Project".

Please read Distribution Notes if you want to distribute NTPort Library with Borland C#Builder project.

1.10 Use with Borland JBuilder

NTPort Library can be used with Borland JBuilder.

Borland JBuilder samples are located in NTPort\JBuilder directory.

Samples	Purpose
FastMode	Illustrates how to use SetFastMode, EnablePorts and DisablePorts procedures to increase the performance under Windows NT/2000/XP/Server 2003.
LPTPort	Illustrates how to get the base address of LPT ports.
PortTest	Illustrates how to read or write PC I/O ports.
Speaker	Illustrates how to control PC speaker.

If you want to use NTPort Library in a new Borland JBuilder project, you should add NTPortJava.java into your projects by selecting the menu "Project | Add to Project".

Please read Distribution Notes if you want to distribute NTPort Library with Borland JBuilder project.

1.11 Contact Information

We welcome any comments or suggestions, etc. We can be reached via post:

Zeal SoftStudio.
No. 1607 Unit 133
Beijing Institute of Technology
Beijing 100081
P.R. China

or

E-Mail:

Products support questions go to... support@zealsoft.com
Sales inquiries and e-mail orders go to... info@zealsoft.com
Other comments and questions go to... haili@public.bta.net.cn

Home page of Zeal SoftStudio

<http://www.zealsoft.com>

Home page for this product

<http://www.zealsoft.com/ntport>

Shipping Your Application



Part

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2 Shipping Your Application

2.1 Distribution Notes

If you want to distribute your application with NTPort Library, you need register your copy.

NTPort Library includes a kernel mode driver, so it needs more steps than general applications to be distributed. We provide Merge Module for Windows Installer and stand-alone driver setup program, which are available only in the registered version, to simplify the process of making an installation disk.

You can choose one of following ways to distribute NTPort Library DLL and drivers with your application. For all ways, you still need administrative right to run the installation program, but after the reboot, any normal users can use your program.

- **Using Merge Module for Windows Installers 2.0**(registered version only)

If you are using a setup program which supports Windows Installer 2.0 technology, you can use this way. You can find `ntport27.msm` in `NTPort\MergeModule` directory. By default, this file is also copied into `<Program Files>\Common Files\Merge Modules` directory for your convenience. This merge module supports Windows x86 edition only. Please refer to `Using Merge Module for Windows Installers 2.0` for more details.

- **Using stand-alone driver setup program**(registered version only)

If your setup program supports executing external program, you can use this way. You can find `NTPortDrvSetup.exe` in `NTPort\Redist` directory. When you run this small setup program, all necessary DLLs and drivers will be copied to the user's system, and system configurations are made automatically. `NTPortJava.dll`(JNI wrapper DLL, only required for Java applications) is not included in this setup program. Please refer to `Using Stand-alone Driver Setup Program` for more details.

- **Using other setup programs**

For other setup programs which don't support executing external program, you need manually add files and set some options in your setup program to distribute NTPort Library DLL and drivers. Please refer to the guide to `Visual Basic 6.0 Package and Deployment Wizard` and `Wise InstallBuilder`, `InstallMaster` and `InstallManager 8.x`.

You can find all runtime files you need to distribute in `NTPort\Redist` directory. The primary redistributable components are:

- `Ntpport.dll` (Application Dynamic Link Library)
- `Zntport.sys` (Windows NT/2000/XP/Server 2003 NTPort Library driver)
- `NTPortJava.dll`(JNI wrapper DLL, only required for Java applications)

32-bit Windows (x86 edition) and Windows x64 edition have different version of `zntport.sys`. The `zntport.sys` in `NTPort\Redist` is 32-bit version, while `zntport.sys` in `NTPort\Redist\x64` directory is x64 edition. You need choose proper `zntport.sys` according to your target system.

When you install your application to a Windows 95/98/Me system, you should install `ntport.dll` to your client's `Windows\System` directory. You need not distribute the `zntport.sys`.

When you install your application to a Windows NT/2000/XP/Server 2003 system, you have two choices:

- 1) If all the users have administrative right, you can install `ntport.dll` and `zntport.sys` to your client's

WinNT\System32 directory. Please note, zntport.sys should be placed under WinNT\System32 directory. In this case, you need not any other configuration.

2) If some users don't have administrative right, you need create an installation program to install the NTPort Library driver. The installation program should do following steps:

- a. Install ntport.dll to WinNT\System32 directory
- b. Install zntport.sys to WinNT\System32\drivers directory
- c. Import the registry settings from ntport2.reg (You can find this file under NTPort\Redist directory)
- d. Reboot the computer (This step is required before running your application).
- e. For Java applications, you also need to distribute NTPortJava.dll to WinNT\System32\drivers directory.

In this case, you still need administrative right to run the installation program, but after the reboot, any normal user can use your program.

NTPort Library supports both methods internally, so you need not change your code no matter which method you use. The method two is comparatively complex, so if possible, we recommend you to use method one.

- **Remotely install the driver**

If you want to manually install NTPort Library, you can also use sc.exe from Windows NT/2000/XP/Server 2003 Resource Kit to do the above step c and d. See also to How to Remotely Install Driver.

More templates or guide to other installation tools may be available at <http://www.zealsoft.com/ntport/>.

2.2 Using Stand-alone Driver Setup Program

If your setup program supports executing external program, you can use this way. You can find NTPortDrvSetup.exe in NTPort\Redist directory. When you run this small setup program, all necessary DLLs and drivers will be copied to the user's system, and system configurations are made automatically. NTPortJava.dll(JNI wrapper DLL, only required for Java applications) is not included in this setup program.

Our stand-alone driver setup program does following steps during installing:

1. Copy NTPort.dll to user's System32 directory.
2. Copy zntport.sys to user's System32\drivers directory under Windows NT/2000/XP/Server 2003.
3. Create and start zntport service under Windows NT/2000/XP/Server 2003.

Our stand-alone driver setup program does following steps during uninstalling:

1. Stop and remove zntport service under Windows NT/2000/XP/Server 2003.
2. Remove NTPort.dll from user's System32 directory.
3. Remove zntport.sys from user's System32\drivers directory under Windows NT/2000/XP/Server 2003.

If you don't want any user intervention or user interface, please passing /S on the command line (case sensitive).

2.3 Using Merge Module for Windows Installer 2.0

Merge module for Windows Installer is only included in the registered version, not evaluation

version. This merge module is designed for Windows Installer 2.0 or higher. Although it works with almost all previous versions of Windows Installer in our tests, but we strongly suggest using it with Windows Installer 2.0 or higher.

The current version of merge module is 32-bit only, and is not supported under Windows XP/Server 2003 x64 edition, but Stand-alone Driver Setup Program supports both 32-bit and x64 editions.

NTPort27.msm can be found at NTPort\MergeModule directory. By default, this file is also copied into <Program Files>\Common Files\Merge Modules directory for your convenience.

Ntport.dll and zntport.sys are included in the merge module, so you need not add both files to your Windows Installer project. Administrative right is required for the installation.

Our merge module does following steps during installing:

1. Copy NTPort.dll to user's System32 directory.
2. Copy zntport.sys to user's System32\drivers directory under Windows NT/2000/XP/Server 2003.
3. Create and start zntport service under Windows NT/2000/XP/Server 2003.

Our merge module does following steps during uninstalling:

1. Stop and remove zntport service under Windows NT/2000/XP/Server 2003.
2. Remove NTPort.dll from user's System32 directory.
3. Remove zntport.sys from user's System32\drivers directory under Windows NT/2000/XP/Server 2003.

We test our merge module with some versions of following products:

- Microsoft Windows Installer SDK 2.0/3.0
- WiX (Windows Installer XML) toolset
- Visual Studio Installer 1.1 and .NET 2002/2003
- InstallShield Developer 7.0/8.0/9.0, InstallShield Express 4, InstallShield X Express, InstallShield 11 Express, InstallShield X to 11
- Wise for Windows Installer 5.0 or higher
- Installer2Go 4.0 or higher
- Dacris ActiSetup 2005

2.4 How to Remotely Install Driver

Windows NT/2000/XP/Server 2003 supports remotely installing and removing device drivers. You can use sc.exe from Windows NT/2000/XP/Server 2003 Resource Kit to start the device drivers. You can do as follows,

- Copy all system files to remote machine using Admin\$-Share
- Create zntport service with sc.exe
- Starts zntport service (driver) via sc.exe

Please refer to install.cmd file in NTPort Library for details. Our kind user Marco Teubert wrote this file.

32-bit Windows (x86 edition) and Windows x64 edition have different version of zntport.sys. The zntport.sys in NTPort\Redist is 32-bit version, while zntport.sys in NTPort\Redist\x64 directory is x64 edition. You need choose proper zntport.sys according to your target system.

2.5 How to Make Installation Disk

All following guides are provided only for your reference, we don't provide any official supports for these setup tools.

Before read following guides, we recommend you read Distribution Notes.

Dacris ActiSetup 2005
 Install2Go 4.0 or higher
 Visual Basic 6.0 Package and Deployment Wizard
 Visual Studio .NET
 Visual Studio Installer 1.1
 Wise for Windows Installer 5.0 or higher
 Wise InstallBuilder, InstallMaster and InstallManager 8.x

You can also download InstallShield Express for Delphi/Borland C++ Builder or InstallShield for Visual C++ demos from <http://www.zealsoft.com/ntport/>.

2.5.1 Dacris ActiSetup 2005

1. Use **Merge Module** section in **Dependencies** section to add the merge module.
2. The main list shows you all the merge modules found in the merge module search paths. Click a checkbox next to NTPort Library Merge Module(NTPortDrvModule) to include it in your installation.
3. Select the **Product Information** section in **General** section.
4. Select the **Launch condition** in the listview.
5. Enter **Privileged** as the launch condition. Using this condition, setup program will abort if the installation is not performed in the context of elevated privileges.
6. Enter "You must be an administrator to install this product." as the description in **Launch Condition Editor**. Of course, you can use your own words here.

2.5.2 Installer2Go

Use the Merge Modules page to add merge modules to an installation.

1. Select **Merge Modules** page.
2. Click **Add New Custom Merge Module** at the shortcut menu. The **Add New Custom Merge Module** dialog appears. Navigate to and select the **NTPort27.msm** to add to the installer project.

2.5.3 Visual Basic 6.0 Package and Deployment Wizard

In the step of Included Files, please check the Included Files List and make sure the ntport.dll and zntport.sys are added. If one or more files are missing, please click the Add button to choose these files. You also need add NTPort\Redist\Ntporvb6.reg file into this list. Ntporvb6.reg file includes all the registry information of Ntpor2.reg and extra settings that make your Setup program to reboot system. Follow the wizard and in the step of Install Locations, select ntport.dll in the table,

and select $\$(WinSysPath)$ from the Install Locations column, and set the destination location of zntport.sys to $\$(WinSysPath)\drivers$.

32-bit Windows (x86 edition) and Windows x64 edition have different version of zntport.sys. The zntport.sys in NTPort\Redist is 32-bit version, while zntport.sys in NTPort\Redist\x64 directory is x64 edition. You need choose proper zntport.sys according to your target system.

2.5.4 Visual Studio .NET

1. Open your Windows application.
2. In the **Solution Explorer**, right-click your Windows application solution, point to Add then click New Project.
3. In the **Add New Project** dialog box, select **Setup and Deployment Projects** in the Project Types pane and select **Setup Project** in the Templates pane. Click **OK**.
4. In the Solution Explorer, right-click the Setup Project, point to **Add** then click Project Output.
5. In the **Add Project Output Group**, select **Primary output** and click **OK**.
6. In the **Solution Explorer**, right-click the Setup Project, point to **Add** then click Merge Module.
7. In the **Add Modules** dialog box, select **NTPort27.msm** merge module to add to your Setup project and click **Open**.

2.5.5 Visual Studio Installer 1.1

1. In the **Project Explorer** hierarchy, select your Windows installer project.
1. On the **Project** menu, click **Add Merge Module(s)**.
Note You can also reach the **Browse for Merge Module File** dialog box by right-clicking the installer project name in the **Project Explorer**, highlighting **Add**, and then selecting **Merge Module(s)**.
3. In the **Browse for Merge Module File** dialog box, navigate to and select the NTPort26.msm to add to the installer project.
1. Click **Open**.
The NTPort27.msm file is added to your installer project and shown in the **Files** node in the **Project Explorer** window.

2.5.6 Wise for Windows Installer

Use the Merge Modules page to add merge modules to an installation.

1. Select Installation Expert > Merge Modules page.
2. From **Current Feature**, select a feature or condition. (Because any item you add must be assigned to a specific feature, you cannot add an item when **All Features** is selected.) Items you add to a feature are only installed on the destination computer if the feature is installed. Items you add to a condition are only installed if the feature is installed and the condition is true.
3. Click Add at the right of the Merge Modules page. The Select Merge Module dialog appears and lists available merge modules in the directories specified in Wise Options.
5. Select NTPort Library modules(NTPortDrvModule) from the list by marking the corresponding checkboxes.

6. Click Next. The Merge Module File Directory dialog appears.
7. Select **<none>** as the destination directory for files in the merge module, so that all files are installed to a predefined directory.
9. Click Finish to complete this process.
10. To add this merge module to other features besides the one you selected in the **Current Feature** drop-down list, select them in the list box. You can select multiple features. If multiple features in the installation depend on this merge module, you should add it to all of them. Only one copy of the merge module is installed on the destination computer regardless of how many features include it.
11. Click Finish.

Known issue: Under our test, Wise for Windows Installer sometimes change the destination directory of merge module to <system32> directory, which will cause a runtime error, "Can't start NTPort Library driver." Please make sure the destination directory to **<none>**, so that our merge module will work well. This problem is only occurred in Wise for Windows Installer.

2.5.7 Wise InstallBuilder, InstallMaster and InstallManager 8.x

You can include NTPort\Wise8\NTPort2.wse script into your installation program. This script supports only 32-bit Windows (x86 edition).

To access Script Editor, click the Script Editor button at the bottom of the window. Select the line in the script above which you want the new action to appear, then double-click the Insert Script action in the Actions list.

Enter the full pathname of the NTPort2.wse script, or use the Browse button to enter it using an Open dialog. The path should be on your computer, not the destination computer. Because the main script and the included scripts are combined into a single file at compile time, not runtime, you cannot use a runtime variable in the Pathname field to specify the Include Script. You can, however, use a compiler variable in this field.

This is an example of the line you might use in your script to include the script:

```
Include Script C:\Program Files\NTPort Library\Wise8\NTPort2.wse
```

Registration



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3 Registration

3.1 Registration Information

The non-evaluation use of NTPort Library in a business, commercial, government or institutional environment requires a license. Please visit <http://www.zealsoft.com/ntport/> for the latest price information. **The copyright dialog will not be shown in registered copy when application is running.** Registered users will obtain many services, such as FREE tech support through e-mail, more samples, *Zeal Tech Tips* articles, and so on.

After you register NTPort Library, the registration key and the registered copy will be sent to you. You can use it with LicenseInfo procedure in your program.

You are free to distribute this product separately from a product of your own, as long as you distribute both this file and the library files.

You can register this product through one of following ways,

On-line Web

Visit <http://order.kagi.com/?3SR> to complete online register process.

Bank Account

Send your registration fee directly to the following bank account and send the order form to info@zealsoft.com at the same time. In general, we will receive your money in 3 workdays. If we receive your money and order e-mail, we will send you the registered copy via e-mail in 24 hours.

INTERMEDIARY BANK'S NAME & SWIFT CODE:

FIRST UNION BANK INTERNATIONAL, NEW YORK
PNBP US3N NYC

INTERMEDIARY BANK'S CHIPS PARTICIPANT NO. :0509

ACCOUNT WITH BANK'S NAME & SWIFT CODE:

CHINA MERCHANTS BANK, BEIJING BRANCH
CMBC CNBS 201

ACCOUNT WITH BANK'S CHIPS UID: 343306

BENEFICIARY: LI Hai, A/C No.0010 15735988

If you have any problems about registration, please feel free to contact me at

Hai Li
Zeal SoftStudio.
No. 1607 Unit 133
Beijing Institute of Technology
Beijing, 100081
PR China

or

E-Mail: haili@public.bta.net.cn

Please visit our homepage at <http://www.zealsoft.com> to download the update version and other controls.

3.2 What If I Register

The registered users of NTPort Library will receive the registration key and registered copy of NTPort Library through e-mail in 24 hours after the on-line registration. You can use the registration

key and LicenseInfo function with the registered copy.

Benefits of the registered user

- Registered user can distribute the Library with applications to customers. The copyright dialog will not be shown in registered copy when application is running.
- Registered user can get FREE technical support through E-mail.
- Minor upgrades and bug fixes are free to registered users.
- Registered users are informed of upgrades and new software releases.



Programming Reference



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4 Programming Reference

4.1 Procedures and Functions

NTPort Library provides following procedures and functions:

EnablePorts procedure
DisablePorts procedure
GetFastMode function
GetLastState function
GetLPTPortAddress function
GetNTPortVersion function
GetFastMode function
IsWin64 function
IsWinNT function
Inp function
Inpd function
Inport function
InportD function
InportW function
Inpw function
LicenseInfo procedure
Outp procedure
Outpd procedure
Outport procedure
OutportD procedure
OutportW procedure
Outpw procedure
SetFastMode function

4.2 EnablePorts, DisablePorts procedure

EnablePorts procedure will enable your application to read or write specific ports. DisablePorts procedure will disable your application to read or write specific ports. Both functions are only required in the fast mode.

Syntax

Visual Basic

```
Declare Sub EnablePorts Lib "ntport.dll" (ByVal portStart As Integer, ByVal portEnd As Integer)
```

```
Declare Sub DisablePorts Lib "ntport.dll" (ByVal portStart As Integer, ByVal portEnd As Integer)
```

C++

```
void APIENTRY EnablePorts(WORD PortStart, WORD PortStop);
void APIENTRY DisablePorts(WORD PortStart, WORD PortStop);
```

Delphi

```
procedure EnablePorts(PortStart:word; PortStop:word);stdcall; external 'ntport.dll';
procedure DisablePorts(PortStart:word; PortStop:word);stdcall; external 'ntport.dll';
```

Remarks

Both functions will work only under Windows NT/2000/XP/Server 2003 (32-bit only). This function is not supported under Windows XP/Server 2003 x64 edition. Under Windows 95/98/Me, you need not call either function.

NTPort Library can run under two modes: normal mode and fast mode, under Windows NT/2000/XP/Server 2003. By default, it runs under normal mode. In this mode, if you use Inp, Inpd, Inport, InportD, InportW, Inpw, Outp, Outpd, Outport, OutportD, OutportW, or Outpw function, you need not call EnablePorts and DisablePorts function. If you use SetFastMode to let the library running under fast mode, you must call EnablePorts before you call any other functions of NTPort Library, and then call DisablePorts to free the resource when your application is quitting. Please refer to the FastMode sample for details.

If you use EnablePorts, the system will allow you read or write ports within the range from PortStart to PortEnd.

Both functions are provided for advanced users who want to use IN/OUT CPU instructions in their applications. Typically, users can use both functions with inline assembler code or third party components.

4.3 GetFastMode function

Returns the setting of fast mode.

Syntax**Visual Basic**

```
Declare Function GetFastMode Lib "ntport.dll" () As Long
```

C++

```
BOOL APIENTRY GetFastMode();
```

Delphi

```
function GetFastMode:LongBool;stdcall; external 'ntport.dll';
```

Remarks

If the GetFastMode function returns True, the fast mode is enabled. By default, the fast mode is disabled.

4.4 GetLastState function

Returns a value indicating the last operation is successful or not.

Syntax**Visual Basic**

```
Declare Function GetLastState Lib "ntport.dll" (ByVal sStatus As String) As Long
```

C++

```
int APIENTRY GetLastState(char *sStatus);
```

Delphi

```
function GetLastState(sStatus:PCHAR): longint;stdcall; external 'ntport.dll';
```

Remarks

You can call GetLastState function after you call any NTPort Library functions to read or write PC ports.

This function will return one of following values:

<u>Value</u>	<u>Meaning</u>
ERROR_SUCCESS	No error.
ERROR_DRIVER	Can't find zntport.sys under Windows NT/2000/XP/Server 2003. Please read Distribution Notes for more information.
ERROR_SCM_CANT_CONNECT	Can't dynamically load the zntport.sys. The user needs administrative right to dynamically load and unload the driver. If the user doesn't have such right, you need to modify some registry settings. Please read Distribution Notes for more information.

If sStatus is not NULL (0), sStatus will return the error message.

4.5 GetLPTPortAddress function

Returns a value indicating the base address of LPT ports.

Syntax**Visual Basic**

```
Declare Function GetLPTPortAddress Lib "ntport.dll" (ByVal portID As Integer) As Integer
```

C++

```
WORD APIENTRY GetLPTPortAddress(WORD portID);
```

Delphi

```
function GetLPTPortAddress(portID:word): word;stdcall; external 'ntport.dll';
```

Remarks

Parameter portID is the specific LPT port address. The portID for LPT1 is 1.

If the return value is zero, the port is absent or there is an error. If there is an error, call GetLastState function to get detail information.

LPT1 is normally assigned base address 378h, while LPT2 is assigned 278h(The lower case h denotes that it is in hexadecimal). But these addresses may change from machine to machine.

The base address, usually called the Data Port or Data Register is simply used for outputting data. The Status Port (base address + 1) is a read only port. Any data written to this port will be ignored. The Control Port (base address + 2) is intended as a write only port.

Known issue

This function works well with the LPT ports on the motherboard and most PCI parallel ports card, but it fails to retrieve the base address of some external cards.

4.6 GetNTPortVersion function

Returns the version of NTPort Library.

Syntax

Visual Basic

Declare Function GetNTPortVersion Lib "ntport.dll" () As Integer

C++

WORD APIENTRY GetNTPortVersion();

Delphi

function GetNTPortVersion:word;stdcall; external 'ntport.dll';

Remarks

Current version of NTPort Library will return 270H.

4.7 IsWin64 function

Returns a value indicates whether the application is running under 64-bit Windows system.

Syntax

Visual Basic

Declare Function IsWin64 Lib "ntport.dll" () As Boolean

C++

BOOL APIENTRY IsWin64();

Delphi

function IsWin64:boolean;stdcall; external 'ntport.dll';

Remarks

This function returns True(-1), if the application is currently running under 64-bit Windows. Under 32-bit Windows system, this function returns False(0).

This version includes 32-bit DLL, 32-bit(x86) driver and x64(AMD64 and EM64T) driver.

Itanium(IA64) will not be supported.

4.8 IsWinNT function

Returns a value indicates whether the application is running under Windows NT/2000/XP/Server 2003 system.

Syntax

Visual Basic

Declare Function IsWinNT Lib "ntport.dll" () As Boolean

C++

BOOL APIENTRY IsWinNT();

Delphi

function IsWinNT:boolean;stdcall; external 'ntport.dll';

Remarks

This function returns True(-1), if the application is currently running under Windows NT/2000/XP/Server 2003. Under Windows 95/98/Me, this function returns False(0).

In version 2.3 or higher, this function will return -1. While in previous version, the return value is 1. This change will be consistent with Visual Basic, and won't affect other language in most cases.

4.9 Inp, Inpw, Inpd function

Returns a value from specific ports.

Syntax

Visual Basic

Declare Function Inport Lib "ntport.dll" (ByVal portID As Integer) As Integer
 Declare Function InportW Lib "ntport.dll" (ByVal portID As Integer) As Integer
 Declare Function InportD Lib "ntport.dll" (ByVal portID As Integer) As Long
 Declare Function Inp Lib "ntport.dll" (ByVal portID As Integer) As Integer
 Declare Function Inpw Lib "ntport.dll" (ByVal portID As Integer) As Integer
 Declare Function Inpd Lib "ntport.dll" (ByVal portID As Integer) As Long

C++

WORD APIENTRY Inp(WORD PortNum);
 WORD APIENTRY Inpw(WORD PortNum);
 DWORD APIENTRY Inpd(WORD PortNum);
 WORD APIENTRY Inport(WORD PortNum);
 WORD APIENTRY InportW(WORD PortNum);
 DWORD APIENTRY InportD(WORD PortNum);

Delphi

function Inp(PortNum:word):word; stdcall; external 'ntport.dll';
 function Inpw(PortNum:word):word; stdcall; external 'ntport.dll';
 function InportW(PortNum:word):word; stdcall; external 'ntport.dll';
 function Inpd(PortNum:word):longint; stdcall; external 'ntport.dll';
 function Inport(PortNum:word):word; stdcall; external 'ntport.dll';
 function InportD(PortNum:word):longint; stdcall; external 'ntport.dll';
 function InportW(PortNum:word):word; stdcall; external 'ntport.dll';
 function InportD(PortNum:word):longint; stdcall; external 'ntport.dll';

Remarks

Inp and Inport will return a byte from the specific port. Inpw and InportW will return a word from specific ports. Inpd and InportD will return a double word from specific ports.

Parameter *PortNum* is the port address that you want to read from. For example, 61H is PC speaker control port.

If GetLastState function return a non-zero value, there is an error occurred and the return value of

Inp, Inpw or Inpd doesn't have any meaning.

4.10 Inport, InportD, InportW function

Returns a value from specific ports.

Syntax

Visual Basic

```
Declare Function Inport Lib "ntport.dll" (ByVal portID As Integer) As Integer
Declare Function InportW Lib "ntport.dll" (ByVal portID As Integer) As Integer
Declare Function InportD Lib "ntport.dll" (ByVal portID As Integer) As Long
Declare Function Inp Lib "ntport.dll" (ByVal portID As Integer) As Integer
Declare Function Inpw Lib "ntport.dll" (ByVal portID As Integer) As Integer
Declare Function Inpd Lib "ntport.dll" (ByVal portID As Integer) As Long
```

C++

```
WORD APIENTRY Inp(WORD PortNum);
WORD APIENTRY Inpw(WORD PortNum);
DWORD APIENTRY Inpd(WORD PortNum);
WORD APIENTRY Inport(WORD PortNum);
WORD APIENTRY InportW(WORD PortNum);
DWORD APIENTRY InportD(WORD PortNum);
```

Delphi

```
function Inp(PortNum:word):word; stdcall; external 'ntport.dll';
function Inpw(PortNum:word):word; stdcall; external 'ntport.dll';
function InportW(PortNum:word):word; stdcall; external 'ntport.dll';
function Inpd(PortNum:word):longint; stdcall; external 'ntport.dll';
function Inport(PortNum:word):word; stdcall; external 'ntport.dll';
function InportD(PortNum:word):longint; stdcall; external 'ntport.dll';
function InportW(PortNum:word):word; stdcall; external 'ntport.dll';
function InportD(PortNum:word):longint; stdcall; external 'ntport.dll';
```

Remarks

Inp and Inport will return a byte from the specific port. Inpw and InportW will return a word from specific ports. Inpd and InportD will return a double word from specific ports.

Parameter *PortNum* is the port address that you want to read from. For example, 61H is PC speaker control port.

If GetLastState function return a non-zero value, there is an error occurred and the return value of Inp, Inpw or Inpd doesn't have any meaning.

4.11 LicenseInfo procedure

Set the registration information.

Syntax

Visual Basic

```
Declare Sub LicenseInfo Lib "ntport.dll" (ByVal sUser As String, ByVal IKey As Long)
```

C++

```
void APIENTRY LicenseInfo(LPSTR sUserName, DWORD IKey);
```

Delphi

```
procedure LicenseInfo(sUserName: PChar; IKey: longint); stdcall; external 'ntport.dll';
```

Remarks

This function is only available in REGISTERED version, not EVAL version. When you register this product, you will receive a letter with the download link of the registered version.

You must call LicenseInfo procedure before you call any other functions of NTPort Library to avoid nag screen.

If you register NTPort Library, we will send you a registration key based on your registration name. Parameter *sUser* is your registration name, and Parameter *IKey* is your registration key that we send to you. Please note the registration name is case-sensitive.

4.12 Outp, outpw, outpd procedure

Write a value to specific ports.

Syntax**Visual Basic**

```
Declare Sub Outp Lib "ntport.dll" (ByVal portID As Integer, ByVal nByte As Integer)
Declare Sub Outpw Lib "ntport.dll" (ByVal portID As Integer, ByVal nWord As Integer)
Declare Sub Outpd Lib "ntport.dll" (ByVal portID As Integer, ByVal dwDWord As Long)
Declare Sub Outport Lib "ntport.dll" (ByVal portID As Integer, ByVal nByte As Integer)
Declare Sub OutportW Lib "ntport.dll" (ByVal portID As Integer, ByVal nWord As Integer)
Declare Sub OutportD Lib "ntport.dll" (ByVal portID As Integer, ByVal dwDWord As Long)
```

C++

```
void APIENTRY Outp(WORD PortNum, WORD Data);
void APIENTRY Outpw(WORD PortNum, WORD Data);
void APIENTRY Outpd(WORD PortNum, DWORD Data);
void APIENTRY Outport(WORD PortNum, WORD Data);
void APIENTRY OutportW(WORD PortNum, WORD Data);
void APIENTRY OutportD(WORD PortNum, DWORD Data);
```

Delphi

```
procedure Outp(PortNum:word; Data:word); stdcall; external 'ntport.dll';
procedure Outpw(PortNum:word; Data:word); stdcall; external 'ntport.dll';
procedure Outpd(PortNum:word; Data:longint); stdcall; external 'ntport.dll';
procedure Outport(PortNum:word; Data:word); stdcall; external 'ntport.dll';
procedure OutportW(PortNum:word; Data:word); stdcall; external 'ntport.dll';
procedure OutportD(PortNum:word; Data:longint); stdcall; external 'ntport.dll';
```

Remarks

Outp and Outport will write a byte from a specific port. Outpw and OutportW will write a word to specific ports. Outpd and OutportD will write a double word to specific ports.

Parameter *PortNum* is the port address that you want to write to. For example, 61H is PC speaker

control port. Parameter *Data* is the data that you want to write to the ports.

If `GetLastState` function return a non-zero value, there is an error occurred.

4.13 Outport, OutportW, OutportD procedure

Write a value to specific ports.

Syntax

Visual Basic

```
Declare Sub Outp Lib "ntport.dll" (ByVal portID As Integer, ByVal nByte As Integer)
Declare Sub Outpw Lib "ntport.dll" (ByVal portID As Integer, ByVal nWord As Integer)
Declare Sub Outpd Lib "ntport.dll" (ByVal portID As Integer, ByVal dwDWord As Long)
Declare Sub Outport Lib "ntport.dll" (ByVal portID As Integer, ByVal nByte As Integer)
Declare Sub OutportW Lib "ntport.dll" (ByVal portID As Integer, ByVal nWord As Integer)
Declare Sub OutportD Lib "ntport.dll" (ByVal portID As Integer, ByVal dwDWord As Long)
```

C++

```
void APIENTRY Outp(WORD PortNum, WORD Data);
void APIENTRY Outpw(WORD PortNum, WORD Data);
void APIENTRY Outpd(WORD PortNum, DWORD Data);
void APIENTRY Outport(WORD PortNum, WORD Data);
void APIENTRY OutportW(WORD PortNum, WORD Data);
void APIENTRY OutportD(WORD PortNum, DWORD Data);
```

Delphi

```
procedure Outp(PortNum:word; Data:word); stdcall; external 'ntport.dll';
procedure Outpw(PortNum:word; Data:word); stdcall; external 'ntport.dll';
procedure Outpd(PortNum:word; Data:longint); stdcall; external 'ntport.dll';
procedure Outport(PortNum:word; Data:word); stdcall; external 'ntport.dll';
procedure OutportW(PortNum:word; Data:word); stdcall; external 'ntport.dll';
procedure OutportD(PortNum:word; Data:longint); stdcall; external 'ntport.dll';
```

Remarks

`Outp` and `Outport` will write a byte from a specific port. `Outpw` and `OutportW` will write a word to specific ports. `Outpd` and `OutportD` will write a double word to specific ports.

Parameter *PortNum* is the port address that you want to write to. For example, 61H is PC speaker control port. Parameter *Data* is the data that you want to write to the ports.

If `GetLastState` function return a non-zero value, there is an error occurred.

4.14 SetFastMode procedure

Set the setting of fast mode.

Syntax

Visual Basic

```
Declare Sub SetFastMode Lib "ntport.dll" (ByVal bOption As Long)
```

C++

```
void APIENTRY SetFastMode(BOOL bOption);
```

Delphi

```
procedure SetFastMode(bOption:LongBool);stdcall; external 'ntport.dll';
```

Remarks

This function will work only under Windows NT/2000/XP/Server 2003 (32-bit only). This function is not supported under Windows XP/Server 2003 x64 edition. Under Windows 95/98/Me, you need not to call this function.

NTPort Library can run under two modes: normal mode and fast mode, under Windows NT/2000/XP/Server 2003. By default, it runs under normal mode. In this mode, if you use Inp, Inpd, Inport InportD, InportW, Inpw, Outp, Outpd, Outport, OutportD, OutportW, Outpw, you need not call EnablePorts and DisablePorts function.

But if you call Inp and Outp functions in a long loop, fast mode is generally faster than normal mode. You can use SetFastMode to switch between fast mode and normal mode. When the parameter bOption is set to True, the library will work under fast mode. In fast mode, you must call EnablePorts before you call any other functions of NTPort Library, and call DisablePorts to free the resource when your application is quit. Please refer to the FastMode sample for details.



Frequently Asked Questions



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5 Frequently Asked Questions

5.1 Frequently Asked Questions

Please visit <http://www.zealsoft.com/ntport/> for more FAQ.

1. How to convert old BASIC program to Visual Basic project?

Q: I have to convert a Qbasic program into a Visual Basic program. Now I got a problem with the Qbasic out command and I want to know if I can solve the problem with you product.

A: Yes. NTPort Library is an ideal replacement of old BASIC IN or OUT statement. You can change IN function to Inp function and OUT statement to Outp sub procedure.

2. How to increase the performance under Windows NT/2000/XP/Server 2003?

Q: I use Inport and Outport functions, but it seemed both functions are slow when I call them in a long loop under Windows NT/2000. Is there a way to speed up?

A: If you use Inport and Outport in a long loop, please try **SetFastMode**, **EnablePorts** and **DisablePorts** functions that can increase the performance under Windows NT/2000/XP/Server 2003 (32-bit only). There is a sample called FastMode to demonstrate how to use these functions. The fast mode is not supported under Windows XP/Server 2003 x64 edition.

3. Does NTPort Library provide the feature similar to Peek and Poke?

Q: I need to convert an old BASIC program, which used Peek and Poke. Does your library support these functions?

A: NTPort Library currently doesn't support memory mapped I/O, but we have another product, MemAccess Library, which is designed to direct access to physical memory. Please visit <http://www.zealsoft.com/memaccess/> to download it.

4. Does NTPort Library support .NET platform?

Q: I need to migrate my product to .NET platform. Does NTPort Library support Microsoft .NET?

A: NTPort Library is fully tested under .NET platform, and the Visual Basic.NET (2002/2003), Visual C#.NET, Borland C#Builder samples are included. But the DLL file, ntport.dll, is an unmanaged DLL. Please refer to Using with Visual Basic.NET and Using with Visual C#, Using with Delphi for .NET and Using with C#Builder.

5. What's the difference between Inp and Inport function?

Q: I notice that there are Inp and Inport function in your library. What's difference?

A: We provide pairs of functions with same feature to help programmers convert their old programs with less source code modification. We recommend you use Inp and Outp, because they are slightly fast compared to Inport and Outport.

6. Do I need zntport.sys under Windows 95/98/Me?

Q: I want to use your product under Windows 95/98/Me. Do I need zntport.sys?

A: No, you needn't. The driver zntport.sys is required only under Windows NT/2000/XP/Server 2003.

7. How many files do I need to copy to my client's machine?

Q: I copied my program and ntport.dll to my client's Windows NT machine, but it failed to work. After I installed your NTPort Library package, my program worked fine. What files do I need to copy?

A: You can find all runtime files you need to distribute in NTPort\Redist directory. Under Windows NT/2000/XP/Server 2003, you need to distribute zntport.sys. Please read Distribution Notes for details.

8. Can I use NTPort Library to enable third party components to work under Windows

NT/2000/XP?

Q: I have a third party component, which can work under Windows 95/98/Me, but it will cause privileged instruction exception under Windows NT/2000/XP. I ensure the component only accesses PC I/O ports. Can I use NTPort Library to enable my component to work under Windows NT/2000/XP?

A: If the component only accesses PC I/O ports and you know which ports it uses, you can call **EnablePorts** function before you call the component's functions and call **DisablePorts** when your program will end.

9. Do I need to log as an administrator when I use NTPort Library?

Q: Do I need to log as an administrator when I use NTPort Library under Windows NT/2000? We do not want every person who uses our application to have full authority to access every part of the operating system.

A: From the version 2.1, NTPort Library supports both administrator and non-administrator account, but administrator privileges are required to install NTPort Library on Windows NT/2000/XP/Server 2003. Because Windows NT/2000/XP/Server 2003's limitation, you need more system configuration if the user doesn't have administrative right. Please read Distribution Notes for details.

10. What's the meaning of error message "Can't connect to Service Control Manager"?

Q: I want to test your NTPort Library, if it is the right for my application. But on an NT4 machine I become following error message, after click the Inport- or Outport button in your Porttest program: "Cant connect to Service Control Manager". Can you tell me what's wrong?

A: Obviously, you don't have administrative right. From the version 2.1, NTPort Library supports both administrator and non-administrator account, but because Windows NT/2000/XP limitation, you need more system configuration if the user don't have administrative right. Please read Distribution Notes for more information. You can also refer to **GetLastState** function about the meaning of other error messages.

11. Can I use NTPort Library in a 16-bit application?

Q: Can I use NTPort Library in a 16-bit application written with Borland C++ v3.1?

A: We no longer support 16-bit applications. If you want to use NTPort Library in a 16-bit application, please use Automation or thunk technology. Please refer to following articles in Microsoft Knowledge Base,

- [HOWTO: all?2-bit ode rom?6-bit ode nder indows?5](#) (Q154093)

12. Does NTPort Library support 64-bit Windows?

Q: I want to develop for Windows XP x64 edition. Does your driver support it?

A: 64-bit Windows requires 64-bit driver. NTPort Library supports Windows XP/Server 2003 x64 edition for both AMD64 and Intel EM64T, but won't support Intel Itanium(IA64) platform. Both 32-bit(x86) driver and x64 driver are included in this version, but only 32-bit DLL is included. We will provide x64 DLL in the future version. Most features are same in 32-bit edition and x64 edition, but fast mode is not supported in x64 edition yet.

History



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6 History

6.1 Version History

- 2.7 Add Windows XP/Server 2003 x64 edition driver.
Fix the bug that calling IsWin64 function will cause access violation error(#40048).
Support Borland Developer Studio 2006.
Support Visual Studio 2005.
- 2.6.1 Fix the bug that Setup program can't start NTPort Library Driver, Error: 1073(#40040)
- 2.6 GetLPTPortAddress function supports MosChip/NetMos PCI parallel ports cards.
Add IsWin64 function.
Add stand-alone driver setup program(Registered copy only).
Add print manual(.pdf).
Add Delphi for .NET samples.
Add C#Builder samples.
Add JBuilder samples.
Update FastMode sample.
- 2.5 Add merge module for Windows Installer 2.0(Registered version only).
Fix the issue that causes a first-chance exception in Visual C#.
Fix the issue that GetLPTPortAddress function returns incorrect value in some Windows 9x system.
- 2.4 Fix the issue that displays a first chance exception in VC++ Debug Output window
Add Visual Basic.NET samples
Add Visual C# samples
Add Visual C++ Console sample
- 2.3 Add GetLPTPortAddress function
Increase the performance under all platforms and all modes
Add LPTPort sample
- 2.2 Add SetFastMode and GetFastMode functions to increase the performance
Add FastMode sample
Add Delphi ClassTest sample
- 2.1 Support both administrator and non-administrator accounts
Add Borland C++ Builder samples
Add several installation templates and demos
- 2.0 Increase the performance under Windows NT/2000
Support Windows 2000
- 1.0 Initial Release

6.2 Upgrade Issue

If you are the registered user of previous version of NTPort Library, please notice that the evaluation version of NTPort Library can't be turned to the registered version even if you have a valid serial number. Please visit <http://www.zealsoft.com/upgrade/> to download the upgrade version. Before install the upgrade version, you must install the previous registered copy of NTPort Library.

NTPort Library 2.7 is fully compatible with previous version, so you need not to recompile your project.

The IsWinNT function will return -1 if the application is running on Windows NT/2000/XP/Server

2003. In the version prior to 2.3, the return value is 1.

Merge module of this version is renamed to NTPort27.msm.



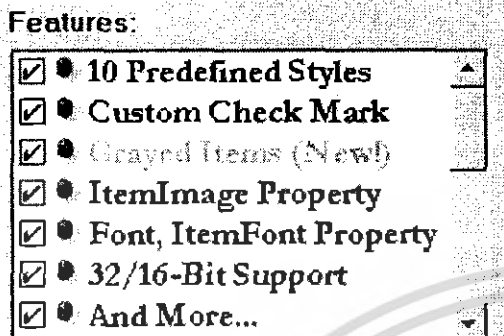
Other Products



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7 Other Products

7.1 CheckListBox ActiveX Control

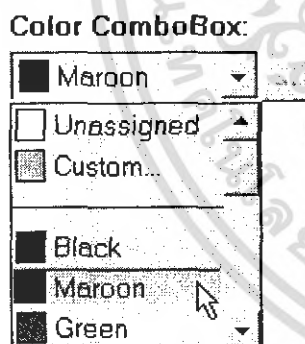


CheckListBox ActiveX Control provides support for listbox containing checkboxes as well as the same control that is built into *Microsoft Word and Excel*. In this control, each item's font, color, enabled and checked state can be individually modified. The control also supports 3D look, picture items and custom check mark. You can turn on/off the checkmark. The package includes samples written in *Visual Basic, Visual Basic.NET, Visual C#, C#Builder, Visual C++ and Internet Explorer*. *CheckListBox* is also called *Check List*.

CheckListBox ActiveX Control is *ActiveX.Com* and *SoftSeek.Com* editors' Top Pick.

You can download FREE trial version from <http://www.checklistbox.com>.

7.2 Color ComboBox ActiveX Control



Color ComboBox ActiveX Control offers an easy method for a user to select a color from a combo box based on a predefined color-set or to define a custom color.

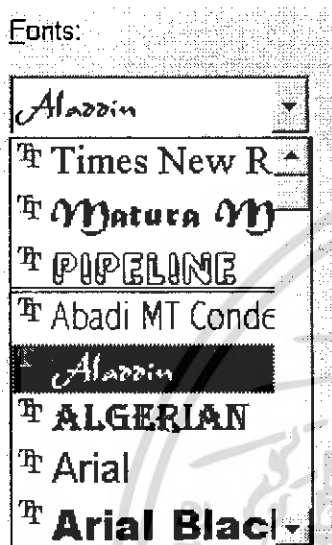
Features

- Displays built-in default or user-definable color list
- Several predefined styles
- Optional edit button for custom colors
- Support for automatically changing color entry
- Built-in ChooseColor dialog access

- Find a color or a string with easy
- Easy to use even without one line code
- Special selected item color and 3D effect
- User-definable color bar width and height

You can download FREE trial version from <http://www.zealsoftstudio.com>.

7.3 Font List & Combo ActiveX Control



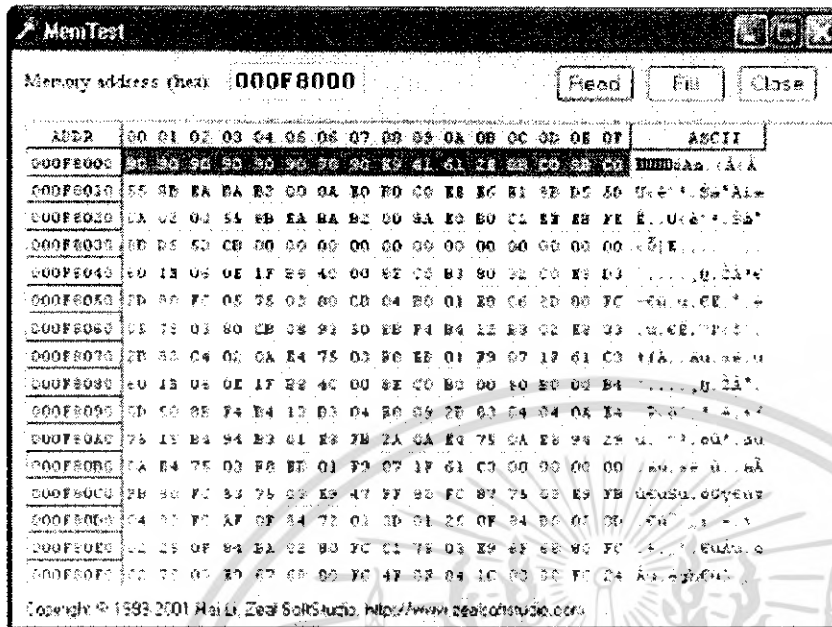
Font List & Combo ActiveX Control allows you choose all the available fonts from a listbox or combobox.

Main features include

- both ListBox and ComboBox control in one ActiveX control.
- automatic enumeration of all available fonts on your system, either the screen or the printer
- previewing each font name with its own font characteristics
- built-in recently used fonts sub-list
- draw a logo in front of each font names indicating font type
- getting and setting currently selected font name

You can download FREE trial version from <http://www.zealsoftstudio.com>.

7.4 MemAccess Library



MemAccess Library gives programmers control of hardware devices from Win32 application without using the Windows Drivers Development Kit (DDK). The tool allows real-time direct access to memory mapped I/O. Typical applications include ISA or PCI bus devices. MemAccess Library is also an ideal replacement of old BASIC Peek or Poke statement. MemAccess Library provides support for Windows 95/98/Me and Windows NT/2000/XP/Server 2003.

You can download FREE trial version from <http://www.zealsoft.com/memaccess/>.

Index

- . -

.NET 4, 5, 6, 7

- A -

Address 8

- B -

base address 22
 Borland C#Builder 7
 Borland C++ Builder 7
 Borland JBuilder 8

- C -

C# 5, 7
 C++ 5, 7
 CheckListBox 36
 Color ComboBox 36
 Contact Information 8

- D -

Delphi 6
 Delphi for .NET 6
 DisablePorts procedure 20, 27
 Distribution 10, 12, 13
 Distribution Rights 17

- E -

E-Mail 8, 17
 EnablePorts procedure 20, 27

- F -

FAQ 30
 Font List & Combo 37
 Function 20

- G -

GetFastMode function 21
 GetLastState function 21
 GetLPTPortAddress function 22
 GetNTPortVersion function 23

- H -

Home Page 8, 17

- I -

Inp 24, 25
 Inpw , inpd , Inport , InportD , InportW function
 24, 25
 Installation 10, 12, 13
 IsWinNT function 23

- J -

Java 8
 JBuilder 8

- L -

LicenseInfo procedure 25
 LPT ports 22

- M -

MemAccess Library 38
 Merge Module for Windows Installer 2.0 11

- O -

Outp 26, 27
 Outpw , Outpd , Outport , OutportD , OutportW
 procedure 26, 27

- P -

Peek 38
 Poke 38

Procedure 20
products 36, 37, 38

- R -

Registration 17
Registration Information 17

- S -

Service 17
SetFastMode procedure 27
Sub 20

- U -

Upgrade 33
Use with Borland C#Builder 7
Using with Visual Basic.Net 4
Using with Visual C# 5

- V -

Visual Basic 4
Visual C++ 5

- W -

Windows NT 23

