

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

**หุ่นยนต์กู้ภัย II
(RESCUE ROBOT II)**

นายชัชวาล ฤกษ์สุนทร
นายณรงค์ อินตาทอง

เลขหมู่.....
เลขทะเบียน.....**62329**
วัน,เดือน,ปี.....**16 ส.ค. 2549**

b.....**11621205**
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมสารสนเทศ
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RESCUE ROBOT II

BY

Mr. CHATWAN

KHUNKUUNTOD

Mr. NARONG

INTAPHUNG




**A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENT FOR THE DEGREE OF
BACHELOR IN DEPARTMENT OF INFORMATION ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

2004

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | |
|-------------------|--|
| หัวข้อวิทยานิพนธ์ | หุ่นยนต์กู้ภัย II |
| THESIS TITLE | RESCUE ROBOT II |
| ชื่อนักศึกษา | นายชัชวาล คุณขุนทด รหัสประจำตัว 46015622 |
| | นายณรงค์ อินตาพวง รหัสประจำตัว 46015623 |
| อาจารย์ที่ปรึกษา | รศ.ดร.ปิติเขต สุรักษา ผศ.บุญยัชนะ ภูระหงษ์ |
| ระดับการศึกษา | ปริญญาตรีวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมสารสนเทศ |
| ภาควิชา | วิศวกรรมสารสนเทศ |
| ปีการศึกษา | 2548 |

ปริญญาานิพนธ์ฉบับนี้ได้รับการอนุมัติให้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิศวกรรมศาสตรบัณฑิต คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร
ลาดกระบัง



(รศ.ดร.ปิติเขต สุรักษา)

อาจารย์ผู้ควบคุมปริญญาานิพนธ์



(ผศ.บุญยัชนะ ภูระหงษ์)

อาจารย์ผู้ควบคุมปริญญาานิพนธ์

ลิขสิทธิ์ของคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์

หุ่นยนต์กู้ภัย II

THESIS TITLE

RESCUE ROBOT II

ชื่อนักศึกษา

นายชัชวาล ฤกษ์สุนทร

รหัสประจำตัว 46015622

นายฉรรงค์ อินดาพวง

รหัสประจำตัว 46015623

อาจารย์ผู้ควบคุมวิทยานิพนธ์

รศ.ดร.ปิติเขต สุธีรักษา

ศศ.บุญจันทร์ ภูระหงษ์

ระดับการศึกษา

ปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชา

วิศวกรรมสารสนเทศ

ปีการศึกษา

2548

บทคัดย่อ

โครงการนี้เป็นกรนำเอาสัญญาณจากระบบกล้อง ตัวตรวจจับอุณหภูมิ คอมพิวเตอร์และระบบเชิงทศประยุกต์ใช้เป็นระบบการระบบนำร่องของหุ่นยนต์ ตรวจวัดอุณหภูมิ และสร้างแผนที่ซึ่งการทำงานจะแยกออกเป็น สามส่วน คือ หนึ่งระบบการประมวลผลสัญญาณภาพจากกล้องโดยจะนำเอาสัญญาณมาประมวลผลเพื่อตรวจจับวัตถุที่มีการเคลื่อนไหวในส่วนที่สองเป็นการตรวจวัดอุณหภูมิว่าพื้นที่ที่ตรงนั้นมีอุณหภูมิเท่าใดและรู้ว่าหุ่นยนต์อยู่ ณ ตำแหน่งใดจากข้อมูลทั้งหมดหุ่นยนต์จะวิเคราะห์ได้ว่ามีวัตถุใดขวางและลักษณะเช่นใดมีอุณหภูมิเท่าไรส่วนที่สามคือหุ่นยนต์สามารถข้ามสิ่งกีดขวางได้และเข้าไปในพื้นที่ที่อันตรายได้โดยการคอนโทรลจากโปรแกรมคอมพิวเตอร์

| | | | |
|---------------------|-------------------------------------|-------------|--------------|
| THESIS TITLE | RESCUE ROBOT II | | |
| STUDENT | Mr. Chatchawan | Khunkuuntod | No. 46015622 |
| | Mr. Narong | Intaphung | No. 46015623 |
| ADVISOR | Assoc.Prof. Dr.Pitikhate Sooraksa | | |
| | Asst.Prof.Boonchana Purahong | | |
| COURSE | Bachelor of Information Engineering | | |
| DEPARTMENT | Information Engineering | | |
| YEAR | 2005 | | |

Abstract

This project proposes utilization of signal from camera system, sensor temperature, computer and compass system to apply for surveying area to navigation system of robot separated in three parts .First is system image processing, video signals are used in motion detection process. Second our robot can check temperature where it's in there, and can know its position by create into map that use in the surveying route system. Third our robot can shun any roadblocks and barricade by human control from computer-monitor.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบคุณอาจารย์ที่ปรึกษาโครงการ รศ.ดร.ปิติเขต สุรักษา และผศ.บุญยชนะ ภูระหงษ์ เป็นอย่างสูงและ ขอขอบคุณ พี่แมว ที่ให้คำปรึกษาในส่วนของโปรแกรม Vision ซึ่งเป็นส่วนหลักของโครงการนี้ และขอขอบคุณ พี่คิว พี่เต่า ที่ให้คำปรึกษาในส่วนของงานด้านฮาร์ดแวร์และอื่น พี่ป๊องที่ให้คำปรึกษาในส่วนของเทคนิคการเขียนโปรแกรม

ขอบคุณคุณแม่และอาจารย์ทุกท่านที่ให้การสนับสนุนเสมอมาและเพื่อนๆ พี่ๆ น้องๆ ที่คอยช่วยเหลือและให้กำลังใจกันตลอดการทำงาน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

| | หน้า |
|--|------|
| บทคัดย่อภาษาไทย | ก |
| บทคัดย่อภาษาอังกฤษ | ข |
| กิตติกรรมประกาศ | ค |
| สารบัญ | |
| สารบัญรูป | |
| สารบัญตาราง | |
| บทที่ 1 บทนำ | |
| 1.1 ความเป็นมาและความสำคัญของปัญหา | 1 |
| 1.2 วัตถุประสงค์ | 1 |
| 1.3 ขอบเขตของโครงการ | 1 |
| 1.4 ผลที่คาดว่าจะได้รับ | 2 |
| บทที่ 2 ทฤษฎีและเอกสารของงานวิจัยที่เกี่ยวข้อง | |
| 2.1 ทฤษฎีไมโครคอนโทรลเลอร์ | 3 |
| 2.1.1 คุณสมบัติของไมโครคอนโทรลเลอร์ | 3 |
| 2.1.2 การใช้งาน Timer/Counter | 5 |
| 2.1.3 ทฤษฎีการสื่อสารผ่านพอร์ตอนุกรม | 6 |
| 2.1.4 การรับส่งข้อมูลแบบอนุกรม | 8 |
| 2.1.5 อัตราการส่งข้อมูลของพอร์ตอนุกรม | 9 |
| 2.2 การใช้เข็มทิศแบบดิจิทัล CMPS03 | 10 |
| 2.2.1 คุณสมบัติของเข็มทิศดิจิทัล CMPS03 | 11 |
| 2.3 พื้นฐานการประมวลผลภาพ | 12 |
| 2.3.1 ประเภทการกระทำภาพ (Types of Image Operation) | 13 |
| 2.3.2 พื้นฐานความสัมพันธ์ระหว่างพิกเซล (Basic Relation between Pixels) | 15 |
| 2.3.3 สัญญาณรบกวนในรูปภาพดิจิทัล(Noise Digital Image) | 16 |
| 2.3.4 การทำคอนโวลูชัน (Convolution) | 17 |
| 2.3.5 การกรองสัญญาณภาพในโดเมนเวลา (Image Filtering in Time Domain) | 19 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

| | หน้า |
|--|------|
| 2.4 หลักการทำงานดีซีมอเตอร์ | 22 |
| 2.5 การเคลื่อนที่ของ Object | 23 |
| 2.6 วงจรตรวจจับอุณหภูมิ | 26 |
| 2.6.1 หลักการเบื้องต้นของไอซี DS1820 | 27 |
| 2.6.2 การอินเตอร์เฟสผ่านสายเส้นเดียว | 27 |
| 2.6.3 คุณสมบัติ | 27 |
| บทที่ 3 การออกแบบและวิธีการดำเนินงานวิจัย | |
| 3.1 การออกแบบการทำงานของระบบ | 33 |
| 3.1.1 การออกแบบทางด้านซอฟต์แวร์ User Interface | 33 |
| 3.1.2 ส่วน Device Control Unit | 37 |
| 3.2 อุปกรณ์ที่ใช้ทดลอง | 41 |
| 3.3 หลักการของการเขียนโปรแกรมขับดีซีมอเตอร์ | 43 |
| 3.4 การควบคุมและใช้เข็มทิศ | 44 |
| 3.5 การแสดงอุณหภูมิ | 44 |
| 3.6 การใช้งาน Component ของกล้อง | 45 |
| บทที่ 4 ผลการทดลอง | |
| 4.1 ผลการทดลองตามวัตถุประสงค์หลัก | 47 |
| 4.1.1 การอ่านค่าจากระบบเข็มทิศ | 47 |
| 4.1.2 ส่วนของการใช้งานกล้องผ่านทาง VideoOCX | 48 |
| 4.2 การข้ามสิ่งกีดขวางของหุ่นยนต์ | 50 |
| 4.3 การแสดงอุณหภูมิ | 52 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

| | หน้า |
|-------------------------------|------|
| บทที่ 5 บทสรุปและข้อเสนอแนะ | |
| 5.1 ผลสรุปและปัญหางานวิจัย | 53 |
| 5.2 ข้อเสนอแนะและการพัฒนาระบบ | 54 |
| บรรณานุกรม | |
| ภาคผนวก | |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

| รูป | หน้า |
|--|------|
| รูปที่ 2-1 รายละเอียดโครงสร้างหลักของไมโครคอนโทรลเลอร์ | 4 |
| รูปที่ 2-2 การจัดขามาตรฐานของไมโครคอนโทรลเลอร์ MCS-51 ในอนุกรม AT89C5X | 5 |
| รูปที่ 2-3 บิตต่างๆในรีจิสเตอร์ TMOD | 5 |
| รูปที่ 2-4 บิตต่างๆในรีจิสเตอร์ TCON | 6 |
| รูปที่ 2-5 การส่งข้อมูลแบบอนุกรม | 7 |
| รูปที่ 2-6 บิตต่างๆของการรับส่งข้อมูลแบบอนุกรม | 8 |
| รูปที่ 2-7 บิตต่างๆของรีจิสเตอร์ SCON | 9 |
| รูปที่ 2-8 การเชื่อมต่อ MCS-51 กับ IC MAX232A เพื่อการรับส่งข้อมูลแบบอนุกรม | 10 |
| รูปที่ 2-9 ลักษณะภายนอกของ เซมิทิสแบบดิจิทัล CMPS03 | 11 |
| รูปที่ 2-10 ขาที่ต่อใช้งาน เซมิทิสแบบดิจิทัล CMPS03 | 11 |
| รูปที่ 2-11 การต่อเข้ากับอุปกรณ์เซมิทิสแบบดิจิทัล CMPS03 | 12 |
| รูปที่ 2-12 การเกิดแรงบิดในตัวดิซิมอเตอร์ | 22 |
| รูปที่ 2-13 สัญลักษณ์ของ Object ที่ใช้แทน Robot | 23 |
| รูปที่ 2-14 การหมุนทิศทางของเส้นตรง | 23 |
| รูปที่ 2-15 การเคลื่อนที่ของเส้นตรงตามทิศทางของลูกศร | 24 |
| รูปที่ 2-16 การเคลื่อนที่ของวงกลมตามทิศทางของลูกศร | 25 |
| รูปที่ 2-17 วงจรตรวจจับอุณหภูมิ | 26 |
| รูป 2-18 แผนผังของระบบบัสแบบ 1-Wire Bus | 27 |
| รูป 2-19 จังหวะเวลาในการทำกระบวนการตรวจสอบว่ามีอุปกรณ์อยู่บนบัส | 28 |
| รูป 2-20 จังหวะเวลาที่ไมโครคอนโทรลเลอร์ใช้เขียนข้อมูล '0' หรือ '1' ไปยังอุปกรณ์บนบัส | 29 |
| รูป 2-21 จังหวะเวลาที่ไมโครคอนโทรลเลอร์ใช้อ่านบิตข้อมูล จากอุปกรณ์บนบัส | 30 |
| รูปที่ 3-1 Application User Control and Display | 32 |
| รูปที่ 3-2 ส่วนบอกอุณหภูมิของหุ่นยนต์ | 33 |
| รูปที่ 3-3 ส่วนของการควบคุมการเคลื่อนที่และองศา | 33 |
| รูปที่ 3-4 ส่วนของการปรับภาพ | 33 |
| รูปที่ 3-5 ส่วนของแผนที่ที่หุ่นยนต์เคลื่อนที่ | 34 |
| รูปที่ 3-6 ส่วนของจอภาพ | 34 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

| รูป | หน้า |
|---|------|
| รูปที่ 3-7 Block User Control and Display | 35 |
| รูปที่ 3-8 Flow Chart ส่วนของDevice Control Unit | 36 |
| รูปที่ 3-9 Device Control Unit | 37 |
| รูปที่ 3-10 Flow Chart ของส่วน User Control & Display Unit | 39 |
| รูปที่ 3-11 โครงหุ่นยนต์ | 40 |
| รูปที่ 3-12 ชุดเครื่องรับ- ส่งสัญญาณวิทยุ | 41 |
| รูปที่ 3-13 วงจรควบคุม และรับส่งสัญญาณพร้อมวงจรขับมอเตอร์ | 41 |
| รูปที่ 4-1 ผลการทดสอบโปรแกรมโดยรวมของส่วน User Control & Display Unit | 46 |
| รูปที่ 4-2 การอ่านค่าจากเซ็นเซอร์และบอกพิกัดบนแผนที่ | 47 |
| รูปที่ 4-3 Method ของการเซต Driver ของกล้อง | 48 |
| รูปที่ 4-4 หน้าจอของโปรแกรมเมื่อทำการปิดกล้อง | 49 |
| รูปที่ 4-5 หน้าจอของโปรแกรมเมื่อทำการปิดกล้อง | 49 |
| รูปที่ 4-6 หุ่นยนต์ข้ามสิ่งกีดขวางด้วยดินตะขามด้านขวา | 50 |
| รูปที่ 4-7 หุ่นยนต์ข้ามสิ่งกีดขวางด้วยดินตะขามด้านซ้าย | 50 |
| รูปที่ 4-8 หุ่นยนต์ขึ้นบันได | 51 |
| รูปที่ 4-9 หุ่นยนต์ลงบันได | 51 |
| รูปที่ 4-10 ทียบอุณหภูมิระหว่างตัวหุ่นยนต์กับค่าที่ส่งแสดงผลหน้าจอ | 52 |

สารบัญตาราง

ตาราง

หน้า

ตารางที่ 2-1 ตัวอย่างความถี่สัญญาณนาฬิกาที่ใช้กำหนด Baud Rate ค่าต่างๆ

10



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องจากในปัจจุบันในทางด้านอุตสาหกรรมได้มีการนำหุ่นยนต์มาใช้งานมากขึ้น จึงจำเป็นต้องมีการพัฒนาความสามารถของหุ่นยนต์ให้มีความสามารถมากขึ้น เพื่อจะตอบสนองความต้องการในการใช้งานของมนุษย์ โดยจะเห็นได้ว่าหุ่นยนต์จะช่วยอำนวยความสะดวกได้หลายอย่าง ซึ่งในอดีตไม่มีหุ่นยนต์มาใช้งานในการแก้ปัญหาจะเกิดความยุ่งยาก และซับซ้อนต่าง ๆ จากนั้นเมื่อมีหุ่นยนต์แล้ว จึงก็ต้องมีระบบในการควบคุมหุ่นยนต์ซึ่งเปรียบเสมือนกันเป็นส่วนของการคิดที่ใช้ในการทำงานหรือควบคุมส่วนต่าง จะเห็นว่าส่วนควบคุมนี้สำคัญอย่างยิ่งในการทำงานให้สัมฤทธิ์ผลตามต้องการและสามารถให้ผู้ใช้ดำเนินการต่อภารกิจต่อไปได้ ซึ่งถ้าขาดส่วนนี้ไปก็ไม่สามารถที่จะทำงานให้สำเร็จได้ตามเป้าหมายที่กำหนด

1.2 วัตถุประสงค์

- 1.2.1 เพื่อศึกษาและพัฒนากระบวนการนำร่องของหุ่นยนต์กู้ภัย (Rescue Robot)
- 1.2.2 เพื่อศึกษาและวิเคราะห์ตำแหน่งเพื่อสร้างแผนที่โดยระบบ เจ็มทิส
- 1.2.3 เพื่อศึกษาและพัฒนาทางด้านโครงสร้างของหุ่นยนต์ได้ทุกสภาพการ
- 1.2.4 เพื่อวัดอุณหภูมิสิ่งแวดล้อมรอบตัวของหุ่นยนต์กู้ภัย

1.3 ขอบเขตของโครงการ

- 1.3.1 ในส่วนของการประมวลผลสัญญาณจากระบบเจ็มทิส สามารถนำสัญญาณจากระบบ เจ็มทิส มาประมวลผลและวิเคราะห์เพื่อสร้างแผนที่ให้หุ่นยนต์กู้ภัย (Rescue Robot) ได้
- 1.3.2 ในส่วนของการสร้างสามารถเคลื่อนที่เข้าไปในพื้นที่ที่เข้าถึงได้ยากและสามารถตรวจจับวัตถุที่กำลังเคลื่อนไหวได้โดยใช้ทฤษฎีการประมวลผลภาพ ด้วย (Image Processing)
- 1.3.3 ในส่วนของการวัดอุณหภูมิสามารถวัดอุณหภูมิของสิ่งแวดล้อมรอบตัวของหุ่นยนต์กู้ภัย (Rescue Robot) ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ผลที่คาดว่าจะได้รับ

1.4.1 ระบบของการประมวลผลและวิเคราะห์ตำแหน่งสร้างแผนที่สามารถนำไปใช้กับหุ่นยนต์กู้ภัย(Rescue Robot) ได้

1.4.2 ระบบของการประมวลผลและการตรวจจับวัตถุที่กำลังเคลื่อนไหวนำไปใช้กับหุ่นยนต์กู้ภัย(Rescue Robot) ได้

1.4.3 จากการวิเคราะห์เพื่อสร้างแผนที่โดยระบบ เชื่อมทิศ และการตรวจจับวัตถุที่กำลังเคลื่อนไหวโดยกล้องจะช่วยให้หุ่นยนต์กู้ภัย (Rescue Robot) ทำงานได้อย่างมีประสิทธิภาพมากขึ้น

1.4.4 ระบบการวัดอุณหภูมิสามารถวัดอุณหภูมิสิ่งแวดล้อมได้และนำไปใช้กับหุ่นยนต์กู้ภัย (Rescue Robot) ได้

1.4.5 โครงสร้างของหุ่นยนต์สามารถเคลื่อนที่เข้าไปในพื้นที่ที่เข้าถึงได้ยาก

จากบทนำที่ได้กล่าวมา ต่อไปจะได้พบกับส่วนต่างๆของงานวิจัย ซึ่งแบ่งออกเป็นบทต่างๆดังนี้

บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้องกับงานวิจัย

บทที่ 3 การออกแบบและวิธีการดำเนินงานวิจัย

บทที่ 4 ผลการทดลองงานวิจัย

บทที่ 5 บทสรุปและข้อเสนอแนะ

บทที่ 2

ทฤษฎีและเอกสารของงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ (Microcontroller) เป็นอุปกรณ์อิเล็กทรอนิกส์แบบหนึ่งที่รวมเอาหน่วยประมวลผล หน่วยคำนวณทางคณิตศาสตร์และลอจิก วงจรรับสัญญาณอินพุต วงจรจับสัญญาณเอาต์พุต หน่วยความจำ วงจรกำเนิดสัญญาณนาฬิกาไว้ด้วยกัน ทำให้สามารถนำไปใช้งานแทนวงจรรีเลย์อิเล็กทรอนิกส์ที่ซับซ้อนได้เป็นอย่างดี ช่วยลดจำนวนอุปกรณ์และขนาดของระบบ ในขณะที่มีความสามารถสูงขึ้น ภายใต้งบประมาณที่เหมาะสม

ไมโครคอนโทรลเลอร์มาจากคำ 2 คำรวมกันคือ “ไมโคร” (micro) ซึ่งหมายถึงไมโครโปรเซสเซอร์ (microprocessor) ซึ่งเป็นอุปกรณ์ประมวลผลข้อมูลขนาดเล็ก ภายในประกอบด้วย หน่วยประมวลผลกลางหรือซีพียู (CPU : Central Processing Unit) หน่วยคำนวณทางคณิตศาสตร์และลอจิก (ALU : Arithmetic Logic Unit) วงจรเชื่อมต่อหน่วยความจำ และวงจรมีสัญญาณนาฬิกา อีกคำหนึ่งคือคำว่า “คอนโทรลเลอร์” (controller) หมายถึง อุปกรณ์ควบคุม ดังนั้นไมโครคอนโทรลเลอร์จึงเป็นอุปกรณ์ที่ใช้ในการควบคุม โดยที่สามารถเขียนโปรแกรมเพื่อกำหนดรูปแบบการควบคุมได้อย่างอิสระ

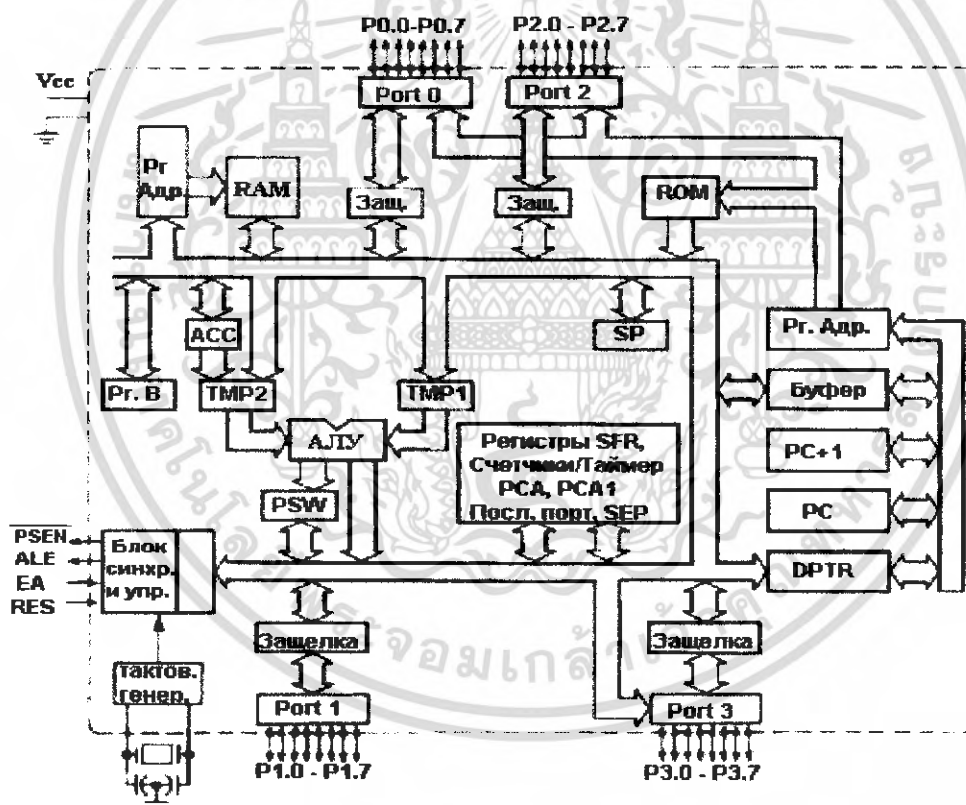
2.1.1 คุณสมบัติของไมโครคอนโทรลเลอร์ ตระกูล MCS-51 อนุกรม AT89Cxx

- เป็นไมโครคอนโทรลเลอร์ที่ใช้ซีพียูขนาด 8 บิต
- ภายในมีหน่วยความจำโปรแกรมเป็นแบบแฟลชสามารถลบและเขียนใหม่ได้พันครั้ง
- หน่วยความจำข้อมูลพื้นฐานเป็นหน่วยความจำแบบแรม ในบางเบอร์จะมีหน่วยความจำแบบEEPROM เพิ่มเติม
- ขาพอร์ตเป็นแบบสองทิศทาง สามารถใช้งานเป็นได้ทั้งอินพุตและเอาต์พุต
- มีวงจรมีสื่อสารอนุกรมแบบ Full Duplex
- Timer/Counter ขนาด 16 บิตอย่างน้อย 2 ตัว
- สามารถรองรับแหล่งกำเนิด interrupt ได้ 6 ประเภท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สามารถรองรับแหล่งกำเนิด interrupt ได้ 6 ประเภท
- สามารถขยายหน่วยความจำภายนอกเพิ่มเติมได้สูงสุด 64 Kbytes
- มีวงจรกำเนิดสัญญาณฟิสิกส์ภายในชิพ

ในรูปที่ 2-1 เป็นโครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ MCS-51 ในอนุกรม AT89Cxx จะเห็นได้ว่าโครงสร้างของ AT89Cxx จะเหมือนกับไมโครคอนโทรลเลอร์ตระกูล MCS-51 พื้นฐาน หากแต่แตกต่างกันเฉพาะหน่วยความจำโปรแกรมแบบแฟลชที่เพิ่มเติมเข้ามาหากเป็นไมโครคอนโทรลเลอร์ในอนุกรม 87xx หน่วยความจำโปรแกรมภายในจะเป็นแบบ EEPROM และบางเบอร์สามารถโปรแกรมได้เพียงครั้งเดียว



Архитектура микроконтроллера 8XC51

รูปที่ 2-1 รายละเอียดโครงสร้างหลักของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชของ Atmel[1]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | | | |
|-------------|----|----|------------|
| P1 0 | 1 | 40 | VCC |
| P1 1 | 2 | 39 | P0 0 (AD0) |
| P1 2 | 3 | 38 | P0 1 (AD1) |
| P1 3 | 4 | 37 | P0 2 (AD2) |
| P1 4 | 5 | 36 | P0 3 (AD3) |
| P1 5 | 6 | 35 | P0 4 (AD4) |
| P1 6 | 7 | 34 | P0 5 (AD5) |
| P1 7 | 8 | 33 | P0 6 (AD6) |
| RST | 9 | 32 | P0 7 (AD7) |
| (RXD) P3 0 | 10 | 31 | EA/VPP |
| (TXD) P3 1 | 11 | 30 | ALE/PROG |
| (INT0) P3 2 | 12 | 29 | PSFN |
| (INT1) P3 3 | 13 | 28 | P2 7 (A15) |
| (T0) P3 4 | 14 | 27 | P2 6 (A14) |
| (T1) P3 5 | 15 | 26 | P2 5 (A13) |
| (WR) P3 6 | 16 | 25 | P2 4 (A12) |
| (RD) P3 7 | 17 | 24 | P2 3 (A11) |
| XTAL2 | 18 | 23 | P2 2 (A10) |
| XTAL1 | 19 | 22 | P2 1 (A9) |
| GND | 20 | 21 | P2 0 (A8) |

| | | | |
|-------------|----|----|-------------|
| RST/VPP | 1 | 20 | VCC |
| (RXD) P3.0 | 2 | 19 | P1.7 |
| (TXD) P3.1 | 3 | 18 | P1.6 |
| XTAL2 | 4 | 17 | P1.5 |
| XTAL1 | 5 | 16 | P1.4 |
| (INT0) P3.2 | 6 | 15 | P1.3 |
| (INT1) P3.3 | 7 | 14 | P1.2 |
| (T0) P3.4 | 8 | 13 | P1.1 (AIN1) |
| (T1) P3.5 | 9 | 12 | P1.0 (AIN0) |
| GND | 10 | 11 | P3.7 |

AT89C51

AT89C2051

รูปที่ 2-2 การจัดขามาตรฐานของไมโครคอนโทรลเลอร์ MCS-51 ในอนุกรม AT89C5x[4]

2.1.2 การใช้งาน Timer/counter

การใช้งาน Timer/counter นั้นมีรีจิสเตอร์ที่ต้องใช้ 2 ตัวคือ TMOD และ TCON โดยที่ TMOD เป็นรีจิสเตอร์ขนาด 8 บิตไม่สามารถเข้าถึงระดับบิตได้ แบ่งการทำงานออกเป็น 2 ส่วนคือ 4 บิตต่างใช้เลือกโหมดการทำงาน Timer/Counter 0 และ 4 บิตบนใช้เลือกโหมด Timer/Counter 1 ในการทำงานของ Timer นั้นจะเริ่มการนับตั้งแต่ 0000H จนถึง FFFFH (ในกรณี 16 บิต) และเมื่อเปลี่ยนจาก FFFFH เป็น 0000H จะเกิดการโอเวอร์โฟลจขึ้น

| Gate | C/T | M1 | M0 | Gate | C/T | M1 | M0 |
|------|-----|----|----|------|-----|----|----|
|------|-----|----|----|------|-----|----|----|

รูปที่ 2-3 บิตต่างๆ ในรีจิสเตอร์ TMOD

โดยที่ Gate ใช้เลือกลักษณะการควบคุมการทำงานของ Timer/Counter

C/T ใช้เลือกการทำงานระหว่าง Timer/Counter

M1, M0 ใช้เลือกโหมดการทำงานของ Timer/Counter

“00” เลือกการทำงานโหมด Timer/Counter 13 บิต

“01” เลือกการทำงานโหมด Timer/Counter 16 บิต

“10” เลือกการทำงานโหมด Timer/Counter 8 บิตแบบตั้งค่าอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

“11” สำหรับ Timer0 เลือกให้ทำงานในโหมด Timer/Counter แยกส่วน โดยแยก ออกเป็น Timer/Counter 8 บิต 2 ตัว รีจิสเตอร์ TLO จะได้รับการควบคุมการเปิดปิดจากบิตTR0 ใน รีจิสเตอร์ TCON และรีจิสเตอร์ TH0 ซึ่งเป็น Timer/Counter 8 บิตอีกตัวหนึ่ง

บิต TR0 ในรีจิสเตอร์ TCON และรีจิสเตอร์ TH0 ซึ่งเป็น Timer/Counter 8 บิตอีกตัวหนึ่ง จะ ได้รับการควบคุมจากบิต TR1 ในรีจิสเตอร์ TCON ในกรณีของ Timer1 เป็นการสั่งให้ Timer/Counter1 หยุดทำงาน

ส่วนรีจิสเตอร์ TCON ใช้ควบคุมการทำงานของ Timer/Counter เป็นรีจิสเตอร์ขนาด 8 บิต สามารถเข้าถึงระดับบิตได้ มีโครงสร้างดังรูป

| | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|
| TF1 | TR1 | TF0 | TR0 | IE1 | IT1 | IE0 | IT0 |
|-----|-----|-----|-----|-----|-----|-----|-----|

รูปที่ 2-4 บิตต่างๆ ในรีจิสเตอร์ TCON

โดยที่ TF1/0 แสดงการโอเวอร์โฟลว์ของ Timer 1,0 จะเซตโดย hardware และเคลียร์โดย software
TR1/0 ควบคุมการเปิดปิด Timer 1,0 จะเซตและเคลียร์โดย software
IE1/0 แสดงการอินเตอร์รัพท์จาก INT1 และ INTO
IT1/0 บิตเลือกชนิดสัญญาณอินเตอร์รัพท์จากภายนอก INT1 และ INTO

การเริ่มและหยุด Timer นั้นสามารถควบคุมได้ที่บิต TRx ในรีจิสเตอร์ TCON โดยปกติแล้ว TRx จะเคลียร์หลังจากระบบลูกรีเซต ซึ่งเป็นการให้ Timer ไม่นับและ TRx นี้จะเซตให้เริ่มทำงานได้ ด้วย SETB TRx และสามารถหยุดการทำงานได้ด้วยคำสั่ง CLR TRx

การคำนวณ Timer นั้นสามารถคำนวณได้ดังนี้

2.1.3 ทฤษฎีการสื่อสารผ่านพอร์ตอนุกรม

เป็นการรับส่งข้อมูลที่ละบิต จนครบ 1 ไบต์ ถ้าต้องการส่งข้อมูล 1 ไบต์ คือ D₀ - D₇ อาจส่ง บิต D₀ ออกไปก่อนแล้วตามด้วย D₁ ไปเรื่อยๆ จนถึง D₇ การส่งข้อมูลทั้งสองแบบมีข้อดีข้อเสีย แตกต่างกันคือ การส่งข้อมูลแบบขนานสามารถส่งข้อมูลได้เร็ว คือ ส่งทีละบิตจะได้ข้อมูลครบ 1 ไบต์ แต่ถ้าต้องส่งเป็นระยะไกลๆ จะสิ้นเปลืองสายสัญญาณมาก ถ้าเป็นการส่งแบบอนุกรม เมื่อต้องการส่ง

ข้อมูลเป็นระยะๆ จะช่วยประหยัดสายสัญญาณเนื่องจากจะใช้สายอย่างน้อยเพียง 2 เส้น คือ สายสัญญาณกับสายกราวด์ แต่การรับส่งข้อมูลจะใช้เวลานานเนื่องจากการส่งทีละบิต ในบทนี้จะกล่าวถึงพื้นฐานการรับส่งข้อมูลแบบอนุกรม โดยเน้นที่ตัว MCS-51 เป็นสำคัญ

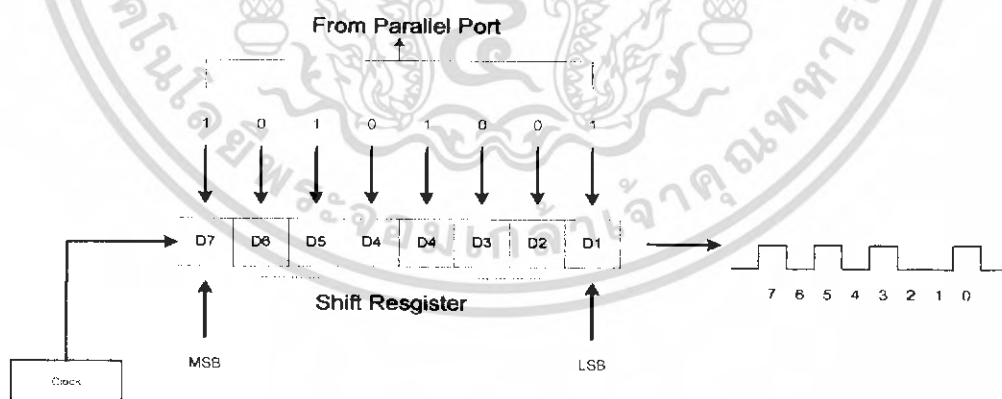
2.1.3.1 การสื่อสารข้อมูลแบบ Asynchronous

เป็นการรับและส่งข้อมูลโดยไม่จำเป็นต้องมีสัญญาณนาฬิกาพร้อมด้วยแต่จะใช้การกำหนดค่าอัตราเร็วในการรับส่งข้อมูลให้มีค่าเท่ากัน ซึ่งเรียกอัตราเร็วนี้ว่า Baud rate (bit/second)

รูปแบบของข้อมูลที่ใช้ในการรับส่งแบบ Asynchronous ประกอบด้วย 4 ส่วนด้วยกัน คือ

1. บิตเริ่มต้น (start bit) มีขนาด 1 บิต
2. บิตข้อมูลแบบอนุกรม มีขนาด 8 บิต
3. บิตตรวจสอบพาริตี (parity bit) มีขนาด 1 บิตหรือ ไม่มี
4. บิตปิดท้ายหรือบิตหยุด (stop bit) มีขนาด 1 บิต

เมื่อไมโครคอมพิวเตอร์ต้องการจะรับส่งข้อมูลแบบอนุกรม ตัวไมโครคอมพิวเตอร์จะส่งข้อมูลออกไปทางพอร์ทแบบขนานก่อน จากนั้นจะมีอุปกรณ์มาต่อที่พอร์ท เพื่อแปลงข้อมูลแบบขนานให้เป็นแบบอนุกรมอีกทีหนึ่ง (Parallel to Serial Conversion) ตัวแปลงข้อมูลนี้อาจจะพิจารณาได้ง่ายๆว่า เป็น Shift Register ดังรูป



รูปที่ 2-5 การส่งข้อมูลแบบอนุกรม

สำหรับตัวรับข้อมูลแบบอนุกรมนั้นเมื่อตัวรับข้อมูลทำงานจะเป็นการรับเข้ามาใน shift register แล้วส่งข้อมูลให้ไมโครคอมพิวเตอร์แบบขนานอีกทีหนึ่ง (Serial to Parallel) ระบบคอมพิวเตอร์ใน

ปัจจุบันจะมีตัวแปลง Parallel to Serial และ Serial to Parallel อยู่ในชิพไอซี เรียกว่า Universal Asynchronous Receiver Transmitter (UART) การส่งข้อมูลแบบอนุกรมนี้จะต้องมีการเพิ่มเติมข้อมูลบางอย่างเข้าไปเพื่อให้การรับส่งข้อมูลสามารถทำงานได้ถูกต้องมากขึ้น โดยมีการเติมค่าบิตต่างๆ ลงไปตามรูปที่ 2.4

| | | | | | | | | | | |
|-------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|--------|------|
| Start | D ₀ | D ₁ | D ₂ | D ₃ | D ₄ | D ₅ | D ₆ | D ₇ | Parity | Stop |
|-------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|--------|------|

รูปที่ 2-6 บิตต่างๆ ของการรับส่งข้อมูลแบบอนุกรม

ถ้ามีการส่งข้อมูลแบบ 8 บิต จะต้องส่งบิตแรกออกไปก่อน เรียกว่า บิตเริ่มต้น (Start Bit) ถ้ามีการส่งข้อมูลหลายๆ ไบต์ออกมา บิตนี้จะเป็นตัวบอกว่ามีข้อมูลใหม่มาแล้ว โดยทั่วไปบิตเริ่มต้นมักมีระดับลอจิกเป็น “0” ต่อจากบิตเริ่มต้นจะเป็นข้อมูลบิต D₀ ถึง D₇ จากนั้นจะตามด้วยบิตตรวจสอบ และบิตหยุด เพื่อบอกการสิ้นสุดของข้อมูล บิตหยุดอาจมีจำนวนมากกว่า 1 บิตก็ได้ เช่น ½ บิต, 2 บิต

การสื่อสารแบบอนุกรมนี้ การกำหนดอัตราเร็วในการรับส่งข้อมูลจะบอกเป็นบิตต่อวินาที (bit per second : bps) ที่เรียกว่าอัตราบิตหรือบิตเรต (baud rate) โดยค่ามาตรฐานที่ใช้กันมีหลายค่า ได้แก่ 110, 150, 300, 600, 1200, 2400, 4800, 9600 และ 19200 บิตต่อวินาที สำหรับความเร็วของการส่งข้อมูล (baud rate) สามารถหาได้จากตารางอัตรานาฬิกาที่ใช้กับ MCS-51

2.1.4 การรับส่งข้อมูลแบบอนุกรม

การรับส่งข้อมูลแบบอนุกรมกับไมโครคอนโทรลเลอร์ MCS-51 นั้นภายในชิพจะมี UART อยู่ในตัว ซึ่งเป็นชื่อของไมโครคอนโทรลเลอร์ พอร์ตอนุกรมของ MCS-51 จะใช้ขา TXD และ RXD ในการรับส่งข้อมูล โดยขาทั้ง 2 จะอยู่ในพอร์ท 3 คือ P3.1 หรือขา 11 เป็น TXD และ P3.0 หรือขา 10 เป็น RXD พอร์ตอนุกรมของ MCS-51 สามารถทำงานแบบ full duplex ได้ คือสามารถรับส่งข้อมูลได้ในเวลาเดียวกัน โดยในการรับส่งข้อมูลจะมีบัฟเฟอร์สำหรับเก็บข้อมูลให้ใช้

Register ที่สำคัญในการรับส่งข้อมูล คือ SBUF และ SCON ซึ่งเป็นรีจิสเตอร์ที่อยู่ใน Special Function Register โดยที่ถ้าเขียนข้อมูลลงไปทีรีจิสเตอร์ Serial Port Buffer (SBUF) จะเป็นการส่งข้อมูลออกทางพอร์ตอนุกรม และถ้าอ่านข้อมูลจาก SBUF นี้จะเป็นการรับข้อมูลจากพอร์ตอนุกรม

สำหรับ Serial Port Control Register (SCON) จะเป็นรีจิสเตอร์ที่สามารถเข้าถึงข้อมูลระดับบิตได้ รีจิสเตอร์นี้จะทำหน้าที่ควบคุมและบอกสถานะต่างๆของการรับส่งข้อมูลแบบอนุกรม บิต TI เมื่อเป็น “1” แสดงว่าข้อมูลได้ถูกส่งออกพอร์ทไปแล้ว, บิต RI เมื่อเป็น “1” แสดงว่าได้รับข้อมูลเข้ามาแล้ว, บิต REN เป็นตัวใช้กำหนดให้มีการรับข้อมูล, บิต SM0 SM1 และ SM2 เป็นบิตที่ใช้กำหนดโหมดการทำงาน, บิต TB8 และ RB8 เป็นบิตตรวจสอบที่ส่งออกและรับเข้าเมื่อใช้งานใน โหมด 2 และ 3

| | | | | | | | |
|-----|-----|-----|-----|-----|-----|----|----|
| SM0 | SM1 | SM2 | REN | TB8 | RB8 | TI | RI |
|-----|-----|-----|-----|-----|-----|----|----|

รูปที่ 2-7 บิตต่างๆ ของรีจิสเตอร์ SCON

ในการจะรับส่งข้อมูลผ่านพอร์ทอนุกรมต้องทำการป้อนค่าให้กับรีจิสเตอร์ SCON ก่อน เพื่อกำหนดโหมดการทำงานและลักษณะต่างๆของการรับและส่งข้อมูล

2.1.5 อัตราการส่งข้อมูลของพอร์ทอนุกรม

ในโหมด 0 และ 2 ไม่สามารถกำหนดอัตราการรับส่งข้อมูลได้ โดยใน โหมด 0 จะมีอัตรา baud rate= ความถี่ oscillatorหารด้วย 12 ในโหมด 1 จะมี 2 ค่าคือ ความถี่ oscillatorหารด้วย 32 ที่ SMOD="0" และหารด้วย 64 ที่ SMOD="1" ซึ่งสามารถกำหนดได้ในรีจิสเตอร์ PCON บิตที่ 7

การคำนวณหาค่าอัตราบอดเรตที่กำหนดด้วย Timer 1 สามารถหาได้ด้วยสมการนี้

$$Baud\ Rate = 2^{SMOD} / 32 \times \text{ความถี่ oscillator} / (12 \times [256 - TH1])$$

โดยที่ SMOD เป็นค่าของบิตภายในรีจิสเตอร์ PCON ซึ่งอาจมีค่าเป็น 0 หรือ 1

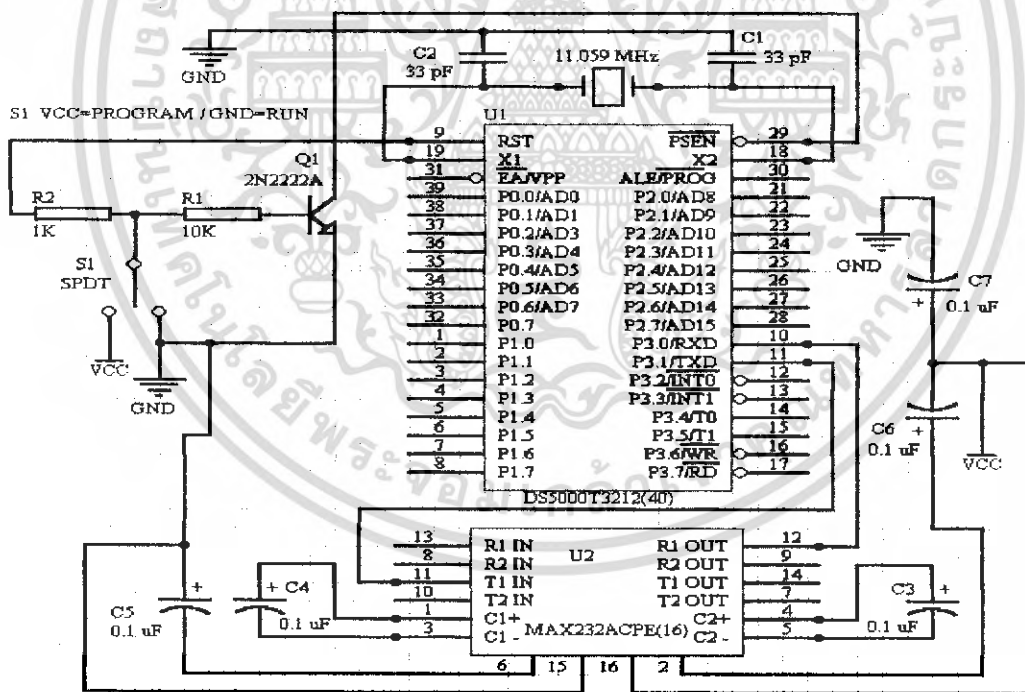
TH1 ค่าภายในรีจิสเตอร์ TH1 ใช้สำหรับ Reload ค่าของการนับเวลา

รูปแบบทั่วไปของการหาค่า Baud Rate ใน โหมด 1 และ 3 สามารถหาได้ดังนี้

$$\text{Baud Rate} = \text{Timer1 Overflow Rate} / 32$$

| ค่า Baud Rate | Crystal | SMOD โหมด | ค่าใน TH1 | ค่า Baud Rate ที่ได้ | Error |
|---------------|---------|-----------|-----------|----------------------|-------|
| 9,600 | 12.00 | 1 | -7(F9H) | 8,923 | 7% |
| 9,600 | 11.059 | 0 | -3(FDH) | 9,600 | 0 |
| 2,400 | 11.059 | 0 | -12(F4H) | 2,400 | 0 |
| 1,200 | 11.059 | 0 | -24(E8H) | 1,200 | 0 |

ตารางที่ 2-1 ตัวอย่างความถี่สัญญาณนาฬิกาที่ใช้กำหนด Baud Rate ค่าต่างๆ

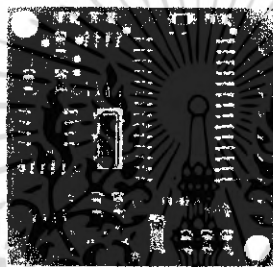


รูปที่ 2-8 การเชื่อมต่อ MCS-51 กับ IC MAX232A เพื่อการรับส่งข้อมูลแบบอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 การใช้ เซมิทิสแบบดิจิทัล CMPS03

ในการทำโครงงานนี้ได้เลือกใช้ เซมิทิสแบบดิจิทัล CMPS03 เป็นตัวสร้างแผนที่ให้กับหุ่นยนต์และหัวใจสำคัญของ CMPS03 คือการตรวจจับสนามแม่เหล็กเบอร์ KMZ 51 Philips จำนวน 2 ตัวเพื่อให้มีความไวเพียงพอในการตรวจจับสนามแม่เหล็กโลกและไมโครคอนโทรลเลอร์เพื่อรับสัญญาณจากตัวตรวจจับมาประมวลผลเป็นข้อมูลดิจิทัล และสัญญาณพัลส์สำหรับแสดงผลทิศทางรายละเอียดดูที่ Datasheet



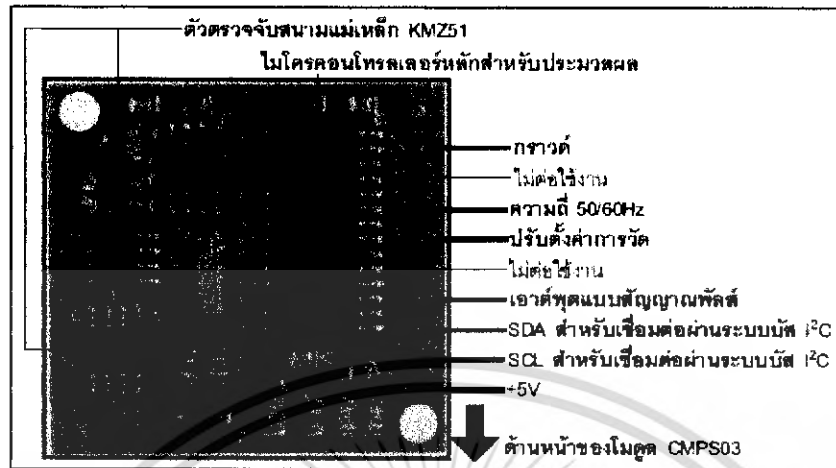
รูปที่ 2-9 ลักษณะภายนอกของ เซมิทิสแบบดิจิทัล CMPS03 [2]

2.2.1 คุณสมบัติของเซมิทิสดิจิทัล CMPS03

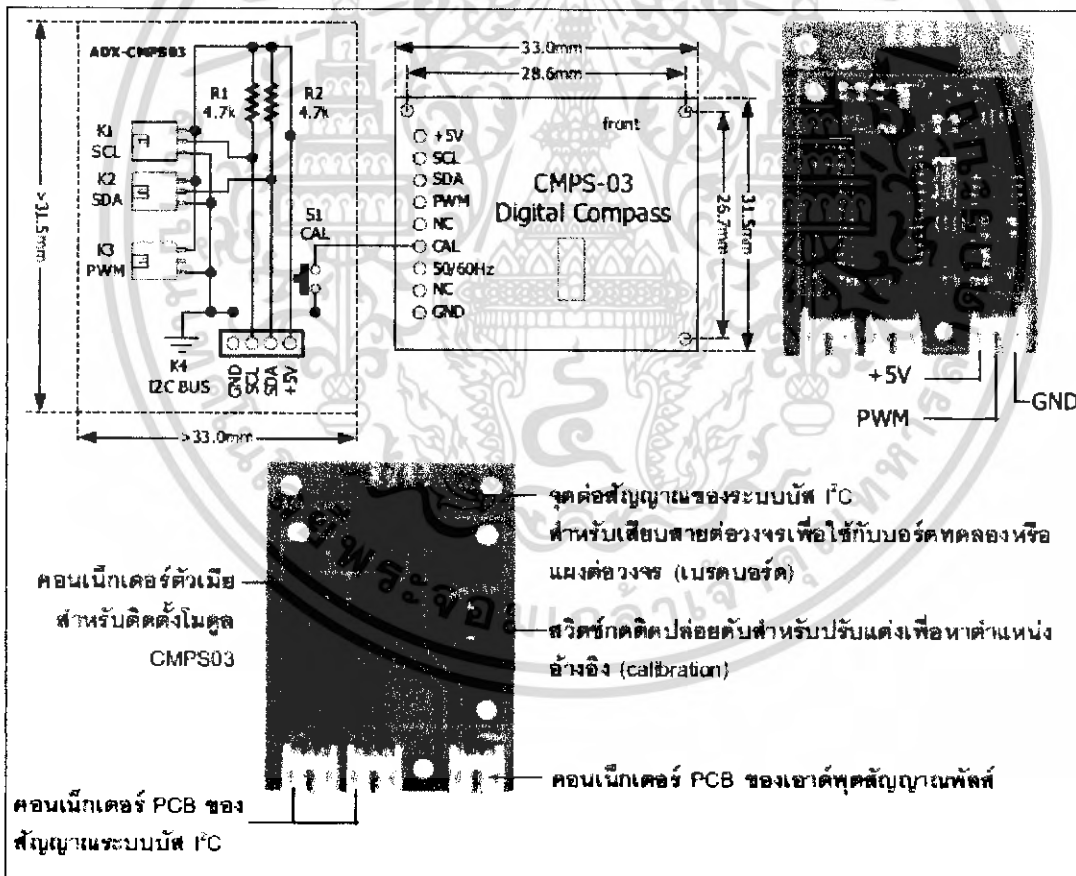
คุณสมบัติ

- ใช้ไฟเลี้ยง +5V ต้องการกระแสไฟฟ้า 20mA
- ใช้ตัวตรวจจับสนามแม่เหล็กเบอร์ KMZ51 ของ Philips จำนวน 2 ตัว เพื่อให้สามารถตรวจจับสนามแม่เหล็กโลกได้อย่างสมบูรณ์และมีความละเอียดมากเพียงพอ
- ความละเอียดของมุม 0.1 องศา
- ค่าความผิดพลาด 3-4 องศา โดยประมาณ หลังจากการปรับตั้ง
- เอาต์พุตแบบสัญญาณพัลส์ ความกว้าง 1 ถึง 37 มิลลิวินาที โดยมีอัตราเพิ่มครั้งละ 0.1 มิลลิวินาที
- เอาต์พุตข้อมูลดิจิทัลผ่านการติดต่อระบบบัส I²C รองรับสัญญาณนาฬิกาความถี่สูงถึง 1MHz โดยให้ข้อมูล 2 รูปแบบคือ 0-255 และ 0-3599
- ขนาดเล็กเพียง 32 x 35 มิลลิเมตร
- สื่อสารกับไมโครคอนโทรลเลอร์ยี่ห้อใดก็ได้ทุกตระกูล อาทิ เมลิกแอสแตมพ์ 2SX/2P, PIC, MCS-51, PSOC, 68HC11 ทั้งผ่านระบบบัส I²C และด้วยการวัดสัญญาณพัลส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-10 ขาที่ต่อใช้งาน เข็มทิศแบบดิจิตอล CMPS03 [2]



รูปที่ 2-11 การต่อเข้ากับอุปกรณ์เข็มทิศแบบดิจิตอล CMPS03 [2]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 พื้นฐานการประมวลผลภาพ

การประมวลผลภาพดิจิทัล(Image Processing) เป็นกระบวนการที่นำข้อมูลภาพดิจิทัลเข้ามาทำการประมวลผล และให้ข้อมูลออกมาเป็นข้อมูลภาพตัวใหม่ ซึ่งการประมวลผลส่วนใหญ่จะเน้นในด้านการปรับปรุงภาพให้ดีขึ้น (Image Enhancement) การวิเคราะห์ภาพ (Image Analysis) การคืนสภาพเดิมของรูปภาพ (Image Restoration) การแบ่งภาพออกเป็นส่วนๆ (Image Segmentation) การแปลงภาพ (Image Transforms) และการบีบอัดรูปภาพ (Image Compression) ในหัวข้อนี้เรามีวัตถุประสงค์คือ เพื่อสร้างพื้นฐานความเข้าใจเกี่ยวกับการประมวลผลภาพดิจิทัล ดังนั้นเนื้อหาส่วนใหญ่จึงเป็นเพียงพื้นฐานเบื้องต้น

2.3.1 ประเภทการกระทำภาพ(Types of Image Operation)

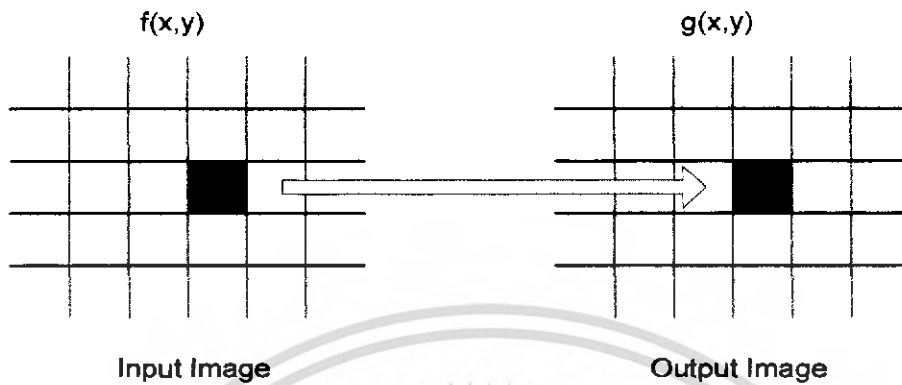
การประมวลผลภาพดิจิทัลจะเป็น กระบวนการที่ทำการ(Operation) ภาพอย่างใดอย่างหนึ่งต่อภาพนำเข้า (Input Image) เพื่อให้ได้ภาพผลลัพธ์ (Output Image) มีลักษณะของภาพเป็นไปตามที่ ต้องการ ซึ่งการกระทำภาพที่ใช้ในการประมวลผลภาพดิจิทัลมีอยู่มากมายหลายแบบ ความเข้าใจเกี่ยวกับคุณลักษณะและการแยกแยะประเภทของการกระทำภาพ จะช่วยให้เราสามารถคาดคะเนภาพผลลัพธ์ที่จะได้จากกรกระทำแต่ละแบบ หรือการประมาณความซับซ้อนของการกระทำภาพที่จะใช้

การกระทำภาพในการประมวลผลภาพดิจิทัลสามารถแบ่งออกได้เป็นประเภทใหญ่ๆ 3 ประเภท คือ

1. การกระทำจุดต่อจุด(Point Operation) การกระทำแบบนี้ ค่าความเข้มแสงในแต่ละพิกเซลของภาพผลลัพธ์ จะขึ้นกับค่าความเข้มแสงของพิกเซลในภาพนำเข้า ณ ตำแหน่งที่สมนัยกันดังรูปที่ 1 ลักษณะการกระทำภาพประเภทนี้ได้แก่ การปรับความสว่าง หรือ ความคมชัดของภาพดิจิทัล การ บวก ลบ คูณ และหาร ภาพดิจิทัล หรือ การกระทำทางตรรกศาสตร์ต่างๆ เป็นต้น

ถ้า $f(x,y)$ และ $g(x,y)$ เป็นภาพนำเข้าและภาพผลลัพธ์ตามลำดับ ค่าของพิกเซล $g(x,y)$ จะมีค่าดังนี้

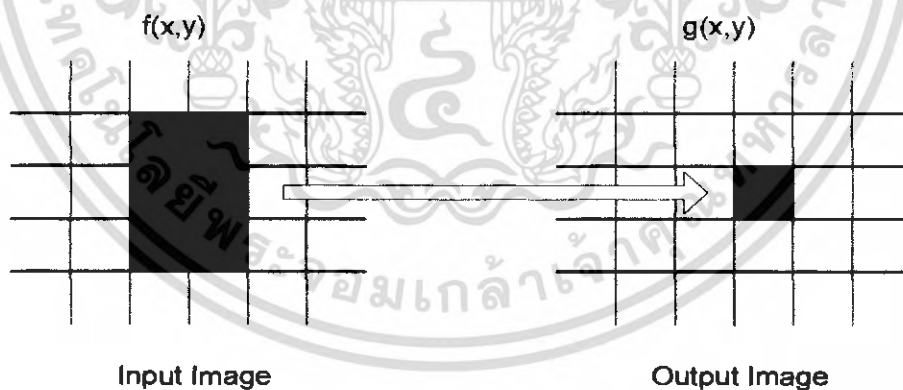
$$g(x_i,y_i) = \tau [f(x_i, y_i)] ; \text{เมื่อ } \tau \text{ เป็นการกระทำภาพใดๆ}$$



2. การทำการเฉพาะบริเวณ (Local Operation) สำหรับการกระทำแบบนี้ค่าความเข้มแสงของพิกเซลแต่ละจุดในภาพผลลัพธ์จะขึ้นกับค่าความเข้มแสงของกลุ่มพิกเซลที่อยู่บริเวณเดียวกัน (Neighborhood Pixel) ในภาพนำเข้า ดังรูปที่ 2 ลักษณะการกระทำภาพประเภทนี้ได้แก่การหาขอบ (Edge Detection) การกรองสัญญาณในโดเมนระยะทาง (Spatial Filtering) เป็นต้น

ถ้า $f(x,y)$ และ $g(x,y)$ เป็นภาพนำเข้าและภาพผลลัพธ์ตามลำดับ ค่าของพิกเซล $g(x,y)$ จะมีค่าดังนี้

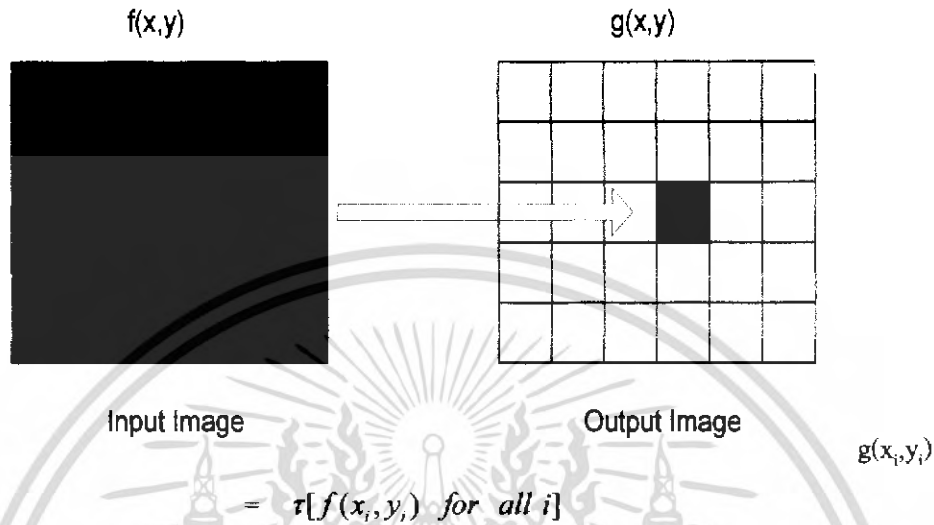
$$g(x,y) = \tau[\text{neighborhood of } f(x,y)]$$



3. การกระทำทั้งหมด (Global Operation) การกระทำแบบนี้ ค่าความเข้มแสงในแต่ละพิกเซลของภาพผลลัพธ์ (Output Image) จะขึ้นกับค่าความเข้มแสงของพิกเซลทุกตัวในภาพนำเข้า ดังรูปที่ 3 ลักษณะการกระทำภาพประเภทนี้ได้แก่ การเทรชโฮลดิ้ง (Thresholding) การทำฮิสโตแกรม (Histogram) เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้า $f(x,y)$ และ $g(x,y)$ เป็นภาพนำเข้าและภาพผลลัพธ์ตามลำดับ ค่าของพิกเซล $g(x,y)$ จะมีค่าดังนี้



2.3.2 พื้นฐานความสัมพันธ์ระหว่างพิกเซล(Basic Relationships between Pixels)

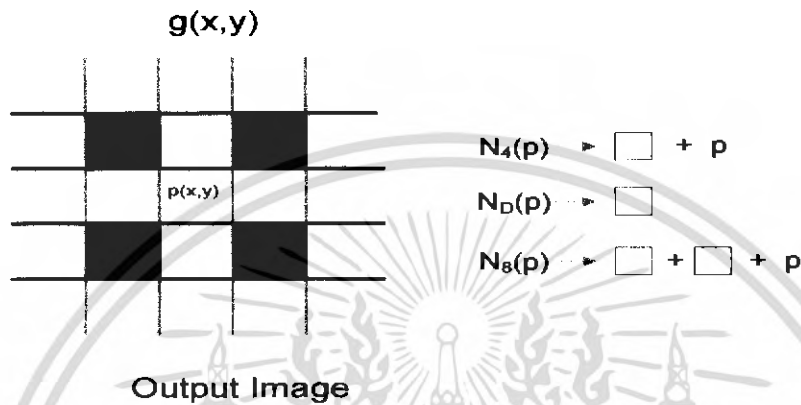
รูปภาพดิจิทัลจะมีคุณสมบัติทั้งในด้านการวัดระยะทางและรูปร่างเชิงเรขาคณิตที่แตกต่างเล็กน้อยสำหรับฟังก์ชันทางระนาบสองมิติแบบต่อเนื่องที่ประกอบด้วยพิกเซลเรียงต่อกันเป็นระนาบสองมิติ การวัดระยะทางความห่างหรือรูปร่างเชิงเรขาคณิตของวัตถุต่างๆ ในรูปภาพ จะกำหนดด้วยระยะห่างระหว่างพิกเซลและรูปร่างการเชื่อมต่อของพิกเซล ดังนั้นความเข้าใจพื้นฐานในด้านความสัมพันธ์ระหว่างพิกเซล ดังนั้นความเข้าใจพื้นฐานในด้านความสัมพันธ์ระหว่างพิกเซลซึ่งเป็นส่วนประกอบของรูปภาพดิจิทัล จึงมีความจำเป็นที่ต้องกล่าวถึง

2.3.2.1 พิกเซลใกล้เคียง (Pixels Neighbors)

ในการประมวลผลภาพดิจิทัลนั้น มีการกระทำหลายแบบที่ใช้กลุ่มของพิกเซลใกล้เคียงกันเป็นหลัก นอกจากนี้ การเชื่อม (Connectivity) ระหว่างพิกเซลจะทำให้เฉพาะในส่วนของพิกเซลที่อยู่ใกล้เคียงกันเท่านั้น ดังนั้นการเข้าใจคำนิยามของพิกเซลใกล้เคียงสามารถกำหนดได้เป็นแบบใหญ่ๆ 2 แบบ คือ พิกเซลใกล้เคียงแบบ 4 ซึ่งพิกเซลประชิดจะเรียงตัวกันเป็นรูปกากบาทและพิกเซลใกล้เคียงแบบ 8 ซึ่งพิกเซลประชิดจะเรียงกันอยู่เป็นรูปสี่เหลี่ยมจัตุรัส ดังรูปที่ 3

ถ้ากำหนดให้พิกเซล p มีตำแหน่งอยู่ที่ (x,y) $N_4(p)$ และ $N_8(p)$ จะเป็นการประชิดพิกเซลแบบ 4 และ 8 ตามลำดับ และ $N_4(p)$ เป็นพิกเซลที่อยู่ในตำแหน่ง $(x+1,y+1), (x+1,y-1), (x-1,y+1)$ และ $(x-1,y-1)$ ในขณะที่สมาชิกของ $N_8(p)$ จะประกอบไปด้วย p และ พิกเซลที่ตำแหน่ง

$(x+1,y), (x-1,y), (x,y+1)$ และ $(x,y-1)$ และสมาชิกของ $N_8(p)$ จะประกอบด้วย $p, N_4(p)$ และ $N_D(p)$



2.3.3 สัญญาณรบกวนในรูปภาพดิจิทัล (Noise in Digital Image)

รูปภาพดิจิทัลที่ได้ในงานจริงๆ มักจะมีคุณภาพลดลงเนื่องจากมีสัญญาณรบกวน (Noise) ปนอยู่ด้วยเสมอ การจำลองลักษณะของสัญญาณรบกวนสำหรับงานทางด้านการประมวลผลภาพดิจิทัลโดยมากจะอยู่ในรูปของฟังก์ชันการกระจาย (Distribution Functions) ทางสถิติต่างๆ ซึ่งในทางปฏิบัติสัญญาณรบกวนส่วนใหญ่จะเกิดขึ้นเนื่องจากปรากฏการณ์ธรรมชาติต่างๆ ในขั้นตอนการเปลี่ยนจากสัญญาณแสงมาเป็นสัญญาณไฟฟ้า ดังนั้นสมมติฐานของสัญญาณรบกวนที่นิยมใช้กันจะเป็นสัญญาณรบกวนแบบเกาส์เซียน (Gaussian) แบบเสมอ (Uniform) และแบบจุด (Impulse)

สัญญาณรบกวนแบบเกาส์เซียน (Gaussian Noise) จะเป็นสัญญาณรบกวนที่มีการกระจายในลักษณะปกติ (Normal) ซึ่งลักษณะการกระจายของสัญญาณรบกวนจะเป็นฟังก์ชันของค่าระดับเทา ดังสมการ

$$p(g) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(g-m)^2}{2\sigma^2}}$$

เมื่อ g เป็นค่าระดับเทา

m เป็นค่าเฉลี่ย

σ^2 เป็นค่าความแปรปรวน

สัญญาณรบกวนแบบเสมอ (Uniform Noise) จะเป็นสัญญาณรบกวนที่มีการกระจายในลักษณะสม่ำเสมอ ซึ่งจะเป็นไปตามสมการ

$$p(g) \begin{cases} 1 & a \leq g \leq b \\ b-a & \\ 0 & elsewhere \end{cases}$$

และสัญญาณรบกวนแบบจุด (Impulse Noise) จะเป็นสัญญาณรบกวนที่พิกเซลเป็นจุดๆ ไป โดยจะเปลี่ยนค่าความเข้มแสงของพิกเซล ณ ตำแหน่งหนึ่งๆ ให้มีค่าแตกต่างไปจากพิกเซลข้างเคียง ซึ่งค่าความเข้มแสงที่จำลองกันโดยมาก จะเป็นในลักษณะความเข้มแสงสีขาวและดำ (Salt and Pepper) นั่นคือรูปภาพดิจิทัลจะถูกทำลายโดยพิกเซลบางตำแหน่งจะถูกเปลี่ยนเป็นจุดดำ หรือ จุดขาว โดยพิกเซลรบกวน (Noise Pixels) สัญญาณรบกวนแบบนี้จะมีฟังก์ชันการกระจายตามสมการ

$$p(g) \begin{cases} A & g = a \\ B & g = b \end{cases}$$

เมื่อ a และ b เป็นค่าระดับเทาเดิม
A และ B เป็นค่าระดับเทาใหม่

2.3.4 การทำคอนโวลูชัน (Convolution)

การทำคอนโวลูชันในเชิงของการประมวลผลภาพดิจิทัล จะเป็นการกระทำระหว่างภาพนำเข้า $f(x,y)$ ที่มีขนาด $N*N$ เมื่อ N เป็นเลขจำนวนเต็มใดๆกับ มาส์ค (Mask) $m(x,y)$ ซึ่งจะเป็นภาพที่มีขนาด $M*M$ เมื่อ M เป็นเลขจำนวนเต็มใดๆ (ปกติจะเป็นเลขคี่) และมีขนาดน้อยกว่า N มากๆ เช่น $3*3$, $5*5$, $7*7$, $9*9$ หรือ $11*11$ เป็นต้น ผลลัพธ์ความเข้มแสงใหม่ที่ได้จากการทำคอนโวลูชันจะถูกเก็บไว้ในภาพผลลัพธ์ $g(x,y)$ ซึ่งขั้นตอนการทำคอนโวลูชันจะเป็นไปตามสมการ

$$g(x,y) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \sum_{k=0}^{M-1} \sum_{l=0}^{M-1} f(i,j)m(x-i,y-l)$$

62329

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสมการจะเห็นว่าการทำงานวอล์กชันจะมีวิธีการเป็นขั้นตอนดังนี้

1. กลับ (Flip) มาส์กเทียบกับตำแหน่งจุดตรงกลางของมาส์ก เมื่อใช้มาส์กมีขนาดเป็นเลขจำนวนเต็มคี่ ซึ่งโดยปกติมาส์กที่ใช้ในการประมวลผลภาพดิจิทัลจะมีลักษณะสมมาตรทั้งในแนว x และ y ทำให้มาส์กที่ได้จะมีลักษณะเหมือนเดิม ดังนั้นขั้นตอนนี้จึงมักจะถูกละไว้และการกำหนดมาส์กจะกำหนดในลักษณะที่เป็นมาส์กหลังจากการกลับแล้ว
2. เลื่อนมาส์กให้ไปทับภาพนำเข้า โดยให้จุดตรงกลางของมาส์กตรงกับพิกเซล $f(x,y)$
3. ทำการคูณค่าความเข้มแสงระหว่างพิกเซลของมาส์กและพิกเซลของภาพนำเข้าที่ ณ ตำแหน่งเดียวกันแบบจุดต่อจุด
4. บวกผลคูณที่ได้ในขั้นตอนที่ 3 ทั้งหมด และนำผลลัพธ์ที่ได้ไปใส่เป็นค่าความเข้มสว่างของผลลัพธ์ ณ ตำแหน่งเดียวกับพิกเซลของภาพนำเข้า นั่นคือ $g(x,y)$
5. เลื่อนตำแหน่งไปที่พิกเซลถัดไปของภาพนำเข้าและทำซ้ำตั้งแต่ ขั้นตอนที่ 2 ถึง 4 จนกระทั่งครบพิกเซลทุกตัวในภาพนำเข้า

ตัวอย่างเช่น ถ้ามีมาส์กขนาด 3×3 ที่มีค่า ω ตำแหน่งต่างๆหลังจากทำการกลับแล้วดังนี้

$$\begin{bmatrix} A & B & C \\ D & E & F \\ G & H & I \end{bmatrix}$$

เมื่อนำมาส์กนี้ไปทับพิกเซล p ณ ตำแหน่ง (x,y) ของภาพนำเข้า $f(x,y)$ ผลของการทำคอวอล์กชัน จะได้ค่าความเข้มแสงของพิกเซล q ณ ตำแหน่งเดียวกันของภาพผลลัพธ์ $g(x,y)$ มีค่าดังนี้

$$q = g(x,y) = Af(x-1,y-1) + Bf(x-1,y) + Cf(x-1,y+1) + Df(x,y-1) + Ef(x,y) + Ff(x,y+1) + Gf(x+1,y-1) + Hf(x+1,y) + If(x+1,y+1)$$

จากนั้นมาส์กจะเลื่อนไปที่ตำแหน่งถัดไปทางขวามือ นั่นคือ $f(x,y+1)$ โดยจะเข้าไปเรื่อยๆจนครบทุกพิกเซลในภาพนำเข้า

จากขั้นตอนการคอนโวลูชันจะเห็นได้ว่า ถ้าเราต้องการให้มาสก์ทุกตัวทาบลงบนภาพนำเข้าพอดี เราจะต้องข้ามไม่ทำการคอนโวลูชันพิกเซลแถวแรกๆหรือแถวท้ายๆและหลักแรกๆหรือหลักท้ายๆซึ่งจำนวนที่หายไปจะขึ้นกับขนาดของมาสก์ ตัวอย่างเช่น มาสก์ขนาด 3*3 การทำคอนโวลูชันจะไม่ทำกับพิกเซลแถวแรกและแถวสุดท้ายและในหลักแรกและหลักสุดท้าย การคอนโวลูชันเป็นพื้นฐานของการกระทำทางการประมวลผลดิจิทัลหลายประเภทใน โดเมนเวลา โดยเฉพาะการกรองสัญญาณภาพ (Image Filtering) ซึ่งจะกล่าวถึงในหัวข้อถัดไป

2.3.5 การกรองสัญญาณภาพในโดเมนเวลา (Image Filtering in Time Domain)

การกรองสัญญาณภาพ (Image Filtering) ในโดเมนเวลาเป็นเทคนิคพื้นฐานที่ใช้งานกันอย่างแพร่หลายสำหรับงานด้านการประมวลผลสัญญาณภาพ การกรองสัญญาณภาพเป็นเทคนิคแบบหนึ่งของการปรับปรุงภาพ (Image Enhancement) ที่มีจุดมุ่งหมายในการกำจัดสัญญาณความถี่ต่ำ ความถี่สูงหรือความถี่ของสัญญาณในช่วงหนึ่งๆการกรองสัญญาณภาพทำได้ทั้งในโดเมนเวลาหรือระยะทางและในโดเมนความถี่ แต่การกรองสัญญาณภาพทำได้ทั้งในโดเมนเวลาจะทำได้ง่ายและรวดเร็วกว่าการกรองสัญญาณในโดเมนความถี่ หลักการของการกรองสัญญาณภาพในโดเมนเวลาเป็นการกระทำแบบเฉพาะบริเวณแบบหนึ่งที่มีพื้นฐานการทำงานเป็นกระบวนการคอนโวลูชันภาพต้นฉบับกับมาสก์แบบต่างๆโดยที่ลักษณะของมาสก์ที่ใช้จะเป็นตัวกำหนดผลลัพธ์ที่จะได้ ซึ่งการกรองสัญญาณภาพในโดเมนเวลาจะทำได้ง่ายและรวดเร็วกว่าการกรองสัญญาณ โดเมนเวลาเป็นการกระทำแบบเฉพาะบริเวณแบบหนึ่งที่มีพื้นฐานการทำงานเป็นกระบวนการคอนโวลูชันภาพต้นฉบับกับมาสก์แบบต่างๆโดยที่ลักษณะของมาสก์ที่ใช้จะเป็นตัวกำหนดผลลัพธ์ที่จะได้ ซึ่งการกรองสัญญาณภาพในโดเมนเวลาสามารถแบ่งออกได้เป็นประเภทใหญ่ 3 ประเภท คือ การทำภาพให้เรียบ (Image Smoothing) การทำภาพให้คมชัด (Image Sharpening) และตัวหาขอบ (Edge Detector)

การทำภาพให้เรียบมีวัตถุประสงค์ในการกำจัดสัญญาณรบกวนซึ่งเป็นสัญญาณประเภทความถี่สูง โดยพื้นฐานอาศัยหลักการทำการเฉลี่ยค่าความเข้มแสงเฉพาะบริเวณหรืออีกนัยหนึ่งคือเป็นการกรองสัญญาณความถี่ต่ำ (Low-pass Filtering) ผลกระทบของการทำภาพให้เรียบก็คือภาพต้นฉบับจะพร่ามัว (Blurring Effect) มีความคมชัดน้อยลง เนื่องจากขอบของวัตถุในรูปภาพจะเป็นบริเวณที่มีการเปลี่ยนแปลงระดับค่าความเข้มแสงและจัดว่าเป็นสัญญาณความถี่สูงจะถูกกรองออกไป ดังนั้นเทคนิคการทำภาพให้เรียบส่วนใหญ่จะเน้นที่การกำจัดสัญญาณรบกวนแต่จะไม่ทำลายของวัตถุ

ในภาพมาสก์ที่ใช้ในการทำภาพให้เรียบจะมีลักษณะที่พิกเซลทุกตำแหน่งจะมีค่าเป็นบวกหมด ตัวอย่างเช่น

$$\text{มาสก์ ค่าเฉลี่ย} = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\text{มาสก์ แบบเกาส์เซียน} = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \text{ หรือ } \frac{1}{32} \begin{bmatrix} 1 & 4 & 1 \\ 4 & 12 & 4 \\ 1 & 4 & 1 \end{bmatrix} \text{ เป็นต้น}$$

โดยทั่วไปมาสก์แบบเกาส์เซียนจะเป็นที่นิยมใช้มากกว่าแบบค่าเฉลี่ย เนื่องจากว่าจะมีผลกระทบต่อภาพร่ามัว (Blur) ของภาพต้นฉบับน้อยกว่า และมีได้หลายแบบ โดยการกำหนดที่ค่าเบี่ยงเบนมาตรฐานของค่าพิกเซลทั้งหมดในมาสก์ ซึ่งตัวกระทำเรียบแบบเกาส์เซียนสองมิติ $G(x,y)$ จะมีค่าตามสมการ

$$G(x,y) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

นอกจากนี้มาสก์แบบเกาส์เซียนจะเหมาะสำหรับการกำจัดสัญญาณรบกวนที่มีการกระจายแบบเกาส์เซียน ซึ่งเชื่อว่าเป็นสัญญาณที่เกิดกับปรากฏการณ์ตามธรรมชาติต่างๆ

การทำภาพเรียบหรือการกำจัดสัญญาณรบกวนยังสามารถใช้มาสก์ที่มีความซับซ้อนมากขึ้น หรือมาสก์แบบไม่เชิงเส้นอีกหลายประเภท เช่น ค่ามัธยฐาน (Median) ค่าโหมด (Mode) หรือการหาค่าเฉลี่ยแบบไม่เชิงเส้น เป็นต้น

การทำภาพให้ชัด และ ตัดขอบจะเป็นการกรองสัญญาณในโดเมนเวลา ที่ใช้มาสก์ที่เป็นประเภทการหาค่าอนุพันธ์ (Derivatives) ของพิกเซลเฉพาะบริเวณ โดยที่การทำให้ภาพชัดขึ้นจะเป็นการกรองสัญญาณในโดเมนเวลา ที่ทำให้บริเวณที่มีการเปลี่ยนแปลงค่าความเข้มแสงในรูปภาพเด่นชัดขึ้น นั่นคือทำงานในลักษณะเป็นการกรองสัญญาณความถี่สูง (High-pass Filtering) ส่วนตัดขอบจะเป็นตัวกระทำที่ต้องการหาค่าตำแหน่งขอบของวัตถุ ที่อยู่ในภาพดิจิทัล มาส์คของการทำภาพให้ชัดและตัดขอบ จะมีลักษณะคล้ายกันคือค่าของพิกเซลในมาสก์จะมีทั้งบวกและลบ แต่จะมีลักษณะการ

วางตัวของค่าที่เป็นบวกและลบต่างกันกล่าวคือ มาส์คการทำภาพให้ชัดจะมีค่าเป็นบวกบริเวณใกล้ๆกับจุดตรงกลางมาส์ค และจะมีค่าเป็นลบอยู่รอบๆดังตัวอย่างต่อไปนี้

$$\text{มาส์ค ค่าเฉลี่ย} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \text{ หรือ } \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

$$\text{มาส์ค แบบเกาส์เซียน} = \begin{bmatrix} -2 & 1 & -2 \\ 1 & 4 & 1 \\ -2 & 1 & -2 \end{bmatrix} \text{ หรือ } \begin{bmatrix} 1 & -2 & 1 \\ -1 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} \text{ เป็นต้น}$$

มาส์คของตัวหาขอบจะมีวัตถุประสงค์ในการหาว่ามีการเปลี่ยนค่าความสว่างแบบฉับพลันในบริเวณใด ซึ่งถ้าพบก็หมายความว่าบริเวณนั้นๆอาจจะเป็นขอบของวัตถุ

ดังนั้นตัวอย่างของมาส์คจะเป็นดังต่อไปนี้

$$\text{โซเบล(Sobel)} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \text{ หรือ } \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

$$\text{พีวีธ(Prewitt)} = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \text{ หรือ } \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 หลักการทำงานของดีซีมอเตอร์

ดีซีมอเตอร์เป็นทรานส์ดิวเซอร์แรงบิดซึ่งมีการออกแบบให้มีคุณลักษณะพิเศษคือแรงบิดของเพลาของดีซีมอเตอร์จะเป็นสัดส่วนโดยตรงกับกระแสอาร์มาเจอร์ แรงบิดของเพลาของดีซีมอเตอร์จะได้จากผลระหว่างสนามแม่เหล็กและขดลวดตัวนำ หลักการนี้แสดงได้ในรูปที่ 2.13 ในที่นี้กระแสที่ไหลในขดลวดตัวนำจะสร้างฟลักซ์ที่ประกอบด้วยเส้นแรงแม่เหล็ก ϕ และขดลวดตัวนำเหล่านั้นอยู่ห่างจากศูนย์กลางการหมุนเท่ากับ r ความสัมพันธ์ระหว่างแรงบิดของเพลาและกระแสเท่ากับ

$$T = K \phi I \quad (2-1)$$

เมื่อ T คือแรงบิดของเพลา มีหน่วยเป็นนิวตัน-เมตร

ϕ คือเส้นแรงแม่เหล็ก มีหน่วยเป็นเวเบอร์

I คือกระแสเป็นแอมแปร์

และ K คือตัวคงที่ ดังนั้นแรงบิดของเพลาจะเป็นสัดส่วนโดยตรงกับผลคูณของเส้นแรงแม่เหล็กและกระแสเมื่อขดลวดตัวนำเคลื่อนที่ในสนามแม่เหล็กก็จะทำให้เกิดโวลต์เตจตกคร่อมตัวมันเอง โวลต์เตจนี้จะเป็นสัดส่วนกับความเร็วของเพลาของมอเตอร์และต้านการไหลของกระแส ความสัมพันธ์ระหว่างโวลต์เตจย้อนกลับนี้และความเร็วของเพลาของมอเตอร์

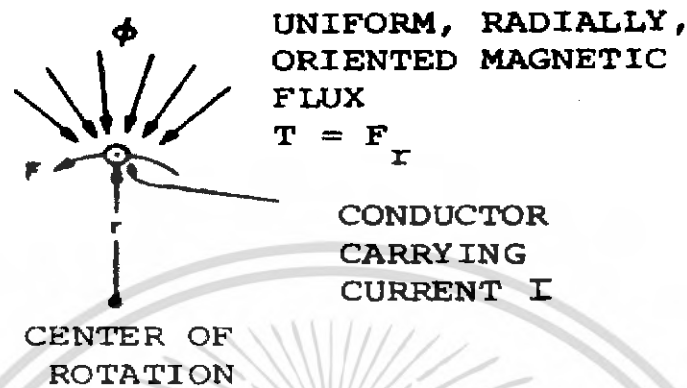
$$E = K \phi \omega \quad (2-2)$$

เมื่อ E คือโวลต์เตจย้อนกลับ emf มีหน่วยเป็นโวลต์

ϕ คือเส้นแรงแม่เหล็ก มีหน่วยเป็นเวเบอร์

ω คือความเร็วของมอเตอร์ มีหน่วยเป็นเรเดียน/วินาที

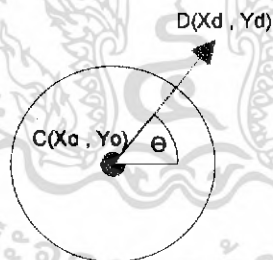
สมการ (2-1) และ (2-2) เป็นสมการที่แสดงถึงหลักการทำงานพื้นฐานของดีซีมอเตอร์



รูปที่ 2-12 การเกิดแรงบิดในตัวค้ำขิมอเตอร์

2.6 การเคลื่อนที่ของ Object

Object ที่นำมาแทนเป็นตัว Robot เพื่อจำลองการเคลื่อนที่ของ Robot ในโปรแกรมจะแทน Robot ด้วย Object วงกลมและเส้นตรงเพื่อใช้บอกทิศทางและแทนด้านหน้าของ Robot ดังรูป



รูปที่ 2-13 สัญลักษณ์ของ object ที่ใช้แทน Robot

จากรูป วงกลมจะใช้แทนตัว Robot และบ่งบอกด้านหน้าของ Robot ด้วยเส้นตรง CD ในการเคลื่อนที่ที่จะเคลื่อนที่ตามระยะทางที่กำหนดในทิศทางของเส้นตรง CD ในการควบคุมการเคลื่อนที่ จะต้องใช้ parameter 2 ค่าคือ ระยะทางในการเคลื่อนที่(R) และมุมของเส้นตรง CD เทียบกับแกน x (θ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

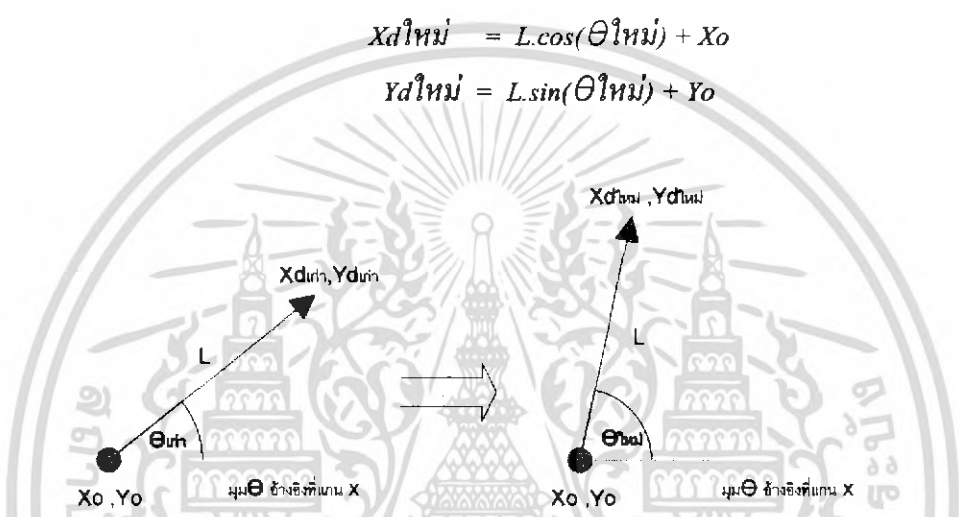
ที่ใช้ในการหาทิศทางในการเคลื่อนที่ ในการเคลื่อนที่นั้นจะพิจารณาเป็น 2 ส่วนคือ ส่วนของเส้นตรง และ ส่วนของวงกลม

การควบคุมการเคลื่อนที่ของเส้นตรงมี 2 ลักษณะคือ การเคลื่อนเส้นตรงไปในทิศทางของลูกศร และการหมุนปลายลูกศรเพื่อเปลี่ยนทิศทาง

- การหมุนปลายลูกศรทำได้โดยใช้สมการ

$$Xd_{ใหม่} = L \cdot \cos(\theta_{ใหม่}) + X_o$$

$$Yd_{ใหม่} = L \cdot \sin(\theta_{ใหม่}) + Y_o$$



รูปที่ 2- 14 การหมุนทิศทางของเส้นตรง

- โดยที่ L คือ ความยาวของเส้นตรง
- θ คือ มุมระหว่างเส้นตรงโดยวัดจากแกน x
- $Xd_{ใหม่}$ คือ ค่าพิกัด x ใหม่ ที่ได้เมื่อมีการหมุนลูกศร(หมุนตัว Robot)
- $Yd_{ใหม่}$ คือ ค่าพิกัด y ใหม่ ที่ได้เมื่อมีการหมุนลูกศร(หมุนตัว Robot)
- X_o คือ ค่าพิกัด x ที่จุดกำเนิดของเส้นตรง
- Y_o คือ ค่าพิกัด y ที่จุดกำเนิดของเส้นตรง

- การเคลื่อนเส้นตรงไปในทิศทางของลูกศรสามารถทำได้โดยใช้สมการ

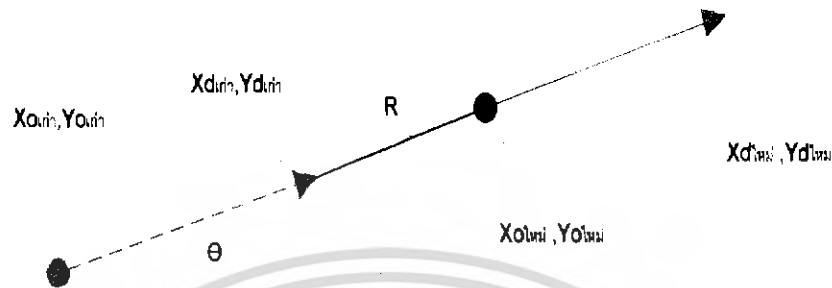
$$Xd_{ใหม่} = R \cdot \cos(\theta) + Xd_{เก่า}$$

$$Yd_{ใหม่} = R \cdot \sin(\theta) + Yd_{เก่า}$$

$$Xo_{ใหม่} = R \cdot \cos(\theta) + Xo_{เก่า}$$

$$Yo_{ใหม่} = R \cdot \sin(\theta) + Yo_{เก่า}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

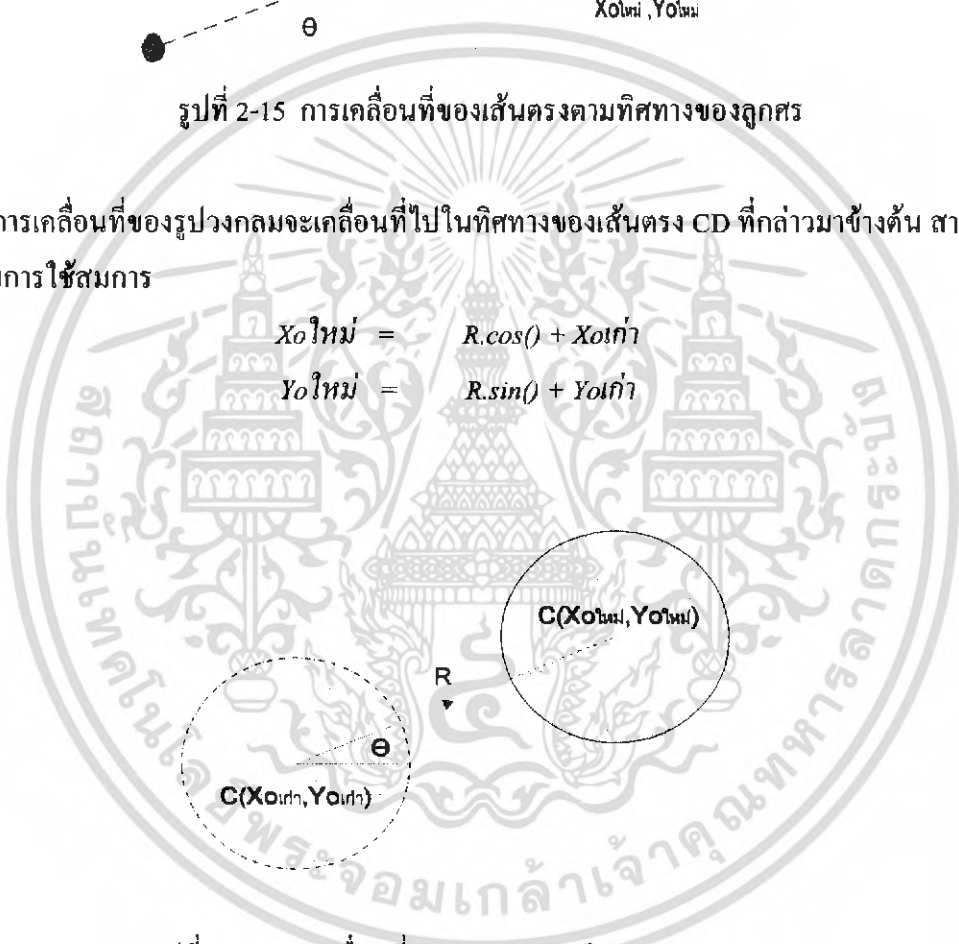


รูปที่ 2-15 การเคลื่อนที่ของเส้นตรงตามทิศทางของลูกศร

การเคลื่อนที่ของรูปวงกลมจะเคลื่อนที่ไปในทิศทางของเส้นตรง CD ที่กล่าวมาข้างต้น สามารถทำได้โดยการใช้สมการ

$$Xoใหม่ = R \cdot \cos(\theta) + Xoเก่า$$

$$Yoใหม่ = R \cdot \sin(\theta) + Yoเก่า$$



รูปที่ 2-16 การเคลื่อนที่ของวงกลมตามทิศทางของลูกศร

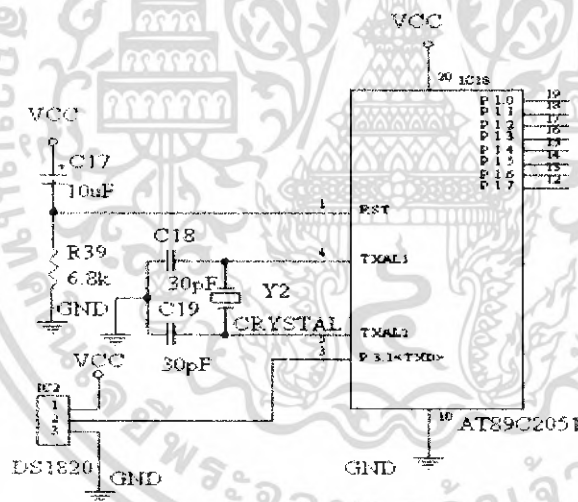
| | | | |
|--------|----------|-----|--|
| โดยที่ | R | คือ | ระยะทางในการเคลื่อนที่ |
| | θ | คือ | มุมระหว่างเส้นตรงโดยวัดจากแกน x |
| | $Xxใหม่$ | คือ | ค่าพิกัด x ใหม่ ที่ได้เมื่อมีการเคลื่อนที่เป็นระยะ R |
| | $Yxใหม่$ | คือ | ค่าพิกัด y ใหม่ ที่ได้เมื่อมีการเคลื่อนที่เป็นระยะ R |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Xxเก่า คือ ค่าพิกัด x เก่าก่อนที่จะมีการเคลื่อนที่
 Yxเก่า คือ ค่าพิกัด y เก่าก่อนที่จะมีการเคลื่อนที่

2.6 วงจรตรวจจับอุณหภูมิ

เราจะใช้ไอซีที่มีระบบการสื่อสารข้อมูลอนุกรมแบบหนึ่งสายซึ่งถือได้ว่าเป็นระบบที่มีความชาญฉลาดและใช้จำนวนสายสัญญาณเพียง 1 เส้นเท่านั้น โดยไม่ต้องมีสายสัญญาณนาฬิกาควบคุมจังหวะการถ่ายถอดข้อมูลเหมือนถ้ระบบสื่อสารข้อมูลอนุกรมแบบอื่นซึ่งก็คือ ไอซี DS1820 นั่นเองโดยเราสามารถควบคุมการรับส่งข้อมูลได้โดยใช้ไมโครคอนโทรลเลอร์เข้าควบคุมคือ ใช้ MCS 51 ในการควบคุมโดยผ่านการโปรแกรมการทำงานของ การสื่อสารข้อมูลเข้าไป ซึ่งเราจะใช้อุปกรณ์ภายนอกต่อร่วมอีกเพียงไม่กี่ตัวดังรูป



รูป 2-17 วงจรตรวจจับอุณหภูมิ

2.6.1 หลักการเบื้องต้นของไอซี DS1820

ไอซี DS1820 เป็นไอซีที่มีระบบการสื่อสารข้อมูลอนุกรมแบบหนึ่งสายซึ่งถือได้ว่าเป็นระบบที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีความชาญฉลาด และใช้จำนวนสายสัญญาณเพียง 1 เส้นเท่านั้น โดยไม่ต้องมีสายสัญญาณนาฬิกา มาควบคุมจังหวะการถ่ายถอดข้อมูลเหมือนกับระบบสื่อสารข้อมูลอนุกรมในแบบอื่น สายข้อมูลจะทำหน้าที่เสมือนเป็นสายสัญญาณนาฬิกาในตัว ส่วนค่าของข้อมูลจะพิจารณาจากลักษณะของรูปสัญญาณที่ปรากฏบนสายสัญญาณในแต่ละช่องของเวลาซึ่งเรียกว่า ไทม์สล็อต (Time Slot) โดยคาบเวลาต่ำสุดและสูงสุดของสถานะต่าง ๆ ในการสื่อสารข้อมูลในแต่ละไทม์สล็อตมีการกำหนดขอบเขตไว้อย่างชัดเจนการถ่ายถอดข้อมูลจะเกิดขึ้นในแต่ละไทม์สล็อตนั้น รูปแบบการถ่ายถอด ข้อมูลจะเป็นแบบอะซิงโครนัสในระดับบิต ไม่มีการกำหนดความยาวของข้อมูลเป็นระดับไบต์ระบบสื่อสารแบบนี้เหมาะที่จะใช้ในการสื่อสารข้อมูลระหว่างไอซีแผงวงจรเดียวกัน

2.6.2 การอินเตอร์เฟสผ่านสายเส้นเดียว

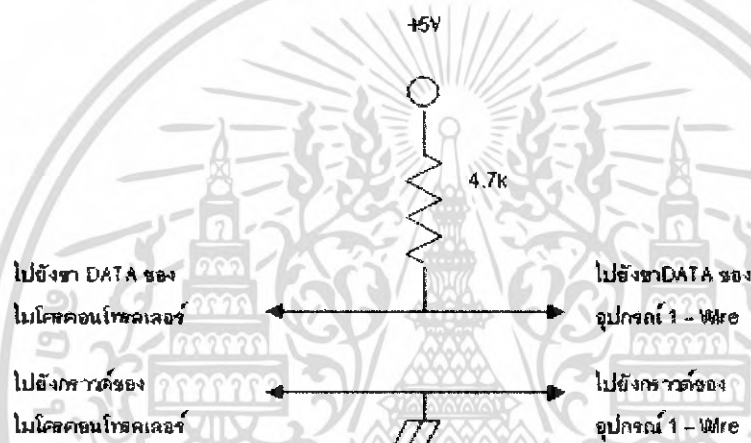
การเชื่อมต่อหรือการอินเตอร์เฟส (Interface) ระหว่าง ไมโครคอนโทรลเลอร์กับอุปกรณ์ภายนอก โดยใช้จำนวนสายสัญญาณให้น้อยที่สุดได้มีการพัฒนาอย่างต่อเนื่องจากหลายบริษัท ผู้ผลิต เช่น การเชื่อมต่ออุปกรณ์ต่อพ่วงแบบอนุกรม (Serial Peripheral Interface, SPI) ในไมโครคอนโทรลเลอร์ 68HC1 ของโมโตโรลา การเชื่อมต่อแบบ SPI นี้ช่วยให้ ไมโครคอนโทรลเลอร์แลกเปลี่ยนข้อมูลกับอุปกรณ์ต่อพ่วงได้ด้วยความเร็วถึง 1 ล้านบิตต่อวินาที โดยใช้สายรับส่งสัญญาณเพียง 3 หรือ 4 เส้น รวมกับสายกราวด์อีกหนึ่งเส้น

2.6.3 คุณสมบัติ

- 3.1) DS1820 สามารถ Interface โดยใช้สายสัญญาณเพียงเส้นเดียว
- 3.2) DS1820 เพียงตัวเดียว สามารถวัดอุณหภูมิได้โดยไม่ต้องต่ออุปกรณ์ร่วม
- 3.3) DS1820 มีย่านวัดอยู่ที่ +125 ถึง -55 C
- 3.4) DS1820 มีความละเอียดในการวัดได้ 0.5 C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์ที่สนับสนุนระบบบัสเพียงเส้นเดียวจะมีสายสัญญาณเพียง 2 เส้นเท่านั้นคือสายกราวด์และสายสัญญาณ ซึ่งเรียกอีกอย่างว่า สาย DATA สายนี้จะจัดการเกี่ยวกับทั้งสัญญาณข้อมูลและสัญญาณนาฬิกาที่ใช้ในการแลกเปลี่ยนข้อมูล สาย DATA นี้จะเป็นชนิด Open Drain ดังนั้นในการออกแบบวงจร จะต้องออกแบบให้มีตัวต้านทานมาพูลอ์ฟสาย DATA นี้ด้วยให้ดูรูปแผนผังแสดงการต่อระบบบัสของ I²C Wire Bus ดังรูป

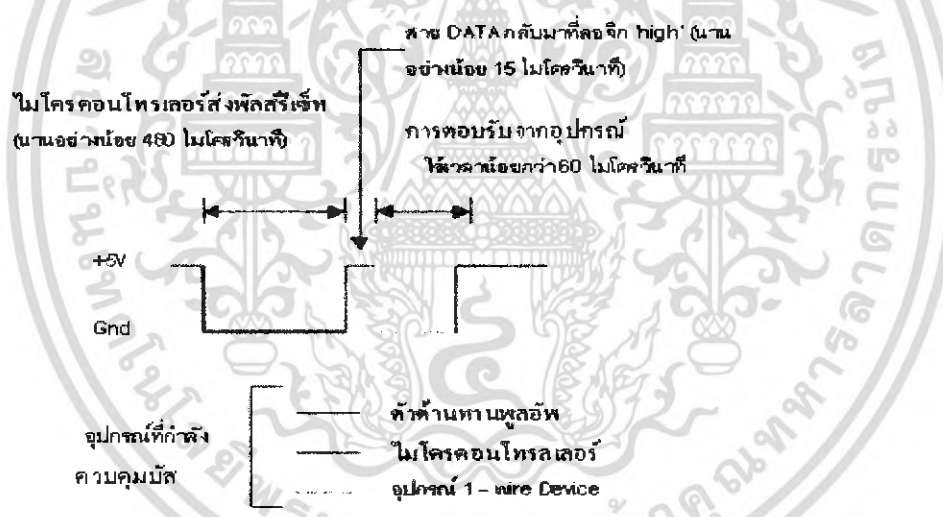


รูป 2-18 แผนผังของระบบบัสแบบ I²C Wire Bus

จากการแลกเปลี่ยนข้อมูลระหว่างไมโครคอนโทรลเลอร์กับอุปกรณ์ที่ใช้ I²C Wire Bus นี้ไม่ได้ง่าย ๆ เหมือนกับการส่งข้อมูลผ่านทางบัสแบบ SPI เพราะในระบบ I²C Wire Bus นั้นสาย DATA จะต้องจัดการเกี่ยวกับจังหวะเวลา (Timing) ระดับสัญญาณ (Level) และทิศทาง (Direction) ของข้อมูลทั้งหมด ทำให้การเขียนซอฟต์แวร์ของไมโครคอนโทรลเลอร์ที่ติดต่อกับอุปกรณ์พวกนี้ต้องมีความซับซ้อนมากขึ้น ในสภาวะพัก (Quiescent State) อุปกรณ์ที่ใช้ระบบ I²C Wire Bus จะทำให้สาย DATA อยู่ในสภาวะลอย (Float) ทำให้ขามีแรงดันเท่ากับแรงดันพูลอ์ฟ (Vcc) ซึ่งปกติก็คือ 5 โวลต์นั่นเอง ส่วนไมโครคอนโทรลเลอร์ก็จะปล่อยให้ขาเอาต์พุตที่ใช้ติดต่อกับขา DATA นี้อยู่ในสภาวะความต้านทานสูง (High-Impedance State) เช่นกัน เมื่ออุปกรณ์ทั้งสองชนิดนี้ปล่อยให้ขา DATA อยู่ในสภาวะลอยแล้วนี้ความต้านทานพูลอ์ฟก็จะช่วยรักษาระดับแรงดันไฟเลี้ยงที่จ่ายให้กับอุปกรณ์ที่ใช้บัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบ 1 – Wire Bus นี้ได้อย่างสม่ำเสมอเพราะอุปกรณ์ที่ใช้มีสแบบ 1 – Wire Bus นี้จะใช้พลังงานในการทำงานน้อยมาก ในการแลกเปลี่ยนข้อมูลกันไมโครคอนโทรลเลอร์และอุปกรณ์ที่ใช้แบบ 1 – Wire Bus จะต้องดำเนินการอย่างระมัดระวังตามลำดับขั้นตอนในการทำให้สาย DATA มีลอจิกเป็น 'low' ปล่อยให้สาย DATA กลับมามีลอจิกเป็น 'high' และตรวจจับการตอบรับกับอุปกรณ์อีกด้านหนึ่ง ช่วงจังหวะเวลาที่ใช้ในกระบวนการนี้จะถูกกำหนดโดยข้อกำหนดเฉพาะของระบบ 1 Wire Bus นี้ซึ่งต้องใช้ไมโครคอนโทรลเลอร์ที่มีการตอบสนองได้อย่างรวดเร็วด้วย ดังนั้นเราจะต้องตรวจสอบความสามารถของระบบให้ดีเสียก่อนที่จะใช้ระบบบัสแบบ 1 – Wire Bus นี้ จากสภาวะพักช่วงต้นการแลกเปลี่ยนข้อมูลจะเริ่มขึ้นด้วยการที่ไมโครคอนโทรลเลอร์กระทำกระบวนการรีเซ็ต (Reset Sequence) ซึ่งทำได้โดยการทำให้สาย DATA มีลอจิก 'low' เป็นเวลาอย่างน้อย 480 ไมโครวินาที แล้วจึงปล่อยให้กลับมามีลอจิกเป็นสภาวะ 'high' อีกครั้งหนึ่ง ดังรูป

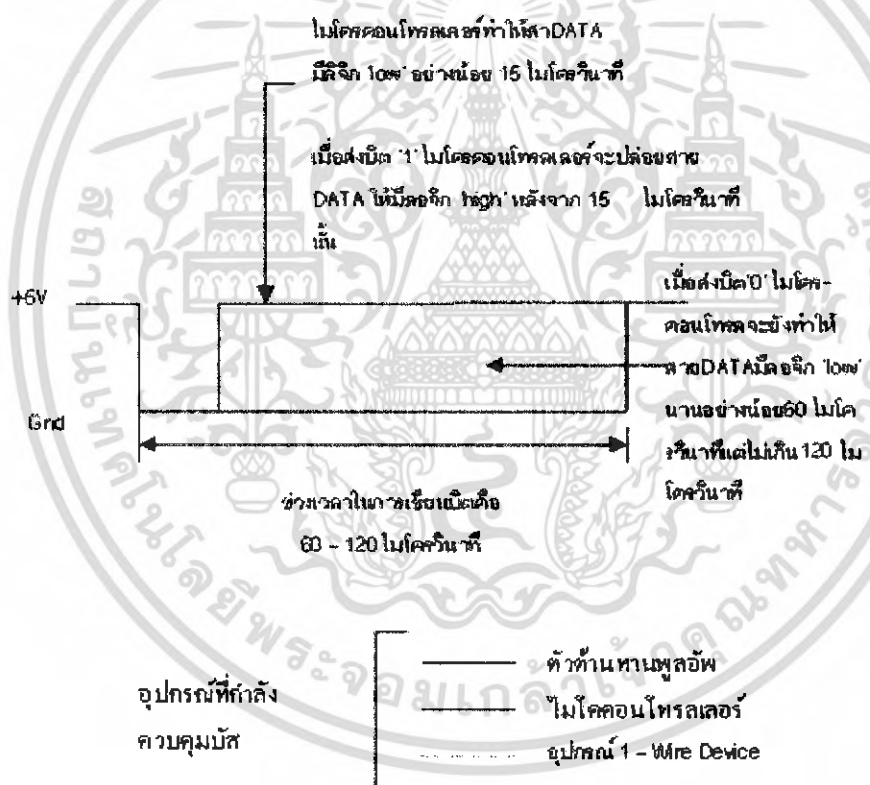


รูป 2-19 จังหวะเวลาในการทำกระบวนการตรวจสอบว่ามีอุปกรณ์อยู่แนบัส

เมื่ออุปกรณ์ที่ใช้มีสแบบ 1 – Wire Bus นี้ตรวจพบสภาวะ RESET นี้ มันก็จะตอบสนองด้วยการส่งพัลส์กลับไปที่บัสเพื่อให้ไมโครคอนโทรลเลอร์รู้ว่าบนสายบัสนี้มีอุปกรณ์แบบ 1 – Wire Device กำลังทำงานอยู่ โดย อุปกรณ์แบบ 1 – Wire Device จะปล่อยให้สาย DATA อยู่ในลอจิก 'high' อย่างน้อย 15 ไมโครวินาที แต่ไม่เกิน 60 ไมโครวินาที จากนั้นมันก็จะให้สาย DATA ลงมามีลอจิกเป็น 'low' ในช่วงเวลาประมาณ 60-240 ไมโครวินาที แล้วจึงปล่อยให้กลับไปที่ลอจิก 'high' เช่นเดิม ช่วงเวลานี้มีชื่อเรียก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

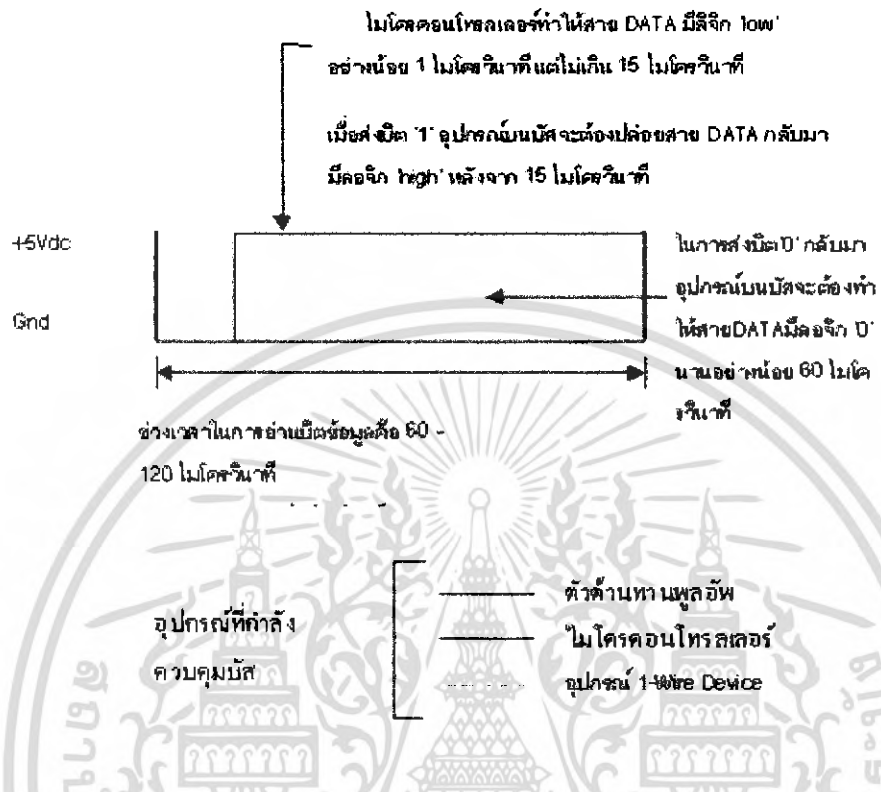
กับหลายชื่อเช่น ช่วงเวลาเริ่มติดต่อ (Initialization) หลังจากอุปกรณ์ 1 Wire Device ปลดปล่อยสาย DATA กลับมาอยู่ในลอจิก 'high' แล้ว ไมโครคอนโทรลเลอร์จะต้องปล่อยให้สาย DATA อยู่ในลอจิกนี้นานอย่างน้อย 240 ไมโครวินาที ต่อไปจากนั้นไมโครคอนโทรลเลอร์ก็จะส่งคำสั่งเริ่มต้น (Initial Command) ขนาด 1 ไบต์ไปยังอุปกรณ์ 1 Wire Bus ซึ่งคำสั่งนี้อาจจะเป็นคำสั่งอะไรก็ได้ ไมโครคอนโทรลเลอร์จะส่งบิตของ คำสั่งนั้นออกไป โดยการเปลี่ยนสถานะของสาย DATA กลับไปกลับมาโดยตอนแรกจะให้ เป็น ลอจิก 'low' แล้วจึงกลับมามีเป็น 'high' ตามช่วงจังหวะเวลาที่เหมาะสม ช่วงเวลานานที่ ไมโครคอนโทรลเลอร์ทำให้สาย DATA มีลอจิก 'low' จะเป็นตัวแยกแยะว่าบิตไหนที่มี ลอจิกเห็น '1' บิตไหนมีลอจิกเป็น '0' ให้ดูแผนภูมิเวลาในการเขียนบิต '1' หรือ '0' ดังรูป



รูป 2-20 จังหวะเวลาที่ไมโครคอนโทรลเลอร์ใช้เขียนข้อมูล '0' หรือ '1' ไปยังอุปกรณ์บนบัส

ช่วงเวลาระหว่าง 15-120 ไมโครวินาที หลังไมโครคอนโทรลเลอร์ทำให้สาย DATA มี ลอจิก 'low' ในตอนนี้จะเรียกว่าช่วงการอ่านสถานะบิตข้อมูล (Sampling Window)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2-21 จังหวะเวลาที่ไมโครคอนโทรลเลอร์ใช้อ่านบิตข้อมูล จากอุปกรณ์บนบัส

ในการอ่านบิตข้อมูลจากบัสแบบ 1-Wire Bus ตอนแรกไมโครคอนโทรลเลอร์จะต้องทำให้สาย DATA มีลอจิก 'low' เป็นเวลานานไม่เกิน 15 ไมโครวินาทีแล้วจึงปล่อยให้สาย DATA กลับมามีลอจิก 'high' เช่นเดิมจากนั้นอุปกรณ์ 1 - Wire Device ก็จะเข้าควบคุมสาย DATA แทนโดยจะส่งบิต '0' โดยการทำให้สาย DATA มีลอจิกเป็น 'low' และส่งบิต '1' โดยการปล่อยให้สาย DATA กลับมามีลอจิก 'high' ตามเดิม เมื่อส่งข้อมูลออกไปยังไมโครคอนโทรลเลอร์เรียบร้อยแล้ว จะมีการพักอยู่ชั่วขณะหนึ่ง จากนั้นอุปกรณ์ 1 - Wire Device จะปล่อยการควบคุมจากสาย DATA ให้เป็นอิสระ และรอรับ คำสั่ง การอ่านของข้อมูลครั้งต่อไป ถ้าอุปกรณ์ 1 - Wire Device ส่งบิต '0' ออกไป ไมโครคอนโทรลเลอร์ก็ จะสามารถตรวจสอบจุดสิ้นสุดของลอจิก '0' นี้ได้ง่าย เพราะสาย DATA จะกลับมาอยู่ที่ลอจิก 'high' ตามเดิม แต่ถ้าตรวจสอบจุดสิ้นสุดของการส่งบิต '1' ของอุปกรณ์ 1 - Wire Device จะต้องใช้เทคนิค มากกว่านี้ เพราะสาย DATA จะอยู่ที่ลอจิก 'high' อยู่แล้วนี่ก็เป็นเหตุผลว่าทำไมจึงหวัะเวลาในการอ่าน เขียนข้อมูลจึงเป็นเรื่องที่สำคัญมาก เมื่อจะอ่านบิต '1' จากอุปกรณ์ 1 - Wire Device

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไมโครคอนโทรลเลอร์จะต้องทำตามเวลาที่แสดงในแผนภูมิเวลาอย่างเคร่งครัด และต้องไม่ทำการอ่านลอจิกของบิตถัดมา จนกว่าจะผ่านเวลาไปแล้วอย่างน้อย 60 ไมโครวินาที

เมื่อได้ศึกษาทฤษฎีจากบทที่ 2 แล้วจะนำไปใช้กับขั้นตอนและวิธีการดำเนินงานและการออกแบบการทดลอง โดยจะแสดงในบทที่ 3



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

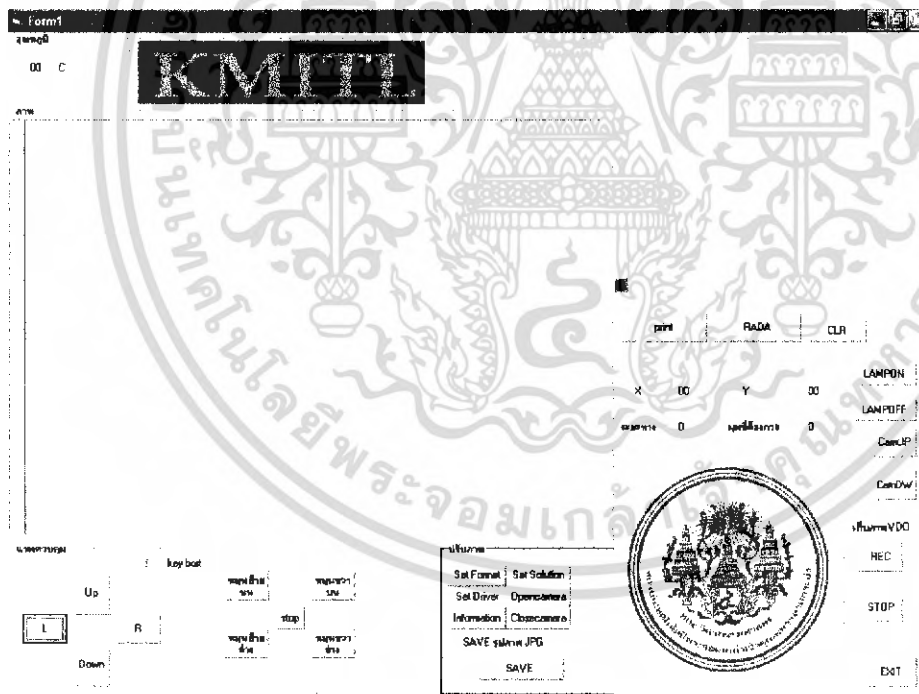
บทที่ 3

การออกแบบและวิธีการดำเนินงานวิจัย

ในบทนี้จะกล่าวถึง หลักการทำงานของหุ่นยนต์ และการออกแบบวงจรอิเล็กทรอนิกส์ รวมถึงการออกแบบโปรแกรมที่ใช้ในโครงงาน ซึ่งจะประกอบด้วย การรับส่งข้อมูลระหว่าง หุ่นยนต์กับคอมพิวเตอร์ส่วนบุคคลโดยคลื่นวิทยุผ่านอุปกรณ์รับ-ส่ง วงจรขับมอเตอร์ และ ส่วนประกอบ ต่าง ๆ ของหุ่นยนต์

3.1 การออกแบบการทำงานของระบบ

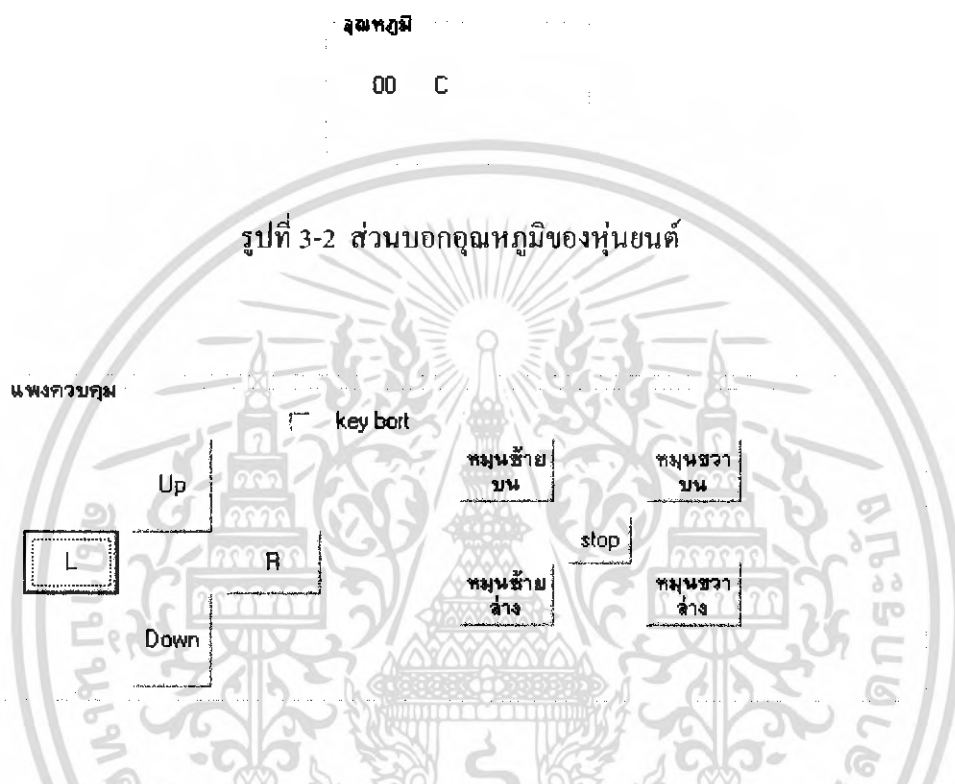
3.1.1 การออกแบบทางด้านซอฟต์แวร์ User Interface



รูปที่ 3-1 Application User Control and Display

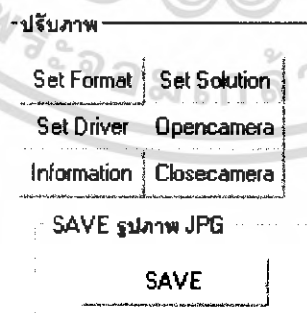
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบในส่วนนี้จะแบ่งออกเป็นส่วนๆ เช่นส่วนแสดงอุณหภูมิ ส่วนควบคุม ส่วนของการปรับภาพ และส่วนแผนที่ เป็นต้น



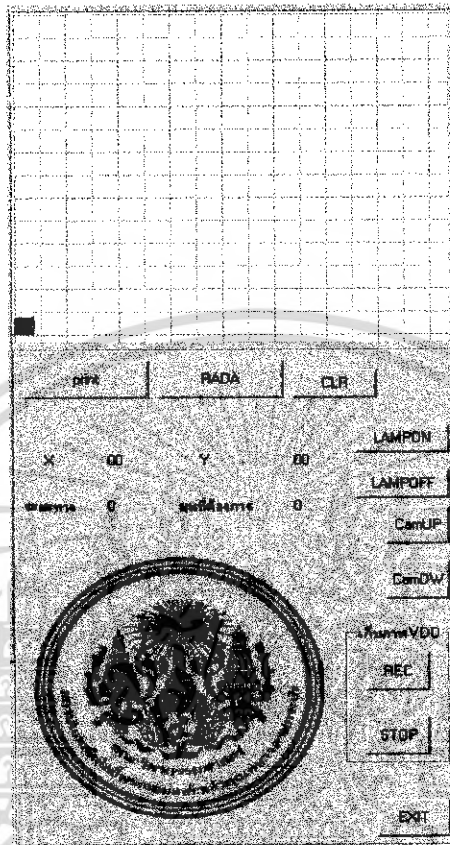
รูปที่ 3-2 ส่วนบอกอุณหภูมิของหุ่นยนต์

รูปที่ 3-3 ส่วนของการควบคุมการเคลื่อนที่และองศา



รูปที่ 3-4 ส่วนของการปรับภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-5 ส่วนของแผนที่ที่หุ่นยนต์เคลื่อนที่

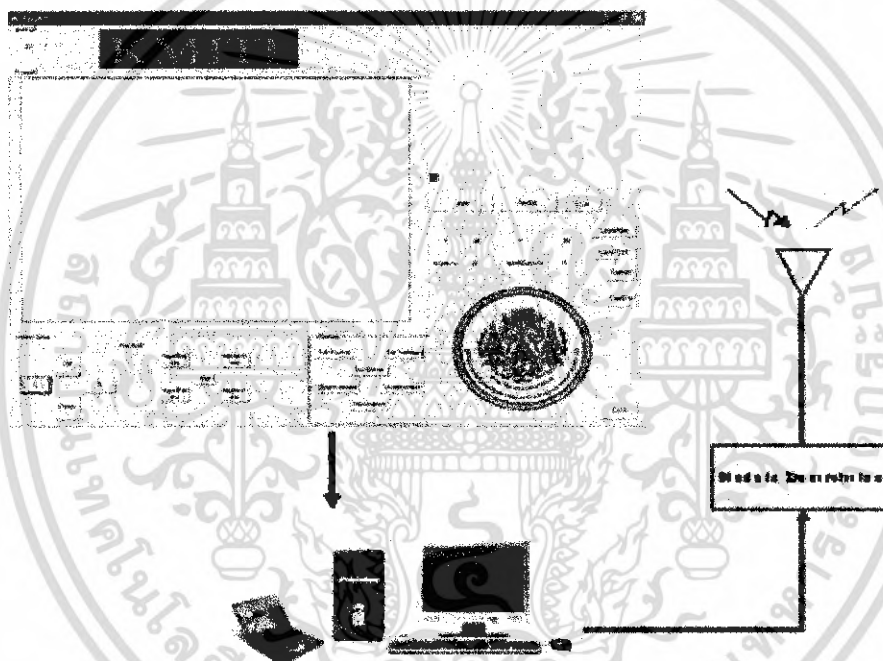


รูปที่ 3-6 ส่วนของจอภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ส่วนของคอมพิวเตอร์และโมดูลเครื่องส่ง

การทำงาน คือ ส่วนของโปรแกรมจะทำการส่งข้อมูลออกจากคอมพิวเตอร์ผ่านพอร์ตอนุกรม และส่งผ่านโมดูลเครื่องส่ง เพื่อไปควบคุม Device Control Unit ให้ทำงานตามที่ต้องการ และในขณะเดียวกันก็จะทำการรับข้อมูลจาก Device Control Unit เข้ามาเพื่อที่จะให้โปรแกรมทำการประมวลผลและคำนวณค่าต่างๆที่ได้มาจาก Device Control Unit เพื่อจะแสดงผลออกมาในรูปของแผนที่ การตรวจจับวัตถุเคลื่อนไหว และตำแหน่งของสิ่งกีดขวางอย่างคร่าวๆ



รูปที่ 3-7 Block User Control and Display

ในส่วนนี้การเขียนโปรแกรมการทำงานจะแสดงตามรายละเอียดใน flow chart โดยสรุปได้ดังนี้

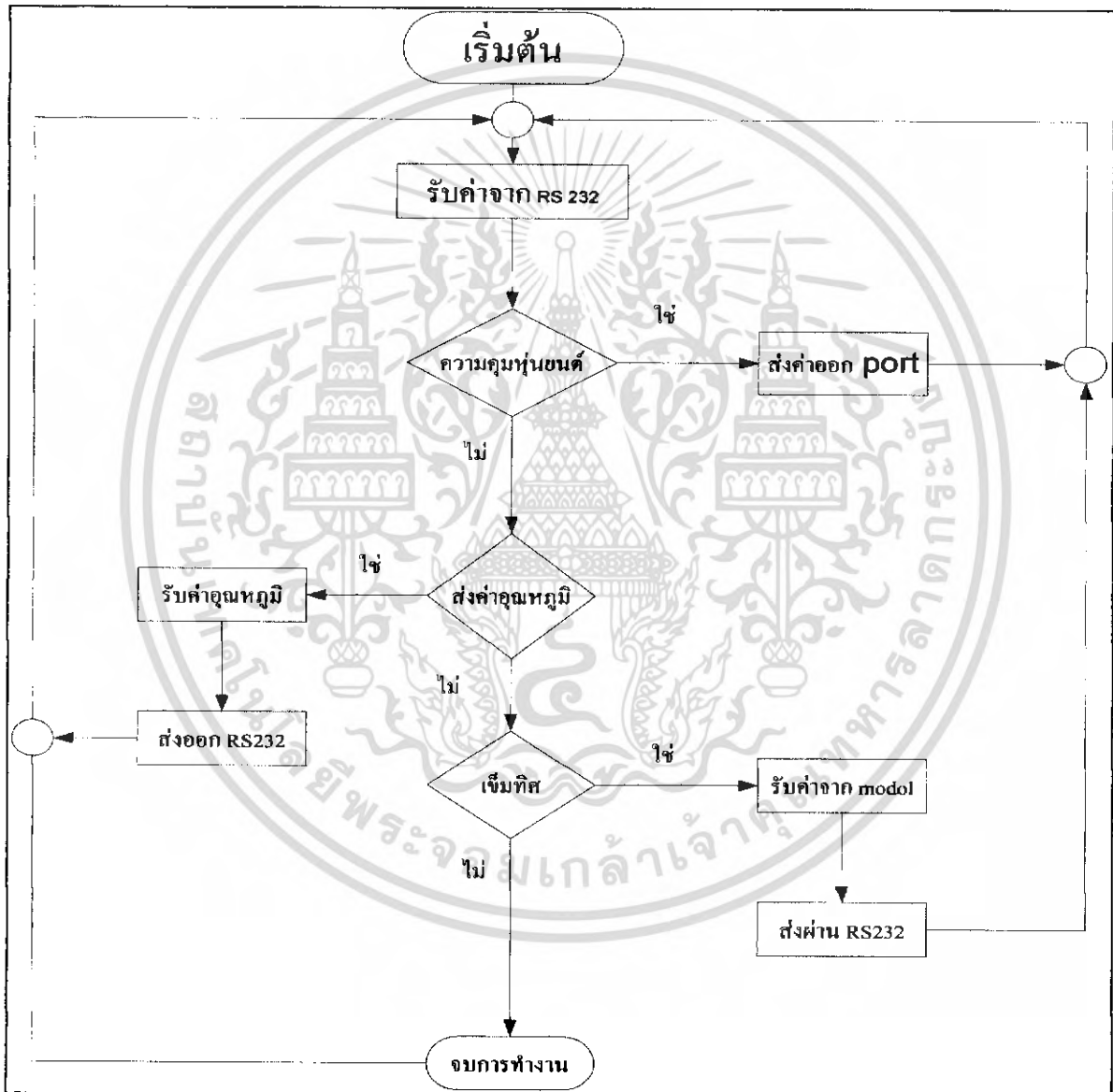
- ทำการกำหนดค่าของพารามิเตอร์ต่างๆ ให้กับ โปรแกรม
- รอรับ Input จาก User โดย User จะเป็นคนสั่งให้โปรแกรมทำอะไร
- ทำการตรวจสอบว่า Input ที่รับมาเข้าเงื่อนไขอะไร เช่น เป็นเงื่อนไขในการเคลื่อนที่ Robot หรือสั่ง Scan Sonar

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ทำการส่งข้อมูลออกไปที่ RF Module ผ่านทางพอร์ทอนุกรม
- รอรับข้อมูลจาก Device Control Unit และแสดงผล

3.1.2 ส่วน Device Control Unit

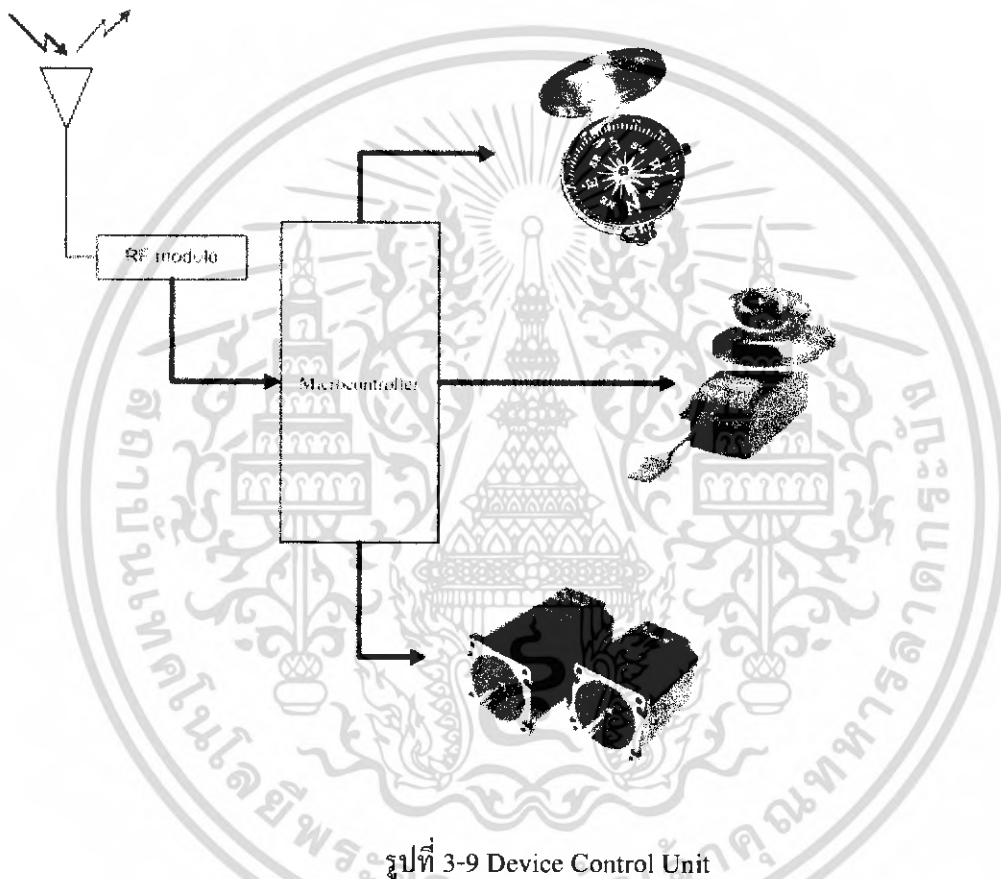
Flow Chart ส่วนของDevice Control Unit แสดงดังนี้



รูปที่ 3-8 Flow Chart ส่วนของDevice Control Unit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากส่วนนี้จะทำการควบคุมอุปกรณ์ต่างๆ โดยจะทำงานสัมพันธ์กับส่วน User Control and Display Unit ส่วนนี้จะทำการรับข้อมูลมาแล้วนำค่าที่รับได้มาทำการประมวลผลตามกระบวนการที่ได้โปรแกรมไว้ เพื่อใช้ควบคุมอุปกรณ์ต่างๆ และในขณะเดียวกันก็จะทำการส่งสัญญาณและข้อมูลต่างๆ กลับไปยังส่วน User Control and Display Unit โดยจะติดต่อได้ทั้งแบบเชื่อมต่อด้วยสายและแบบไร้สาย โดยใช้หลักการสื่อสารผ่านทางพอร์ทอนุกรม



รูปที่ 3-9 Device Control Unit

ในส่วนนี้การเขียนโปรแกรมการทำงานจะแสดงตามรายละเอียดใน flow chart โดยสรุปได้ดังนี้

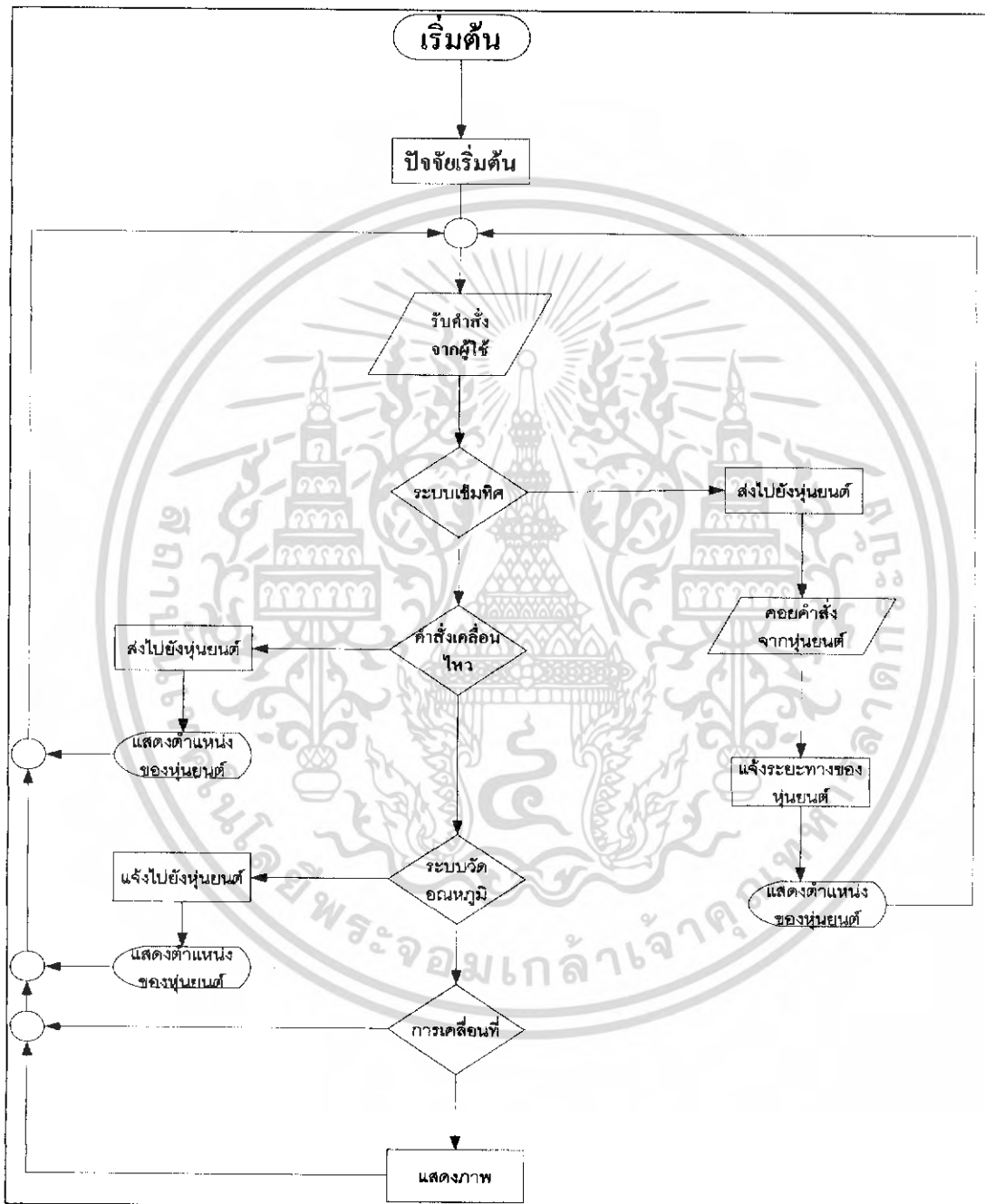
- ทำการกำหนดค่าของพารามิเตอร์ต่างๆ ให้กับ โปรแกรม
- รอรับข้อมูลจากส่วน User Control and Display Unit
- เมื่อรับข้อมูลก็จะทำการตรวจสอบว่าตรงตามเงื่อนไขหรือไม่ เนื่องจากเงื่อนไข

หลักๆมีอยู่ 3 เงื่อนไข ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. เงื่อนไขในการควบคุมการเคลื่อนที่หุ่นยนต์ เป็นฟังก์ชันที่ใช้ควบคุมให้หุ่นยนต์เคลื่อนที่ไปตามทิศทางต่างๆ โดยถ้าข้อมูลที่ส่งมาจากส่วน User Control and Display Unit ตรงกับเงื่อนไขที่ตั้งไว้มันก็จะทำงานตามกระบวนการ
2. เงื่อนไขในการควบคุมเซอร์โวมอเตอร์ จะเป็นฟังก์ชันที่ควบคุมให้เซอร์โวมอเตอร์หมุนไปทางซ้ายและขวา ครั้งละ 5 องศา
3. เงื่อนไขของการควบคุมการ Scan Sonar โดย Sonar จะทำงานได้นั้น เริ่มต้นเมื่อกำหนดพิกัดและกราวด์แล้วเราต้องทำการสร้าง Trigger pulse ให้ Sonar จากนั้นจะทำการสั่งให้ Timer ทำงาน แล้วสุดท้ายก็ทำการตรวจสอบว่ามี Echo pulse กลับมาหรือไม่ ถ้ามี Echo pulse กลับมาแสดงว่า Sonar ทำการตรวจพบวัตถุ ดังนั้นจะได้ค่าเวลา คือ Timer High:TH,Timer Low:TL ออกมาและในช่วงที่รอ Echo pulse ถ้าในระยะเวลา 36 ms ไม่มี Echo pulse กลับมาแสดงว่าตรวจไม่เจอวัตถุในระยะเวลาที่สามารถตรวจได้ (ระยะสูงสุดที่สามารถตรวจได้ขึ้นอยู่กับคุณสมบัติของโซนาร์) เมื่อได้ค่า (Timer high:TH,Timer Low:TL) ก็ทำการส่งกลับไปให้ User Control and Display เพื่อนำค่าไปคำนวณหาระยะทางและสร้างแผนที่ออกมา

Flow Chart ของส่วน User Control & Display Unit แสดงได้ดังนี้



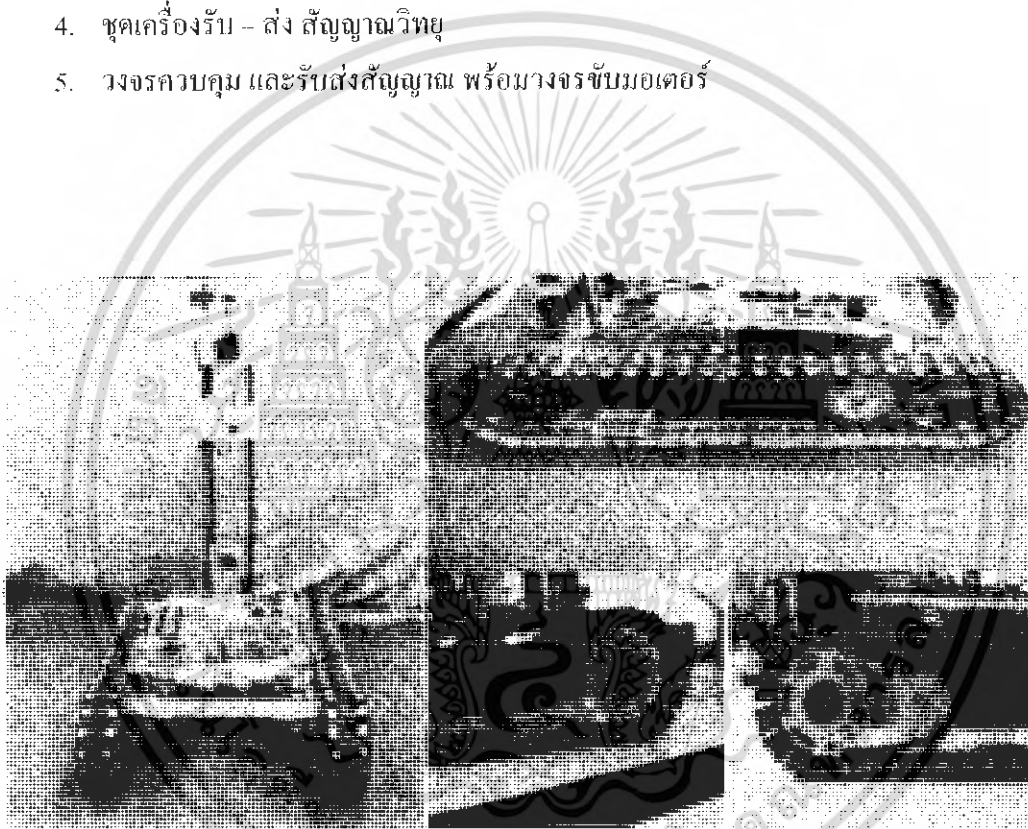
รูปที่ 3-10 Flow Chart ของส่วน User Control & Display Unit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 อุปกรณ์ที่ใช้ในการทดลอง

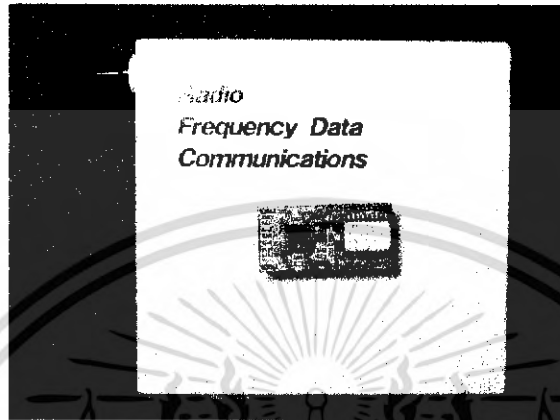
อุปกรณ์ที่ใช้ในการทดลองมีดังนี้

1. คอมพิวเตอร์ส่วนบุคคล
2. โครงหุ่นยนต์ขนาด 45*65*69 เซนติเมตร
3. คีชีมอเตอร์
4. ชุดเครื่องรับ – ส่ง สัญญาณวิทยุ
5. วงจรควบคุม และรับส่งสัญญาณ พร้อมวงจรขับมอเตอร์

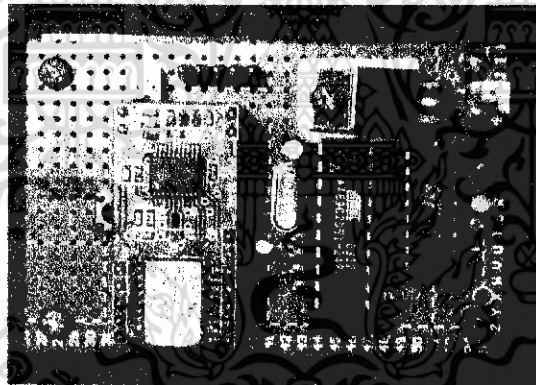


รูปที่ 3-11 โครงหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-12 ชุดเครื่องรับ-ส่งสัญญาณวิทยุ



รูปที่ 3-13 วงจรควบคุม และรับส่งสัญญาณพร้อมวงจรขับมอเตอร์

เมื่อทำการสร้างวงจร และตัวหุ่นยนต์เรียบร้อยแล้วขั้นตอนต่อไปเป็นการทดลองการควบคุม การเคลื่อนที่ของหุ่นยนต์ด้วยการทำงานแบบโคดเคียว และการทำงานแบบมัลติเอเจนต์โดยแบล็คบอร์ด เพื่อทำการเปรียบเทียบประสิทธิภาพในสถานการณ์ต่าง ๆ ซึ่งจะได้อีกถึงในบทที่ 4 ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 หลักการของการเขียนโปรแกรมขับเคลื่อนมอเตอร์

เมื่อเรารู้ว่าการทำงานของมอเตอร์คือต้องจ่ายแรงดันไฟฟ้ากระแสตรงให้ โดยเราจะอาศัยการจ่ายลอจิกจากไมโครคอนโทรเลอร์ไปที่วงจรขับเคลื่อนมอเตอร์(แสดงในภาคผนวก) โดยวงจรขับเคลื่อนมอเตอร์ก็จะทำการขับเคลื่อนมอเตอร์ตามที่ได้สัญญาณมาจากไมโครคอนโทรเลอร์ โดยเราจะทำการจ่ายลอจิกการทำงาน โดยมีสองสถานะคือ 0 กับ 1 โดยถ้าจ่ายลอจิกกลับด้านกันก็จะทำให้มอเตอร์หมุนกลับทางนั่นเอง

ตัวอย่างโปรแกรม

```

;-----Start DC motor Control -----;
Loop_forward: CLR P2.0
               SETB P2.1
               SETB P2.2
               CLR P2.3
               ACALL Delay_100MS
               LJMP Wait_check
;-----;
Loop_reverse: SETB P2.0
               CLR P2.1
               CLR P2.2
               SETB P2.3
               ACALL Delay_100MS
               LJMP Wait_check

```

หมายเหตุ ส่วนสัญญาณจ่ายออกมาจากไมโครคอนโทรเลอร์จะถูกจ่ายออกมาตามที่ได้ออกแบบวงจรไว้ ให้ทำให้เรารู้ว่าจะเขียนโปรแกรมอย่างไร ส่วนวงจรสามารถอ้างอิงได้จากภาคผนวก

3.4 การควบคุมและใช้งานเข็มทิศ

โมดูลเข็มทิศดิจิทัล CMPS03 ออกแบบมาเพื่อช่วยในการกำหนดทิศทางเคลื่อนที่ของหุ่นยนต์อัตโนมัติ และนำมาใช้ในการสร้างเครื่องมือวัดและตรวจสอบที่ระบบอิเล็กทรอนิกส์โดยหัวใจสำคัญของโมดูลเข็มทิศคือตัวตรวจจับสนามแม่เหล็กโลก (Earth magnetic field) และไมโครคอนโทรลเลอร์เพื่อรับสัญญาณจากตัวตรวจจับมาประมวลผลเป็นข้อมูลดิจิทัลและสัญญาณพัลส์สำหรับแจ้งผลการวัด

การอ่านค่าทิศทางจากเอาต์พุตสัญญาณพัลส์ อ่านได้โดยหา PWM ของโมดูลโดยการนำค่าความกว้างของพัลส์ที่ได้จากเอาต์พุตสัญญาณพัลส์ของโมดูลมาระบุตำแหน่งองศาจาก 0 ถึง 359.9 องศาโดยมีย่านของค่าความกว้างสัญญาณพัลส์จาก 1 มิลลิวินาทีไปจนถึง 36.99 มิลลิวินาทีที่มีความละเอียด 0.1 มิลลิวินาทีต่อองศาในสัญญาณพัลส์แต่ละไซเคิลมีช่วงลอคจิก 0 กว้าง 65 มิลลิวินาที ดังนั้นในการนำสัญญาณพัลส์มาประมวลผลเป็นค่ามุมจึงต้องใช้นับค่าความกว้างของสัญญาณพัลส์เป็นหลักในการคำนวณค่าที่โมดูล

3.5 การแสดงอุณหภูมิ

ไอซี DS18B20 เป็นไอซีที่มีระบบการสื่อสารข้อมูลอนุกรมแบบหนึ่งสายซึ่งถือได้ว่าเป็นระบบที่มีความชาญฉลาด และใช้จำนวนสายสัญญาณเพียง 1 เส้นเท่านั้น โดยไม่ต้องมีสายสัญญาณนาฬิกา มาควบคุมจังหวะการถ่ายทอข้อมูลเหมือนกับระบบสื่อสารข้อมูลอนุกรมในแบบอื่น สายข้อมูลจะทำหน้าที่เสมือนเป็นสายสัญญาณนาฬิกาในตัว ส่วนค่าของข้อมูลจะพิจารณาจากลักษณะของรูปสัญญาณที่ปรากฏบนสายสัญญาณในแต่ละช่องของเวลาซึ่งเรียกว่า ไทม์สล็อต (Time Slot) โดยคาบเวลาดำสุดและสูงสุดของสถานะต่าง ๆ ในการสื่อสารข้อมูลในแต่ละไทม์สล็อตมีการกำหนดขอบเขตไว้อย่างชัดเจนการถ่ายทอข้อมูลจะเกิดขึ้นในแต่ละไทม์สล็อตนั้น รูปแบบการถ่ายทอ ข้อมูลจะเป็นแบบอะซิงโครนัสในระดับบิต ไม่มีการกำหนดความยาวของข้อมูลเป็นระดับไบต์ระบบสื่อสารแบบนี้เหมาะที่จะใช้ในการสื่อสารข้อมูลระหว่างไอซีแผงวงจรเดียวกัน แล้วค่าที่ได้ก็จะถูกส่งมายังหน้าจอคอมพิวเตอร์บอกอุณหภูมิต่อผู้ควบคุมหุ่นยนต์

3.6การใช้งาน Component ของกล้อง

การออกแบบวิธีการตรวจจับภาพ จะมีวิธีการดังนี้ ใช้ component ที่ชื่อว่า VideoOCX ในการติดต่อกับกล้องผ่านทางโปรแกรม Visual Basic จากนั้นทำการเก็บภาพที่ต้องการได้และทำการถ่ายวิดีโอเก็บไว้แสดงผลในภายหลังได้

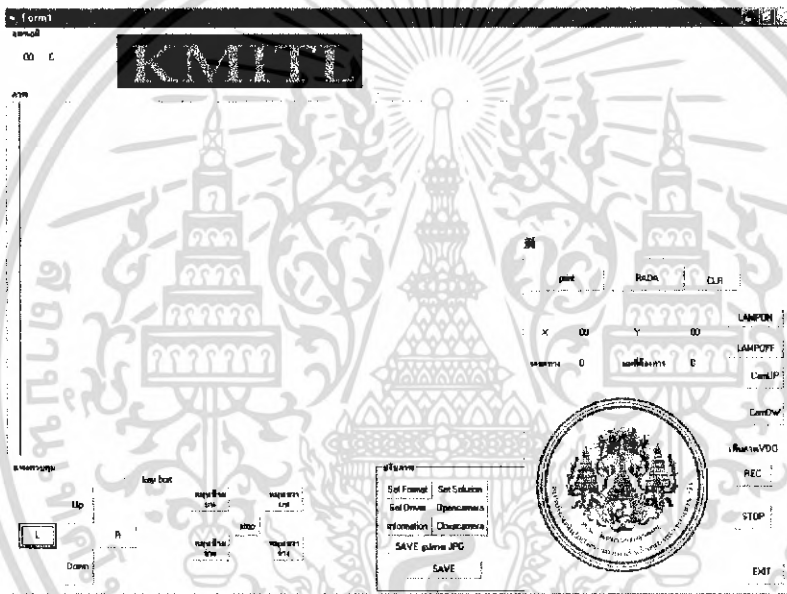
เมื่อทำการสร้างวงจร และตัวหุ่นยนต์เรียบร้อยแล้วขั้นต่อไปเป็นการทดลองการควบคุมการเคลื่อนที่ของหุ่นยนต์ด้วยการทำงานแบบ โคดเค็ชิว และการทำงานแบบมัลติเอเจนต์โดยแบล็คบอร์ด เพื่อทำการเปรียบเทียบประสิทธิภาพในสถานการณ์ต่าง ๆ ซึ่งจะได้กล่าวถึงในบทที่ 4 ต่อไป



บทที่ 4

ผลการทดลอง

จากการออกแบบโปรแกรมสำหรับการควบคุมและแสดงผลที่ใช้ในการติดต่อกับ Device Control Unit เมื่อทำการรันโปรแกรมก็จะได้ผลตามที่ได้อธิบายไว้ในบทที่ 3 โดยจากรูปภาพข้างล่างก็จะแสดงผลจากการรันการทำงาน เมื่อรันโปรแกรมจะมีการเซตค่าต่างๆ และจะได้ผลที่ปรากฏตามรูปที่ 4-1



รูปที่ 4-1 ผลการทดสอบโปรแกรมโดยรวมของส่วน User Control & Display Unit

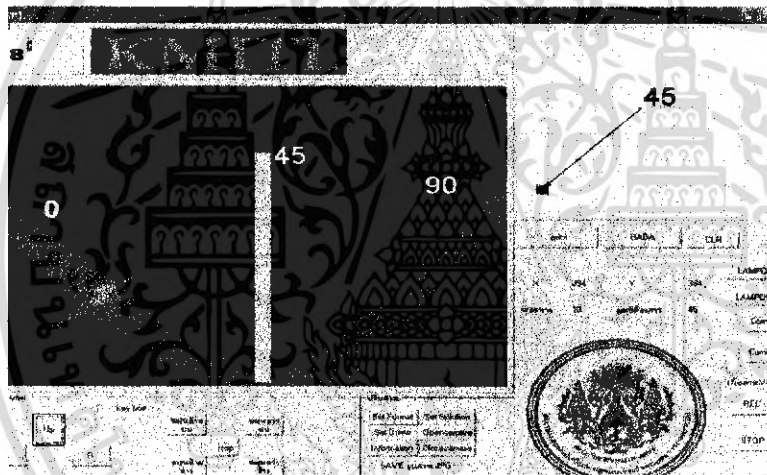
จากผลการทดลองตามรูปที่ 4-1 เราจะสามารถแยกการทำงานออกเป็นส่วนๆและจะอธิบายการทำงานและผลการทดลองที่ได้ ตามขั้นตอนดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1 ผลการทดลองตามวัตถุประสงค์หลัก

การทำงานหลักๆของโปรแกรมตามที่ได้ตั้งเป้าหมายตามวัตถุประสงค์ที่ตั้งไว้จะมีอยู่ 3 ส่วนหลักๆ คือ การใช้ระบบเข็มทิศเพื่อหาตำแหน่งของของหุ่นยนต์ ส่วนของการตรวจจับภาพ และเคลื่อนที่ข้ามสิ่งกีดขวาง โดยจะแสดงผลการทดลองดังต่อไปนี้

4.1.1 ส่วนของการอ่านค่าจากระบบเข็มทิศ



รูปที่ 4-2 การอ่านค่าจากเข็มทิศและบอกพิกัดบนแผนที่

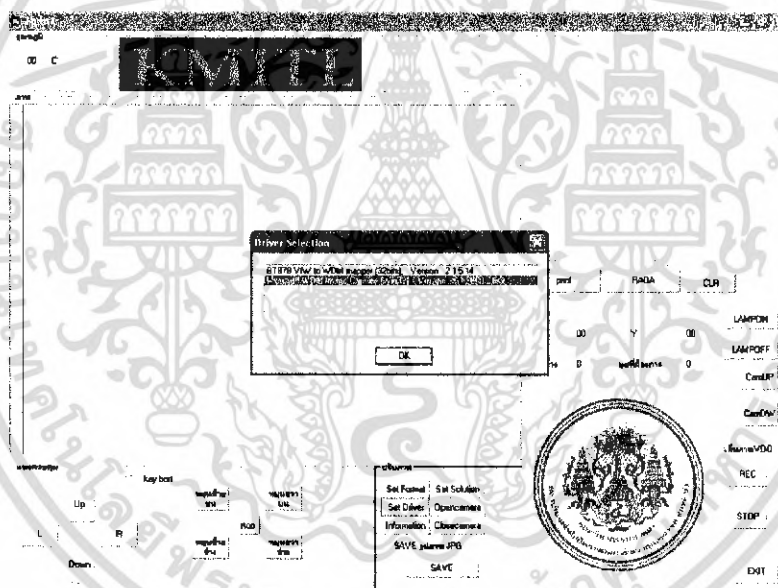
จากรูปสามารถอ่าน โดยการทดลองตีเส้น โดยให้หุ่นยนต์เดินตามเส้นขาว โดยจากการวัดที่ด้านซ้ายจะเป็นการบอกตำแหน่งที่ 0 องศา โดยที่หุ่นยนต์หันหน้าเป็นมุมเป็นมุม 45 องศา เส้นศูนย์ ซึ่งถ้าต้องการให้หุ่นยนต์เดินต่อไปต้องให้หุ่นยนต์หมุนอีก 45 องศาจึงสามารถไปต่อได้ค่าพิกัดที่ได้จะถูกแปลงให้อยู่ใน พิกัด XYทำให้สามารถอ่านค่าได้ง่าย และแสดงผล

4.1.2. ส่วนของการใช้งานกล้องผ่านทาง VideoOCX

4.1.2.1 ส่วนเมตร็อคของ VideoOCX ทำหน้าที่ในการเช็คค่าต่างๆของกล้องได้ให้เหมาะสมตามผู้ใช้ต้องการ

- Set Driver ใช้เช็ค driver ของกล้อง
- Set Format ใช้เช็คขนาดของภาพที่รับเข้ามา, สีของภาพ
- Set Solution ใช้เช็ค source ของกล้อง
- Set Information แสดงข้อมูลที่เกี่ยวข้องกับกล้อง

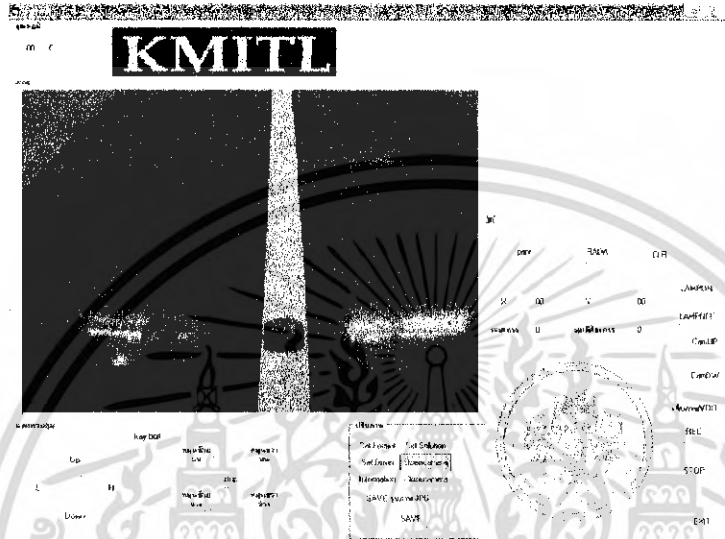
ผลที่ได้เมื่อรันแต่ละเมตร็อคของ VideoOCX จะเป็นดังนี้



รูปที่ 4-3 Method ของการเช็ค Driver ของกล้อง

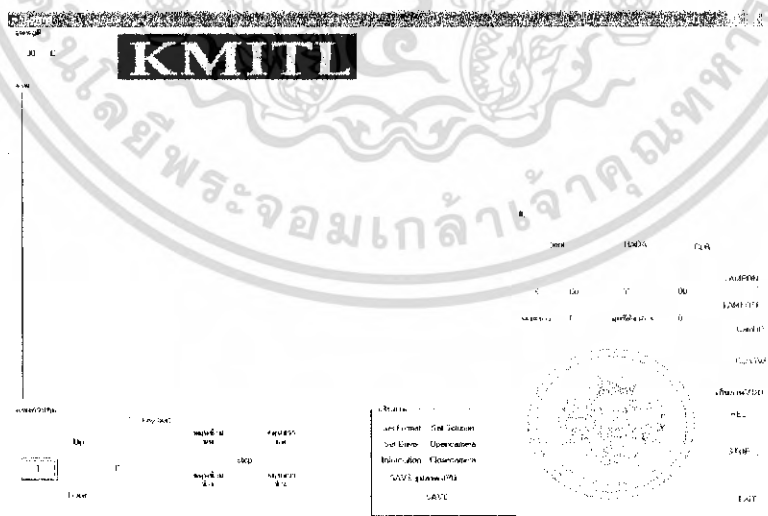
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2.2 Oncamera ทำหน้าที่ส่งให้กล้องทำงานโดยรับภาพแล้วแสดงที่หน้าจอ โดยผลที่ได้จะทำให้เห็นภาพที่หน้าจอ



รูปที่ 4-4 หน้าจอของ โปรแกรมเมื่อทำการเปิดกล้อง

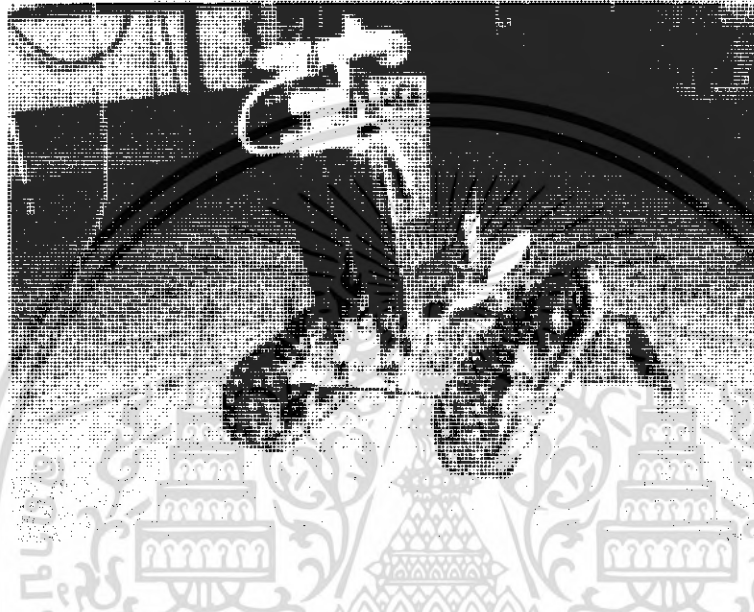
4.1.2.3 Offcamera ทำหน้าที่เรียกปิดการทำงานของกล้องส่งผลให้โปรแกรมหยุดแสดงภาพที่รับมาจากกล้อง



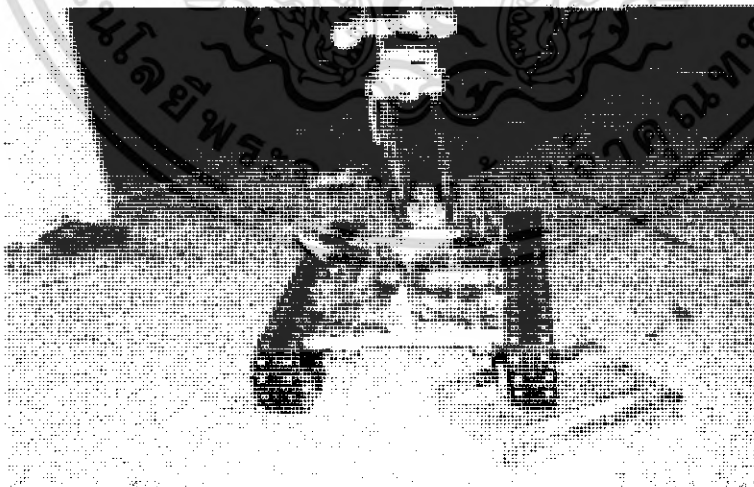
รูปที่ 4-5 หน้าจอของ โปรแกรมเมื่อทำการปิดกล้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การขำมถึงกิดขวางของหุ่นยนต์

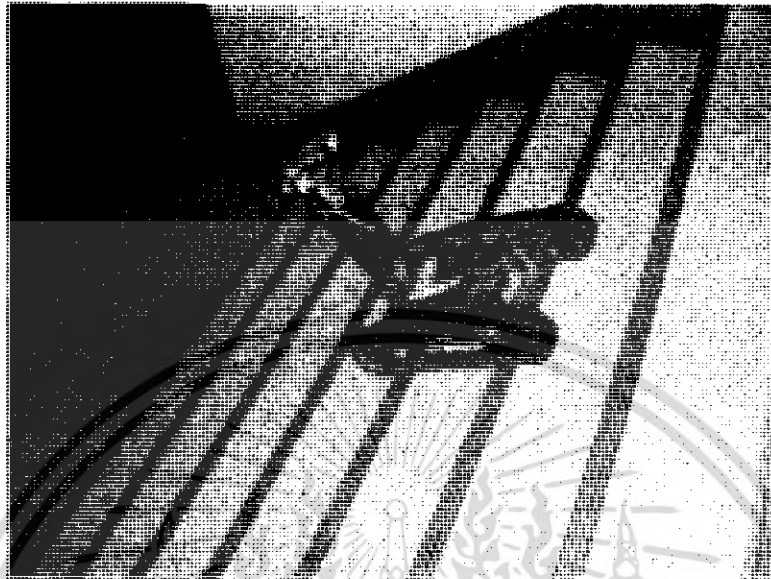


รูปที่ 4-6 หุ่นยนต์ขำมสิ่งกิดขวางด้วยตีนตะขามด้านขวา

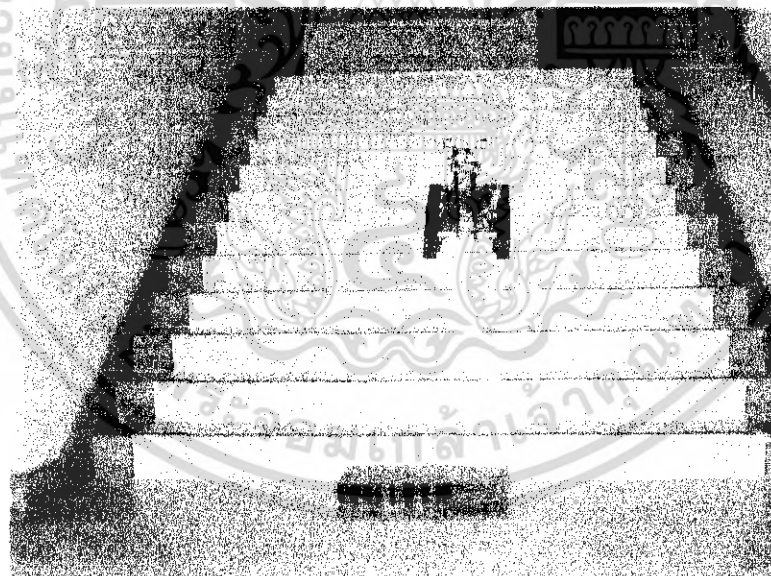


รูปที่ 4-7 หุ่นยนต์ขำมสิ่งกิดขวางด้วยตีนตะขามด้านซ้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



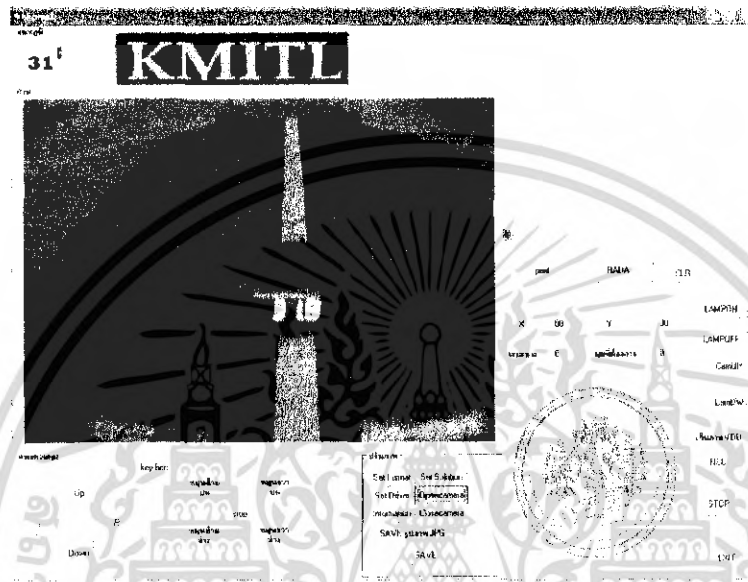
รูปที่ 4-8 หุ่นยนต์ขึ้นบันได



รูปที่ 4-9 หุ่นยนต์ลงบันได

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การแสดงอุณหภูมิ



รูปที่ 4-10 เทียบอุณหภูมิระหว่างตัวหุ่นยนต์กับค่าที่ส่งแสดงผลหน้าจอ

จากงานวิจัยที่ได้นำเสนอมาทั้งหมด 4 บทนี้ ในการทำการทดลองได้พบปัญหาต่างๆที่เกิดขึ้นกับงานวิจัยนี้และได้ทำการแก้ไขปัญหาและหาข้อสรุปออกมา โดยปัญหาต่างๆและข้อสรุปรวมไปถึงข้อเสนอแนะต่างๆได้แสดงไว้ในบทที่ 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทสรุปและข้อเสนอแนะ

จากผลการทดลองที่ได้ในงานวิจัยที่ 1 คือ สามารถทำได้ตามจุดประสงค์และขอบเขตข้อที่ 1 คือ ในเรื่องของระบบแจ้งเตือนและโครงสร้างได้ผลตามแสดงไว้ในบทที่ 4 คือ จะสามารถใช้แจ้งเตือนทำระบุพิกัดของหุ่นยนต์บอกอุณหภูมิและสามารถข้ามสิ่งกีดขวางได้

ส่วนในจุดประสงค์และขอบเขตข้อที่ 2 ในเรื่องของการตรวจสอบวัตถุเคลื่อนไหวนั้นยังไม่สามารถทำงานได้สมบูรณ์ซึ่งหลังจากงานวิจัย 1 จึงได้ทำการพัฒนาต่อจนได้ผลสรุปดังนี้

5.1 ผลสรุปและปัญหาในงานวิจัย

จากผลการทดลองที่ได้ จะรวมผลการทดลองทั้งหมดจากงานวิจัย 1 และ 2 ซึ่งจะสรุปผลจากการทำงานในส่วนงานวิจัยดังนี้

ก) ในส่วนงานวิจัยที่ 1 จะทำเฉพาะส่วนของระบบแจ้งเตือน การสร้างแผนที่ วัตถุอุณหภูมิการเคลื่อนที่ และการออกแบบ Graphic User Interface (GUI) ติดต่อกับผู้ใช้ ซึ่งจากการทดลองงานวิจัยสามารถทำงานได้ตามวัตถุประสงค์ที่ตั้งไว้ซึ่งผลการทดลองทั้งหมดได้แสดงในบทที่ 4

ในส่วนปัญหาที่พบในงานวิจัยที่ 1

- จะเป็นในเรื่องของงบประมาณที่ใช้ในการทำงานวิจัยซึ่งทำให้งานวิจัยสามารถทำงานได้ในระดับหนึ่งเท่านั้นซึ่งยังไม่ดีที่สุด
- ส่วนการควบคุมให้หุ่นยนต์เคลื่อนที่ไปตามทิศทางที่ต้องการนั้นสามารถทำได้ไม่เท่าที่ควรเพราะว่าตัวหุ่นยนต์มีการเคลื่อนที่ที่ไม่มีประสิทธิภาพจึงควรนำระบบที่สร้างขึ้นมาไปใช้กับหุ่นยนต์ที่มีการควบคุมที่มีประสิทธิภาพสูงกว่านี้จะทำให้ทำงานได้สมบูรณ์ขึ้น
- การตรวจหาระยะของวัตถุด้วยระบบแจ้งเตือนนั้นจะมีค่าความผิดพลาดเกิดขึ้นซึ่งเป็นข้อผิดพลาดที่สามารถเกิดขึ้นได้ดังนั้นถ้าต้องการตรวจหาพิกัดที่จะเกิดค่าผิดพลาดน้อยจะต้องทำการหาระยะนั้นหลายๆ ครั้งเพื่อนำมาหาค่าเฉลี่ย

ข) ในส่วนงานวิจัยที่ 2 จะเพิ่มส่วนของการวัดอุณหภูมิ การตรวจจับวัตถุเคลื่อนไหวด้วยกล้องปรับปรุงการทำงานในส่วนของการเคลื่อนที่ของหุ่นยนต์ให้ดีขึ้น และปรับปรุง GUI ให้ดีขึ้น ซึ่งจาก

การทดลองงานวิจัยสามารถทำงานได้ตามวัตถุประสงค์ที่ตั้งไว้ซึ่งผลการทดลองทั้งหมดได้แสดงในบทที่ 4

ในส่วนปัญหาที่พบในงานวิจัยที่ 2

- การใช้กล้องไร้สาย จะมีข้อจำกัดในเรื่องของระยะทางและสัญญาณรบกวน ซึ่งมีผลกระทบโดยตรงกับการตรวจจับวัตถุเคลื่อนไหว
- การใช้งานคอมพิวเตอร์ที่ติดต่อกับกล้องผ่านโปรแกรม Visual Basic จะมีปัญหาในช่วงแรกที่เริ่มใช้

5.2 ข้อเสนอแนะและการพัฒนาระบบ

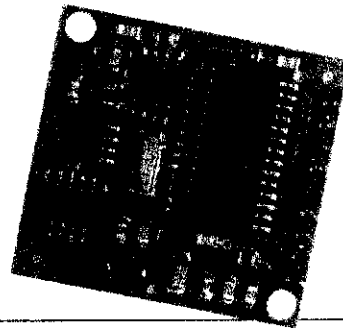
ในงานวิจัยนี้การทำงานจะเป็นระบบ Manual ซึ่งถ้ามีระบบนี้ไปพัฒนาและใช้กับหุ่นยนต์ที่มีประสิทธิภาพในการเคลื่อนที่, ลักษณะทางกายภาพของหุ่นยนต์ควรจะแข็งแรงเพื่อที่จะสามารถเข้าทำงานในพื้นที่จริงได้จะทำให้หุ่นยนต์มีความสามารถสูงขึ้น ระบบนี้ยังสามารถพัฒนาเป็นระบบอัตโนมัติได้ถ้ามีการออกแบบและใช้เทคโนโลยีขั้นสูงเข้ามาช่วย

บรรณานุกรม

- [1] อภิชาติ ภูพลับ, “เริ่มต้นเขียนโปรแกรมติดต่อและควบคุมฮาร์ดแวร์ด้วย Visual Basic”, อินโฟเพรส, 2546
- [2] วรพจน์ กรแก้ววัฒนกุล, ชัยวัฒน์ ลัมพรจิตวิไล, “เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช”, บริษัท อินโนเวทีฟ เอ็กเพอริเมนต์ จำกัด, 2521
- [3] ชีรวัฒน์ ประกอบผล, “การประยุกต์ใช้งานไมโครคอนโทรลเลอร์”, สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น), 2543
- [4] กิตติ ภัคดีวัฒนกุล, จำลอง ครุฑุตสาหะ, “Visual Basic 6 ฉบับครบโปรแกรมเมอร์”, บริษัท เคทีพี คอมพ์ แอนด์ คอนซัลท์ จำกัด, 2542
- [5] Rafael C. Gonzalez, Richard E. Woods, “Digital Image Processing 2nd Edition”, Prentice Hall, 2002
- http://www.msclub.ce.cetpu.edu.ru/MCU_MPU/MCS51/pa_mcs51.html [1]
- http://www.inex.co.th/store/cmpps03_sheet-thai.pdf [2]
- <http://www.superrobotica.com/S330160.htm>[3]
- <http://www.hbfydz.com/xxyd/201.htm>[4]

CMPS03

Digital Compass Module โมดูลเข็มทิศดิจิทัล



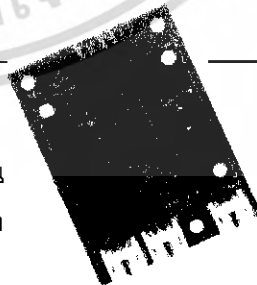
Distributed by: Innovative Experiment Co.,Ltd., Thailand

คุณสมบัติ

- ใช้ไฟเลี้ยง +5V ต้องการกระแสไฟฟ้า 20mA
- ใช้ตัวตรวจจับสนามแม่เหล็กเบอร์ KMZ51 ของ Philips จำนวน 2 ตัว เพื่อให้สามารถตรวจจับสนามแม่เหล็กโลกได้อย่างสมบูรณ์และมีความละเอียดมากเพียงพอ
- ความละเอียดของมุม 0.1 องศา
- ค่าความผิดพลาด 3-4 องศา โดยประมาณ หลังจากการปรับแต่ง
- เอาต์พุตแบบสัญญาณพัลส์ ความกว้าง 1 ถึง 37 มิลลิวินาที โดยมีอัตราเพิ่มครั้งละ 0.1 มิลลิวินาที
- เอาต์พุตข้อมูลดิจิทัลผ่านการติดต่อระบบบัส I²C รองรับสัญญาณนาฬิกาความถี่สูงถึง 1MHz โดยให้ข้อมูล 2 รูปแบบคือ 0-255 และ 0-3599
- ขนาดเล็กเพียง 32 x 35 มิลลิเมตร
- สื่อสารกับไมโครคอนโทรลเลอร์ยอดนิยมได้ทุกตระกูล อาทิ เมลิกแอสตมป์ 2SX/2P, PIC, MCS-51, PSoC, 68HC11 ทั้งผ่านระบบบัส I²C และด้วยการวัดสัญญาณพัลส์

อุปกรณ์เสริม

- บอร์ด ADX-CMPS03 ซึ่งเป็นบอร์ดอะแดปเตอร์สำหรับอำนวยความสะดวกในการเชื่อมต่อกับบอร์ดไมโครคอนโทรลเลอร์และแผงต่อวงจรหรือเบรตบอร์ด
- สาย PCB3A สำหรับเชื่อมต่อกับบอร์ดควบคุมหุ่นยนต์ของ i-nex



๒๒

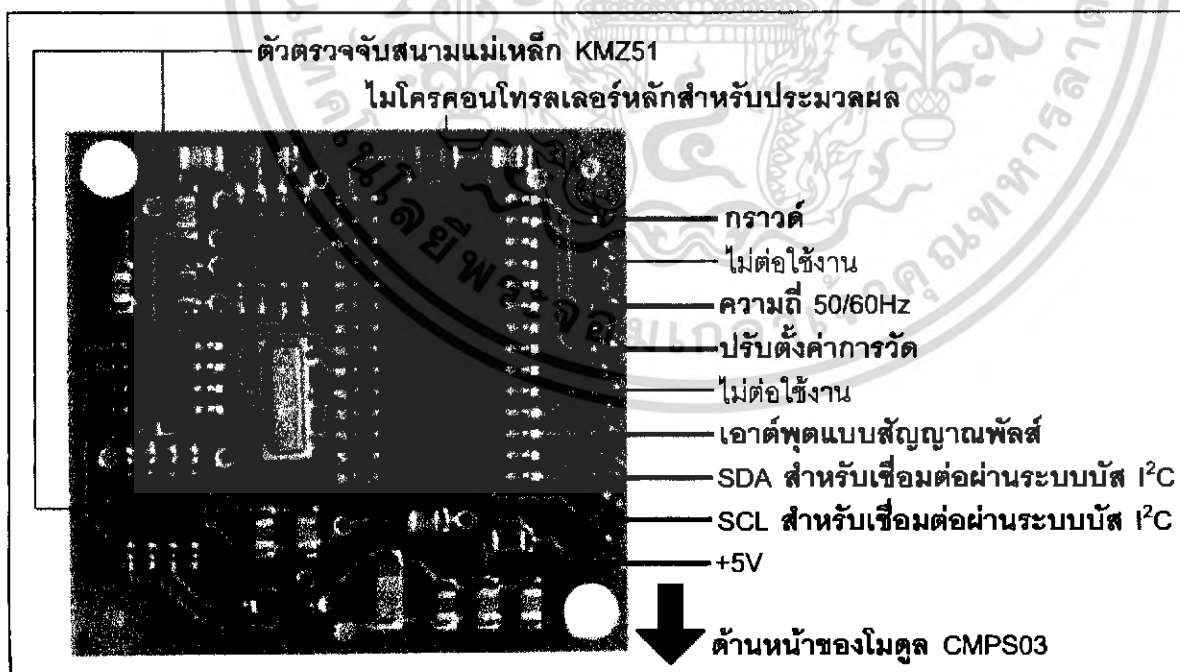
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โมดูลเซ็นเซอร์ทิศทาง CMPS03 เป็นผลงานของ Devantech (www.radio-electronics.co.uk) ออกแบบมาเพื่อช่วยในการกำหนดทิศทางเคลื่อนที่ของหุ่นยนต์อัตโนมัติ และนำมาใช้ในการสร้างเครื่องมือวัดและตรวจสอบที่ระบบอิเล็กทรอนิกส์ โดยหัวใจสำคัญของโมดูล CMPS03 คือ ตัวตรวจจับสนามแม่เหล็กเบอร์ KMZ51 ของ Philips จำนวน 2 ตัว เพื่อให้มีความไวเพียงพอในการตรวจจับสนามแม่เหล็กโลก (Earth magnetic field) และไมโครคอนโทรลเลอร์เพื่อรับสัญญาณจากตัวตรวจจับ มาประมวลผลเป็นข้อมูลดิจิทัลและสัญญาณพัลส์สำหรับแจ้งผลการวัดทิศทาง

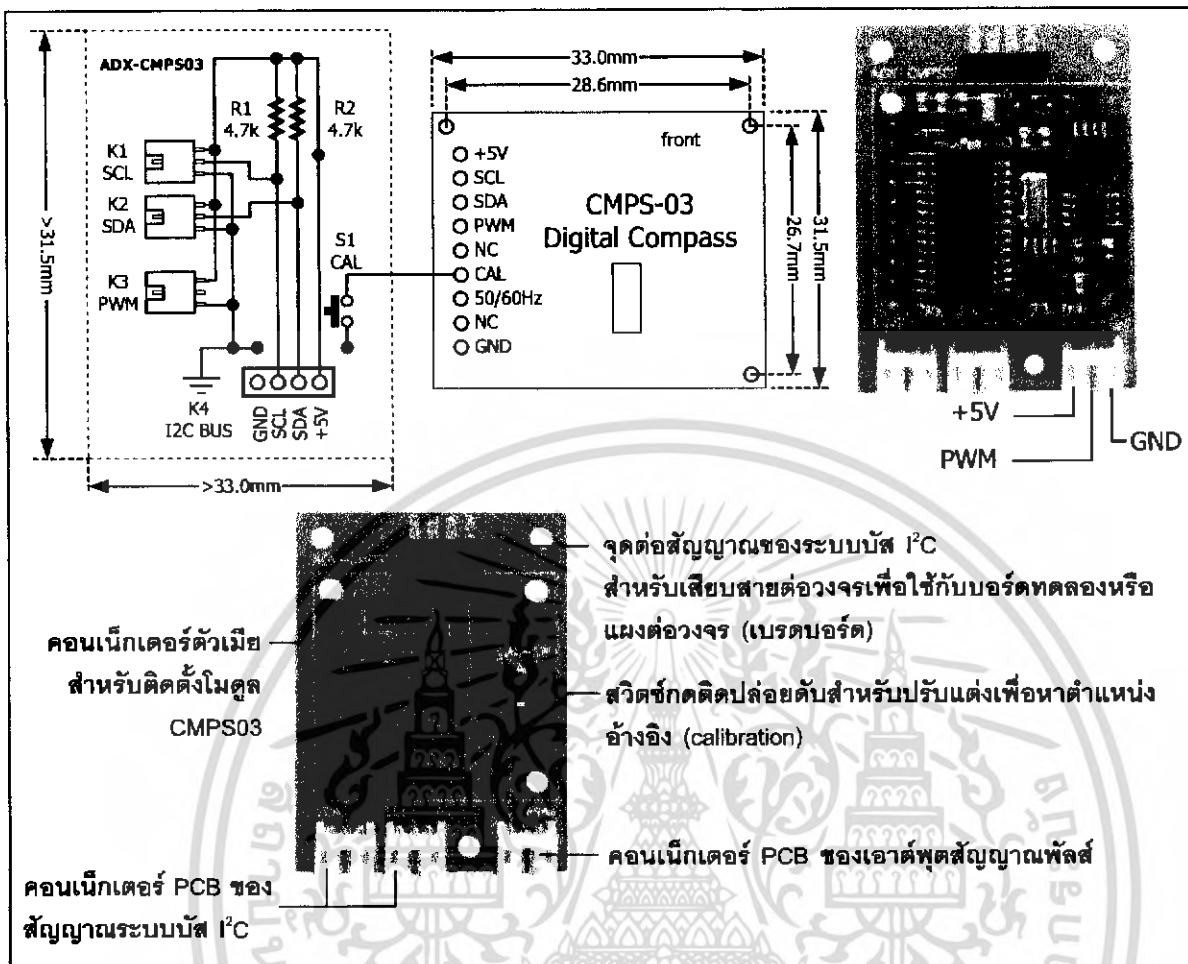
1. ตำแหน่งขาและการต่อใช้งาน

ในรูปที่ 1 แสดงรูปร่างหน้าตาและการจัดขาของ CMPS03 โมดูลเซ็นเซอร์ทิศทาง จะเห็นว่าเป็นแผงวงจรที่มีคอนเนกเตอร์ต่อออกมาเพื่อให้เชื่อมต่อไปใช้งาน อย่างไรก็ตามเพื่ออำนวยความสะดวกแก่ผู้ใช้งานกับบอร์ดควบคุมหุ่นยนต์ของบริษัท อินโนเวตีฟ อิเล็กทรอนิกส์ จำกัด (i-nex : เป็นตัวแทนจำหน่ายสินค้าของ Devantech ในประเทศไทยอย่างเป็นทางการ) จึงได้พัฒนาบอร์ดอะแดปเตอร์รุ่น ADX-CMPS03 เพื่อให้นำโมดูล CMPS03 มาติดตั้ง (โดยบอร์ด ADX-CMPS03 ต้องจัดซื้อแยก)

บนบอร์ด ADX-CMPS03 ได้จัดเตรียมคอนเนกเตอร์ PCB 3 ขาตัวสำหรับเชื่อมต่อกับบอร์ดควบคุมหุ่นยนต์ และคอนเนกเตอร์ IDC ตัวเมียแถวเดียว 4 ขาสำหรับเสียบสายต่อวงจรเบอร์ AWG#22 เพื่อต่อกับแผงต่อวงจรหรือบอร์ด นอกจากนี้ยังมีสวิตช์กดสำหรับปรับตั้งค่า (calibration) เพื่อกำหนดตำแหน่งทิศอ้างอิง โดยวงจรของบอร์ด ADX-CMPS03 แสดงในรูปที่ 2



รูปที่ 1 แสดงรูปร่างและตำแหน่งขาสำหรับการต่อใช้งาน



รูปที่ 2 แสดงวงจรของบอร์ด ADX-CMPS03 และการเชื่อมต่อกับโมดูล CMPS03

2. การปรับแต่งค่าทิศทางอ้างอิงแก้มอดูล CMPS03

เพื่อให้การวัดทิศทางของโมดูล CMPS03 มีความแม่นยำมากที่สุด จึงมีอินพุตสำหรับปรับแต่งค่าทิศทางอ้างอิง ทั้งนี้เพื่อประโยชน์ในการกำหนดทิศทางอ้างอิงเฉพาะสำหรับผู้ใช้งาน โดยต้องป้อนสัญญาณลอจิก “0” เข้าที่ขาอินพุตสำหรับปรับแต่งโมดูล CMPS03 ซึ่งก็คือขา 6 หากใช้บอร์ด ADX-CMPS03 กับโมดูล CMPS03 จะมีสวิตช์กดติดปล่อยดับติดตั้งไว้ให้แล้ว การปรับแต่งมีขั้นตอนดังนี้

- (1) วางโมดูล CMPS03 ขนานกับพื้น หันด้านหน้าของโมดูลไปทางทิศเหนือ กดสวิตช์ 1 ครั้ง
- (2) วางโมดูล CMPS03 ขนานกับพื้น หันด้านหน้าของโมดูลไปทางทิศตะวันออก กดสวิตช์
- (3) วางโมดูล CMPS03 ขนานกับพื้น หันด้านหน้าของโมดูลไปทางทิศใต้ กดสวิตช์ 1 ครั้ง
- (4) วางโมดูล CMPS03 ขนานกับพื้น หันด้านหน้าของโมดูลไปทางทิศตะวันตก กดสวิตช์

เป็นอันสิ้นสุดการปรับตั้งค่าทิศทางอ้างอิงของโมดูล CMPS03 โดยโมดูลจะเก็บค่าอ้างอิงนี้ไว้ในหน่วยความจำอีอีพรอมและไม่ต้องปรับตั้งค่าใหม่อีกเมื่อจ่ายไฟเลี้ยงครั้งใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. การอ่านค่าสัญญาณเอาต์พุตของโมดูล CMPS03

3.1 การอ่านค่าทิศทางจากเอาต์พุตสัญญาณพัลส์

การอ่านค่าสัญญาณในโหมดนี้ เป็นการนำค่าความกว้างพัลส์ที่ได้จากเอาต์พุตสัญญาณพัลส์ของโมดูล CMPS03 มาระบุตำแหน่งองศา จาก 0 ถึง 359.9 องศา โดยมีย่านของค่าความกว้างสัญญาณพัลส์จาก 1 มิลลิวินาทีไปจนถึง 36.99 มิลลิวินาที มีความละเอียด 0.1 มิลลิวินาทีต่อองศา ในสัญญาณพัลส์แต่ละไซเคิล มีช่วงลอคิก “0” กว้าง 65 มิลลิวินาที

ดังนั้นในการนำสัญญาณพัลส์มาประมวลผลเป็นค่ามุม จึงต้องใช้การนับความกว้างของสัญญาณพัลส์เป็นหลักในการคำนวณหาค่ามุมที่โมดูล CMPS03 วัดได้

3.1.1 ตัวอย่างโปรแกรมที่ใช้ควบคุมสำหรับเบสิกสเตมปี 2SX และ i-Stamp

การใช้งานร่วมกับเบสิกสเตมปี 2SX และ i-Stamp นั้น จะใช้คำสั่ง PULSIN ในการนับสัญญาณพัลส์ โดยจะเพิ่มค่าการนับขึ้นทุกๆ 0.8 ไมโครวินาที ดังนั้นที่ความกว้างของพัลส์ที่ 1 มิลลิวินาทีสำหรับตำแหน่ง 0 องศา เบสิกสเตมปี 2SX และ i-Stamp จะนับค่าได้เท่ากับ 1,250 จึงสามารถใช้ค่านี้เป็นจุดอ้างอิงที่ 0 องศา เมื่อต้องการทราบค่ามุมที่แท้จริงให้นำค่ามุมที่นับได้ลบด้วย 1,250 แล้วหารด้วย 125 ก็จะได้ค่ามุมในหน่วยองศาที่ต้องการ รายละเอียดของโปรแกรมแสดงในโปรแกรมที่ 1

ที่ความกว้างพัลส์สูงสุดคือ 36.99 มิลลิวินาที ค่าที่นับได้จากคำสั่ง PULSIN เท่ากับ 46,237 เมื่อลบด้วย 1,250 แล้วหารด้วย 125 เพื่อแปลงเป็นองศา ค่าสูงสุดที่แสดงเป็นผลลัพธ์ได้คือ 359 เป็นค่าหน่วยองศาสูงสุดนั่นเอง

หมายเหตุ ในบางกรณี ค่าที่อ่านได้สูงสุดจากคำสั่ง PULSIN อาจไม่ถึง 46,237 ผู้ใช้งานสามารถปรับเปลี่ยนการคำนวณใหม่ให้ได้มุมเป็น 359.9 องศาได้

```
{
  '$STAMP BS2sx}
  '$PBASIC 2.5}
bearing VAR WORD
main:
  PULSIN 4, 1, bearing          ' Get reading
  bearing = (bearing-1250)/125  ' BS2sx - Calculate Bearing
                                ' in degrees
  DEBUG "Compass Bearing ", DEC3 bearing ,CR
                                ' Display Compass Bearing
  GOTO main
}
```

โปรแกรมที่ 1 แสดงการอ่านค่าสัญญาณพัลส์จากโมดูล CMPS03

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| ตำแหน่งรีจิสเตอร์ | รายละเอียด |
|-------------------|--|
| 0 | ตัวเลขแสดงรุ่นของบอร์ด CMP03 |
| 1 | ส่งค่าตำแหน่งแบบหยาบ (0-255) |
| 2,3 | ส่งค่าตำแหน่งแบบละเอียดด้วยตัวเลข 16 บิต (0-3599) สามารถแปลงค่าเพื่อแสดงองศา 0-359.9 องศาได้โดยตรง |
| 4,5 | สำหรับตรวจสอบค่าภายใน โดยจะแสดงค่าความต่างของ Sensor1 เป็นตัวเลข 16 บิตแบบคิดเครื่องหมาย |
| 6,7 | สำหรับตรวจสอบค่าภายใน โดยจะแสดงค่าความต่างของ Sensor2 เป็นตัวเลข 16 บิตแบบคิดเครื่องหมาย |
| 8,9 | แสดงค่าตัวเลขการปรับแต่งภายใน (calibration value1) เป็นตัวเลข 16 บิตแบบคิดเครื่องหมาย |
| 10,11 | แสดงค่าตัวเลขการปรับแต่งภายใน (calibration value2) เป็นตัวเลข 16 บิตแบบคิดเครื่องหมาย |
| 12,13 | ไม่ใช้งานอ่านค่าได้เป็น 0 |
| 14 | ไม่ใช้งาน ไม่ได้กำหนดค่าไว้ |
| 15 | คำสั่งสำหรับการปรับแต่งค่า โดยเมื่อต้องการปรับแต่งค่า ต้องเขียนข้อมูล 255 เข้าที่รีจิสเตอร์ตำแหน่งนี้ |

ตารางที่ 1 แสดงตำแหน่งรีจิสเตอร์ภายในโมดูล CMPS03

3.2 การอ่านค่าทิศทางเป็นข้อมูลดิจิทัลผ่านระบบบัส I²C

การอ่านค่าจาก โมดูล CMPS03 ให้ได้ค่าที่มีความแม่นยำสูงควรเลือกเอาต์พุตข้อมูลดิจิทัลผ่านระบบบัส I²C โดยโมดูล CMPS03 สามารถส่งข้อมูลของตำแหน่งออกมาที่ความละเอียดสูงสุด 0.1 องศาโดยไม่จำเป็นต้องมีการคำนวณหรือแปลงค่าใด ๆ อีก

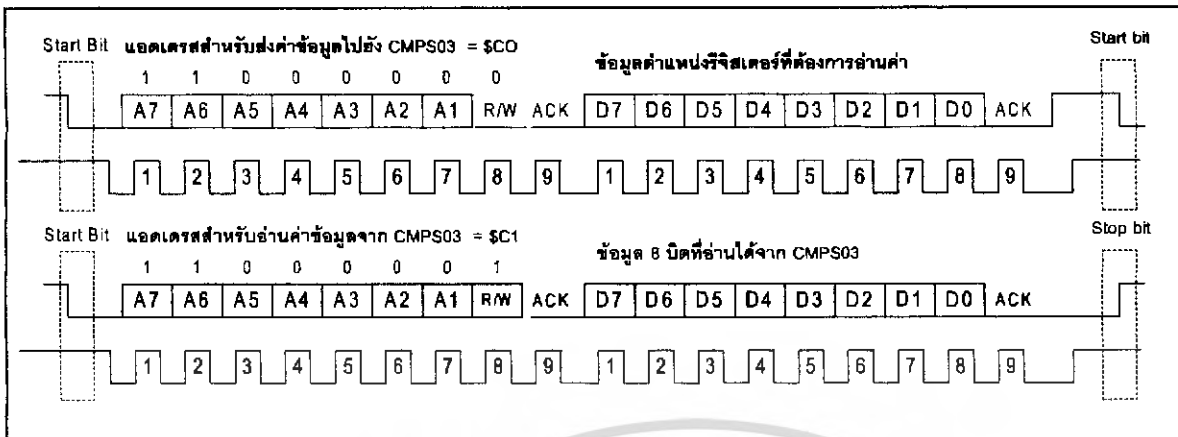
3.2.1 รูปแบบการสื่อสารข้อมูลบัส I²C

บัส I²C เชื่อมต่อกับไมโครคอนโทรลเลอร์โดยใช้สายสัญญาณ 2 เส้น ได้แก่ขา SDA (รับและส่งข้อมูล) และ SCL (ขาสัญญาณนาฬิกา) โดยขาสัญญาณทั้งสองจะต้องต่อตัวต้านทานพูลอัพต่อไว้เพื่อกำหนดสถานะลอจิก “1” ให้กับระบบบัส

3.2.2 ลำดับขั้นการติดต่อ

ค่าแอดเดรสของโมดูล CMPS03 คือ SC0 สำหรับการส่งข้อมูล และ SC1 สำหรับการอ่านค่าข้อมูล โดยขั้นตอนการติดต่อกับโมดูล CMPS03 เพื่ออ่านข้อมูลมีดังนี้

6 • CMPS03 โมดูลเซ็นเซอร์ดิจิทัล



รูปที่ 3 แสดงไทมิงไคอะแกรมของการติดต่อสื่อสารกับโมดูล CMPS03 ผ่านระบบบัส I²C

1. ส่งบิตเริ่มต้นหรือ Start bit เพื่อแจ้งให้ระบบบัส I²C เตรียมพร้อมรับข้อมูล
 2. ส่งค่าแอดเดรส \$C0 เพื่อระบุว่าต้องการติดต่อเพื่อเขียนข้อมูลไปยังกับ โมดูล CMPS03
 3. ส่งค่าตำแหน่งรีจิสเตอร์ภายในโมดูล CMPS03 ที่ต้องการอ่านค่า ซึ่งมีรายละเอียดแสดงในตารางที่ 1
 4. ส่งค่าแอดเดรส \$C1 เพื่อระบุว่าต้องการอ่านค่าข้อมูลจากโมดูล CMPS03
 5. อ่านค่าข้อมูลจาก โมดูล CMPS03 มาเก็บไว้ในหน่วยความจำ
 6. ส่งบิตหยุดหรือ Stop เพื่อหยุดการสื่อสารข้อมูล และกำหนดให้บัสอยู่ในสภาวะบัสว่าง
- จากลำดับขั้นการติดต่อสื่อสารข้างต้น สามารถนำมาเขียนเป็น โปรแกรมตัวอย่างเพื่ออ่านข้อมูลจากโมดูล CMPS03 โดยใช้เบสิกแพลตฟอร์ม 2SX หรือ i-Stamp ได้ดังแสดงในโปรแกรมที่ 2

```

'{$STAMP BS2sx}
'{$PBASIC 2.5}
SDA          CON      6          ' I2C serial data line
SCL          CON      7          ' I2C serial clock line
WrCMPS03    CON      $C0        ' write to compass
RdCMPS03    CON      $C1        ' read from compass
Ack         CON      0          ' acknowledge bit
Nak         CON      1          ' no ack bit
i2cSDA      VAR      NIB        ' I2C serial data pin
i2cData     VAR      WORD       ' data to/from device
REGISTER    VAR      BYTE       ' register address
i2cWork     VAR      BYTE       ' work byte for TX routine
i2cAck      VAR      BIT        ' Ack bit from device
temp       VAR      WORD       ' for rj printing
digits     VAR      NIB
width      VAR      NIB

Init:
  PAUSE 250
  i2cSDA = SDA          ' define SDA pin
  REGISTER = 0         ' compass revision number
  GOSUB Read_Byte
  DEBUG 2,1,1, "Revision Number = " ,DEC2 i2cData

Main:
  REGISTER = 1         ' Show Data 0-255 for 0-360 degree
  GOSUB Read_Byte     ' Read Byte From I2C
  DEBUG 2,1,3, "The Coarse Data(0-255) = " ,DEC i2cData
  REGISTER = 2         ' get Data in degrees, 0.0 - 359.9 Degree
  GOSUB Read_Word
  DEBUG 2,1,5, "Position = " ,DEC i2cData/10,".",DEC1 i2cData," Degree"
  PAUSE 250
  GOTO Main

' Compass Access Subroutines
' Writes low byte of i2cData to REGISTER
Write_Byte:
  GOSUB I2C_Start
  i2cWork = WrCMPS03
  GOSUB I2C_TX_Byte   ' send device address
  i2cWork = REGISTER
  GOSUB I2C_TX_Byte   ' send register number
  i2cWork = i2cData.LOWBYTE
  GOSUB I2C_TX_Byte   ' send the data
  GOSUB I2C_Stop
  RETURN

' Writes i2cData to REGISTER
Write_Word:
  GOSUB I2C_Start
  i2cWork = WrCMPS03
  GOSUB I2C_TX_Byte   ' send device address
  i2cWork = REGISTER
  GOSUB I2C_TX_Byte   ' send register number
  i2cWork = i2cData.HIGHBYTE
  GOSUB I2C_TX_Byte   ' send the data - high byte
  i2cWork = i2cData.LOWBYTE
  GOSUB I2C_TX_Byte   ' send the data - low byte
  GOSUB I2C_Stop
  RETURN

```

โปรแกรมที่ 2 โปรแกรมอ่านค่าจากโมดูล CMPS03 ผ่านระบบบัส I²C ของ I-Stamp (มีต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

' Read i2cData (8 bits) from REGISTER
Read_Byte: GOSUB I2C_Start
            i2cWork = WrCMPS03
            GOSUB I2C_TX_Byte           ' send compass address
            i2cWork = REGISTER
            GOSUB I2C_TX_Byte           ' send register number
            GOSUB I2C_Start             ' repeat start (sets register)
            i2cWork = RDCMPS03
            GOSUB I2C_TX_Byte           ' send read command
            GOSUB I2C_RX_Byte_Nak
            GOSUB I2C_Stop
            i2cData = i2cWork           ' return the data
            RETURN

' Read i2cData (16 bits) from REGISTER
Read_Word: GOSUB I2C_Start
            i2cWork = WrCMPS03
            GOSUB I2C_TX_Byte           ' send compass address
            i2cWork = REGISTER
            GOSUB I2C_TX_Byte           ' send register number
            GOSUB I2C_Start             ' repeat start (sets register)
            i2cWork = RDCMPS03
            GOSUB I2C_TX_Byte           ' send read command
            GOSUB I2C_RX_Byte
            i2cData.HIGHBYTE = i2cWork   ' read high byte of data
            GOSUB I2C_RX_Byte_Nak
            GOSUB I2C_Stop
            i2cData.LOWBYTE = i2cWork   ' read low byte of data
            RETURN

' Low Level I2C Subroutines
'Start
I2C_Start: INPUT i2cSDA                 ' I2C start bit sequence
            INPUT SCL
            LOW i2cSDA                   ' SDA -> low while SCL high
Clock_Hold:
            IF (INS.LOWBIT(SCL) = 0) THEN Clock_Hold ' device ready?
            RETURN

'Transmit
I2C_TX_Byte:
            SHIFTOUT i2cSDA,SCL,MSBFIRST,[i2cWork\8] ' send byte to device
            SHIF TIN i2cSDA,SCL,MSBP RE,[i2cAck\1]   ' get acknowledge bit
            RETURN

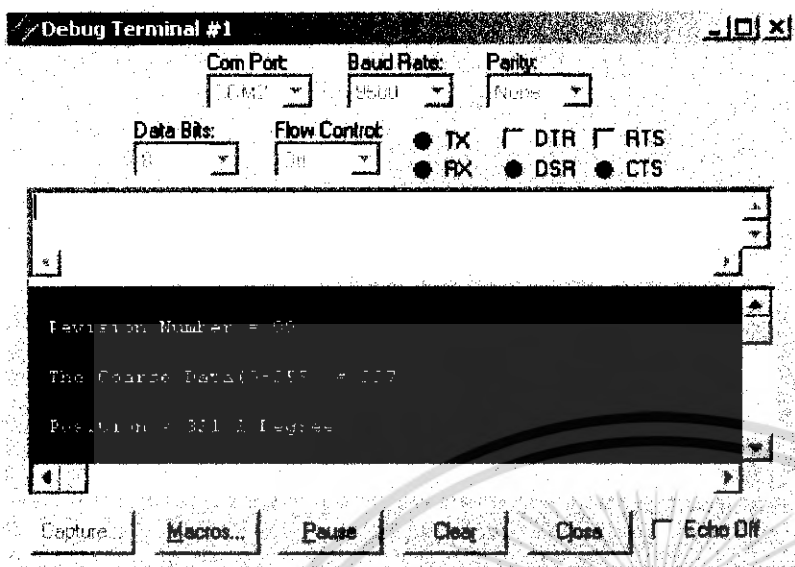
'Receive
I2C_RX_Byte_Nak: i2cAck = Nak           ' no Ack = high
                GOTO I2C_RX
I2C_RX_Byte:    i2cAck = Ack            ' Ack = low
I2C_RX:
            SHIF TIN i2cSDA,SCL,MSBP RE,[i2cWork\8] ' get byte from device
            SHIF TOU T i2cSDA,SCL,LSBFIRST,[i2cAck\1] ' send ack or nak
            RETURN

'Stop
I2C_Stop:     LOW i2cSDA                 ' I2C stop bit sequence
            INPUT SCL
            INPUT i2cSDA                 ' SDA --> high while SCL high
            RETURN

```

โปรแกรมที่ 2 โปรแกรมอ่านค่าจากโมดูล CMPS03 ผ่านระบบบัส I²C ของ I-Stamp (มีต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4 แสดงหน้าจอของ Debug terminal แสดงข้อมูลที่อ่านได้จากโมดูล CMPS03

3.2.3 การทำงานของโปรแกรมที่ 2 ติดต่อกับโมดูล CMPS03 ผ่านระบบบัส I²C ด้วย i-Stamp

เนื่องจากเบสิกแอสตมป์ 2SX และ i-Stamp ไม่มีคำสั่งติดต่อกับระบบบัส I²C ดังนั้น โปรแกรมติดต่อกับ โมดูล CMPS03 ผ่านระบบบัส I²C จึงค่อนข้างยาว เนื่องจากต้องสร้าง โปรแกรมย่อยสำหรับการสื่อสารข้อมูลกับระบบบัส I²C หลังจากนั้นผู้ใช้งานสามารถนำโปรแกรมย่อยนี้ไปประยุกต์ใช้งานกับอุปกรณ์ตัวอื่นๆ ที่ได้ใช้การติดต่อบนระบบบัส I²C ได้ทันที สำหรับขั้นตอนการทำงานหลักๆ ของโปรแกรมที่ 2 มีดังนี้

1. กำหนดรีจิสเตอร์เป็น 0 เพื่อติดต่อบรรทัดแรกของโมดูล CMPS03 จากนั้นอ่านค่ามาแสดงที่หน้าต่าง Debug terminal
2. ให้โปรแกรมวนลูปที่โปรแกรมหลัก จากนั้นส่งค่ารีจิสเตอร์เท่ากับ 1 เพื่ออ่านค่าข้อมูลแบบขยายออกมา แล้วนำมาแสดงผลที่หน้าต่าง Debug terminal
3. ส่งค่ารีจิสเตอร์เท่ากับ 2 เพื่ออ่านค่าข้อมูลแบบละเอียด จากนั้นส่งอ่านค่าข้อมูลจากบัส I²C แบบเวิร์ด (อ่านข้อมูลออกมา 16 บิต)
4. นำค่าข้อมูลที่ได้อ่านด้วย 10 ก่อนเพื่อแปลงค่าที่ได้เป็นองศา แสดงที่หน้าต่าง Debug terminal จากนั้นนำค่าหลักสุดท้ายมาแสดง ซึ่งตำแหน่งสุดท้ายเป็นตำแหน่งของจุดทศนิยม

3.3.4 การปรับแต่งค่าของโมดูล CMPS03 ผ่านทางระบบบัส I²C

การปรับแต่งค่าทำได้โดยการส่งค่า 0xFF ไปยังรีจิสเตอร์ 15 ของโมดูล CMPS03 โดยจะต้องส่งค่า 4 ครั้งและระบุทิศทางหลักๆ 4 ทิศทางเช่นเดียวกับการกำหนดค่าด้วยสวิตช์โดยตรง มีขั้นตอนดังนี้


1. วางโมดูล CMPS03 ขนานกับพื้น หันด้านหน้าของโมดูลไปทางทิศเหนือ จากนั้นเขียนค่า 255 (0xFF) ไปยังรีจิสเตอร์ 15
2. วางโมดูล CMPS03 ขนานกับพื้น หันด้านหน้าของโมดูลไปทางทิศตะวันออก เขียนค่า 255 (0xFF) ไปยังรีจิสเตอร์ 15
3. วางโมดูล CMPS03 ขนานกับพื้น หันด้านหน้าของโมดูลไปทางทิศใต้ กดสวิตช์ เขียนค่า 255 (0xFF) ไปยังรีจิสเตอร์ 15
4. วางโมดูล CMPS03 ขนานกับพื้น หันด้านหน้าของโมดูลไปทางทิศตะวันตก กดสวิตช์ เขียนค่า 255 (0xFF) ไปยังรีจิสเตอร์ 15

หลังจากปรับแต่งค่าแล้วค่าที่ปรับแต่งจะเก็บไว้ที่หน่วยความจำอีอีพรอม ดังนั้นแม้ไม่จ่ายไฟให้กับตัวบอร์ด ข้อมูลที่ปรับแต่งแล้ว จะยังคงอยู่ต่อไป



RDL401A Series

Information

| Machine type | Function | | | | |
|--|--|-------------------------------|-------------------------|---------------------------------|-----------------------------------|
|  RDL401A | <table border="1"><tr><td>AUTO SWITCH TX ↔ RX</td><td>DISPLAY ● ● ●</td></tr><tr><td>RS232 ● ● ● ● ● ● ● ●</td><td>CHANNEL SELECT [Slider]</td></tr></table> | AUTO SWITCH TX ↔ RX | DISPLAY ● ● ● | RS232 ● ● ● ● ● ● ● ● | CHANNEL SELECT [Slider] |
| AUTO SWITCH TX ↔ RX | DISPLAY ● ● ● | | | | |
| RS232 ● ● ● ● ● ● ● ● | CHANNEL SELECT [Slider] | | | | |

■ Standard Package



■ Optional



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

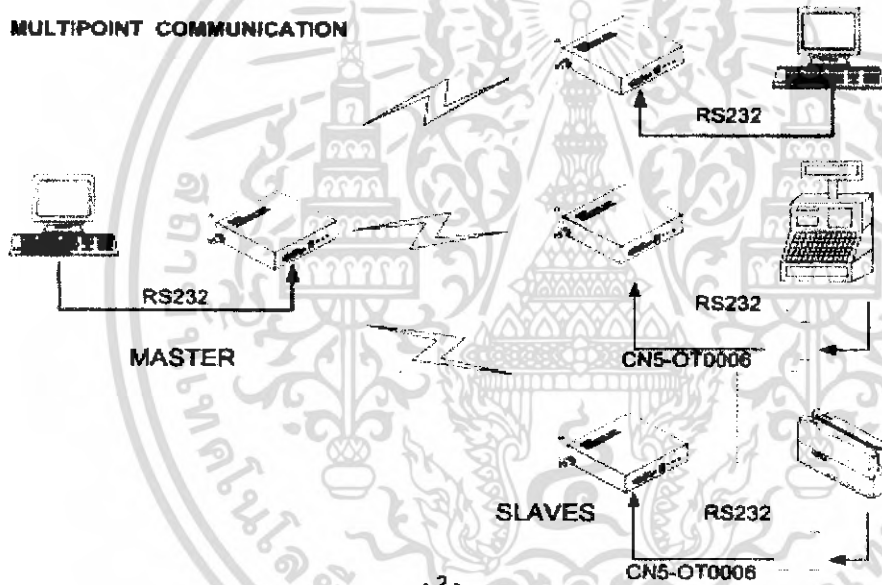
Application

- To replace the RS-232 cable between Computer and Computer (Terminal) .
- Remote control
- Remote monitoring
- Data acquisition
- Computer aided integrated manufacturing
- Point-of-sale system
- Data transmission of movable station

POINT TO POINT COMMUNICATION

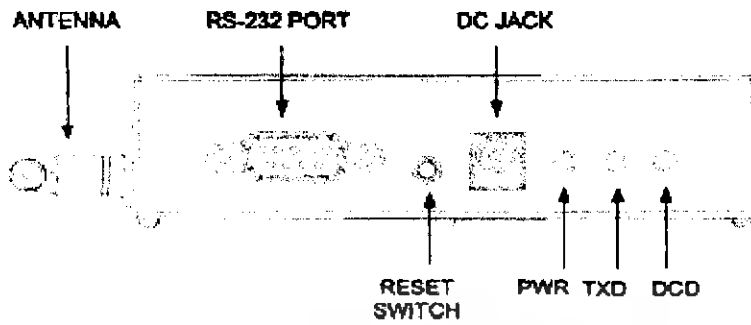


MULTIPOINT COMMUNICATION



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Configuration



CHANNEL Select Switch :

- Switch 1 : ON : 433.92MHz
OFF : 434.33MHz
- Switch 2 : ON : 4800 BPS Disable
OFF : 4800 BPS Enable
- Switch 3 : ON : 9600 BPS Enable
OFF : 19200 BPS Enable
- Switch 4 : Reserved

RS-232 Port -- To be connected with the Computer (Terminal) .

REST Switch -- Press this REST switch to reset radio Channel to the new setting .
Before pressing this switch , the radio Channel will not change although the CHANNEL Select Switch has been changed . (The CHANNEL Select Switch is located on the bottom of the RDL401A unit)

Power Jack -- For DC 9V power input .

PWR Power indicator -- Turn on when the power is applied .

TXD Indicator -- Green LED turns on when the data is sent from the connected Computer (Terminal) to RDL401A unit.

DCD Indicator -- Green LED turns on when it detects a radio transmission from the air (Data Carrier Detect) .

CHANNEL Select Switch -- DIP switch located on the bottom of the RDL401A unit is for changing radio channel 433.92MHz or 434.33MHz. After you make Channel change , you must press RESET switch or power off and then reboot power . Then the RDL401A will be reset to the new radio Channel .

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Installation

1. Install the RS-232 cable between the RDL401A and Computer or Terminal .
(The pin connection is as PIN ASSIGNMENT)
2. Set up one pair of RDL401 , one for Master Computer (terminal) and the other for Remote Computer (terminal) . Set both RDL401A units at the same channel .
3. Connect the DC power supply to the DC Power Jack on RDL401A unit , the PWR LED will turn on.
4. The DCD LED must light green constantly for the receiving unit when detecting a radio transmission from the other unit . If not ,reinstall them to have a constant green DCD light for the receiving unit .
Please note that you must not have both units become transmitting at the same time because RDL401A is for half-duplex communication only . At one time you must have one unit become transmitting and the other unit become receiving .

Note :

To get connection with RDL401A , you must set Computer (terminal) as the following parameters under the operation software you are using :

Standard : Standard RS-232
Baud Rate : 4800/9600/19200 bps
Parity : None
Data Bits : 8
Stop BIT : 1
Flow control : None

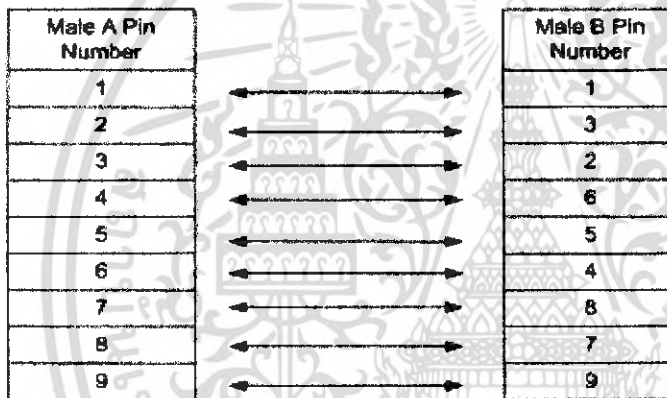
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PIN ASSIGNMENT

9 Pin D Female Connect

| Pin Number | Signal |
|------------|-------------|
| 1 | X |
| 2 | TXD (Out) |
| 3 | RXD (In) |
| 4 | X |
| 5 | Ground |
| 6 | X |
| 7 | X |
| 8 | RTS (Out) |
| 9 | X |

CN5-OTT0008 Connect



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Specification

Frequency : 433.92/434.33MHz

Antenna Impedance : 50 Ohms , unbalanced

Number of Channels : 2 (Frequency Synthesized)

Frequency Stability : +/- 10 ppm (at 0-60 Deg. C)

Data Rate : 4800/9600/19200 Baud

Data format : None Parity , 8 Data Bits , 1 Stop Bit ,
Half-duplex over a single channel Asynchronous
, Serial

Communication Distance : Approx. 50-150 meters (prospective)

Power Supply : 9 VDC

Environment : Operating Temperature 0 ~+60 Deg. C
Storage Temperature -10 ~+70 Deg. C
Humidity : 10 % - 95 % , non-condensing

Dimensions : W 107 * D 102 * H 28 mm

Weight : APPROX. 0.38 KGs

Transmitter Output Power : 5mW maximum

Modulation Deviation : 35 +/- 5 KHz

PLL Programming Time : 5 m sec , maximum

Current Drain : 60 mA maximum at 9 VDC nominal

Sensitivity : - 102 dBm

Selectivity : 85 KHz maximum

Receiver Current : Typical 27 mA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

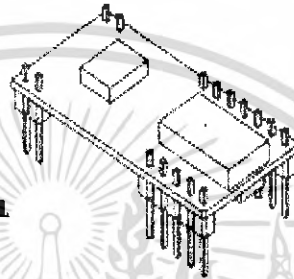
Radio Data Link RF Module with UART Interface

INTRODUCTIONS:

RDL200T (Radio Data Link) RF Module is designed for wireless RS232 serial data communication. It is suitable for most of your wireless data link applications. RDL200T RF Module can be built into your equipment through RS232 interface to replace the cable to become wireless communication. It is a low cost and high quality radio data communication solution.

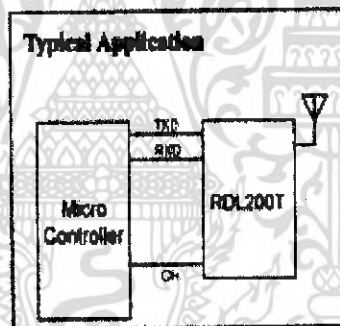
FEATURES:

1. ISM frequency 433.92MHz / 434.33MHz
2. Two channel selectable.
3. Low power consumption with 3.3Vdc
4. RS232 interface (Serial TTL level)
5. Speed up to 9600bps with automatic flow control.
6. Half-duplex with automatic Receiving/Transmitting control.
7. Small package size.
8. Distance: 50-100 meters with appropriate antenna



APPLICATION:

1. Hand-held terminals
2. Computer to computer
3. Computer to terminal
4. Data collection terminal
5. Mobile-around data communications
6. Remote control
7. Remote monitoring
8. POS systems.



SPECIFICATIONS:

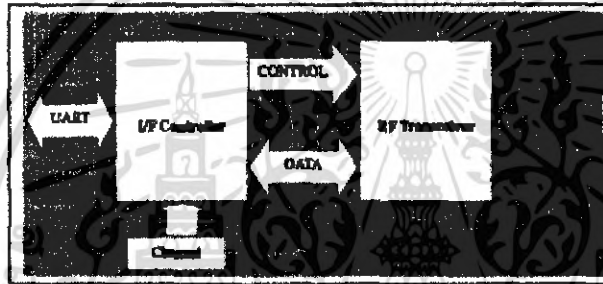
1. Frequency: CH1 433.92MHz / CH2 434.33MHz
2. Modulation: FSK
3. Output power: 7dbm (5mw)
4. Receiving Sensibility: -102dbm
5. Operating voltages: 2.7Vdc-3.3Vdc
6. Power consumption: Tx:60mA / Rx:30mA
7. Interface : RS232 (TTL Level: TXD/RXD)
8. Data Rate : 19200bps/9600bps / 4800bps , Half-duplex
9. Operating temperature: 0°C-55°C (Storage Temperature: -10°C -60°C)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

QUICK REFERENCE DATA

| | | |
|--------------------------------|-----------------|-----|
| Frequency, Channel#1/Channel#2 | 433.92/434.33 | MHz |
| Modulation | FSK | |
| Frequency deviation | +/-30 | KHz |
| Max. RF output power | 7 | dBm |
| Sensitivity | -102 | dBm |
| UART Baudrate | 19200/9600/4800 | bps |
| Supply voltage | 2.7 - 3.3 | Vdc |
| Max. Receive current | 30 | mA |
| Max. Transmit current | 60 | mA |

BLOCK DIAGRAM



I/F Controller: UART and Radio interface unit

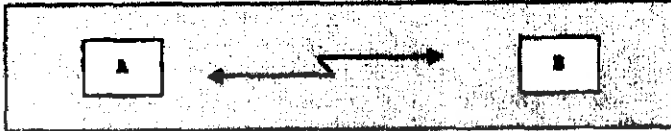
RF Transceiver: A long-range radio transceiver for wireless links operating in the globally available ISM band.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RDL200T APPLICATION

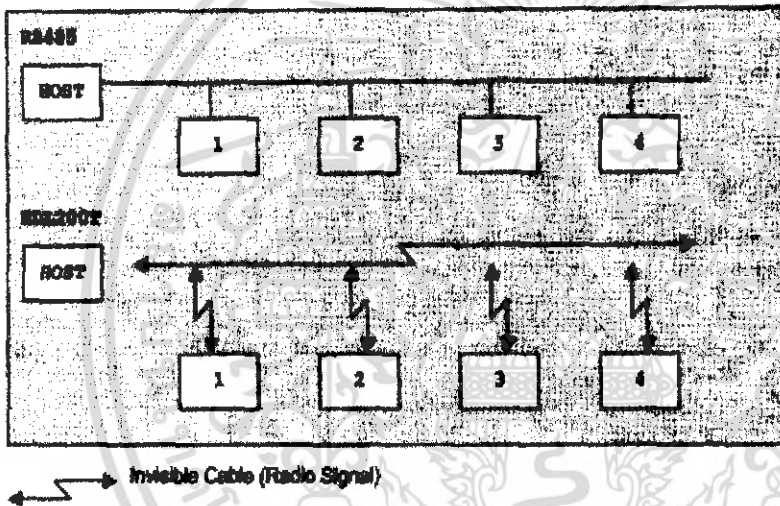
1. Point to Point Radio Data Link

For point to point radio data link, just change your RS232 transceiver IC to RDL200T in your RS232 product.
You don't need to change any firmwares, RDL200T serves as a invisible RS232 cable in your product.



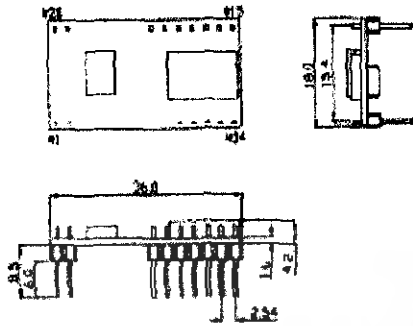
2. Multi-Station Radio Data Link

For Multi-Station radio data link, you have to build a protocol for Multi-Station that's similar to RS485 to assign machine ID for each station.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PACKAGE OUTLINE (DIP28)



Unit: mm

PIN ASSIGNMENT

| PIN | NAME | I/O | FUNCTION |
|----------------|---------|-----|--|
| 1 | ANT | I | *Antenna |
| 2 | ANTG | I | *Antenna Signal GND |
| 3,4,5,6,7,8,9 | NC | | No Connect |
| 10 | /RST | I | Active Low Reset |
| 11 | RXD | I | UART Data Input |
| 12 | TXD | O | UART Data Output |
| 13 | GND | I | GND |
| 14 | TRD/CMD | | Reserved |
| 15 | BPS0 | | 4800bps 0:disable 1:enable |
| 16 | BPS1 | | 0:9600bps 1:19200bps(*short data pack) |
| 17 | CH | I | 0:433.92MHz 1:434.33MHz |
| 18 | RLED | O | RXD Status LED |
| 19 | TLED | O | TXD Status LED |
| 20 | GND | | GND |
| 21 | VCC | | Power Input +3.3Vdc-2.7Vdc |
| 22,23,24,25,26 | NC | | No Connect |
| 27 | GND | | GND |
| 28 | GND | | GND |

*Antenna Signal GND: Connect to antenna cable GND.

*Antenna: 50 ohm / 430-435MHz / -25db.

*19200bps for short data pack (max 70 bytes) only.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROMAG

RDL200T
REV.B

RDL200T START KIT

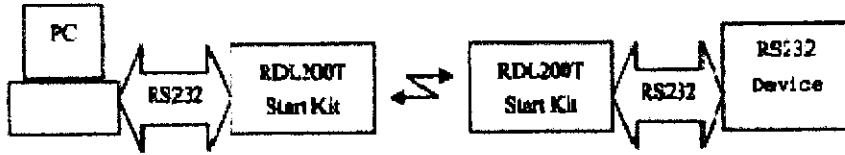
The RDL200TSK Start Kit includes the RDL200T module and RS232 Transceiver IC. You can easily learn and develop your product for wireless communication application.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

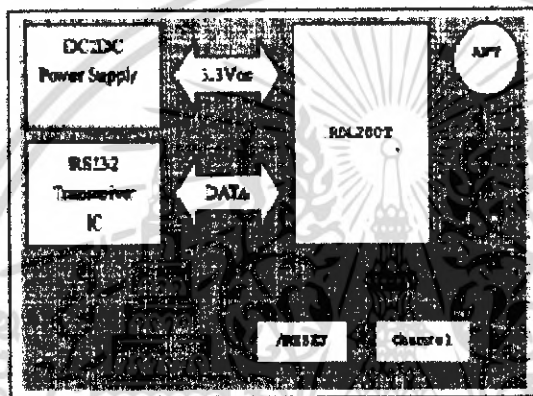
START KIT PACKAGE APPLICATION

Example: Point to Point radio data link

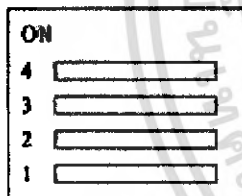


Keep your communication way and easy to upgrade your product to become wireless product with RDL200T

ASY-REL200TSK BLOCK DIAGRAM



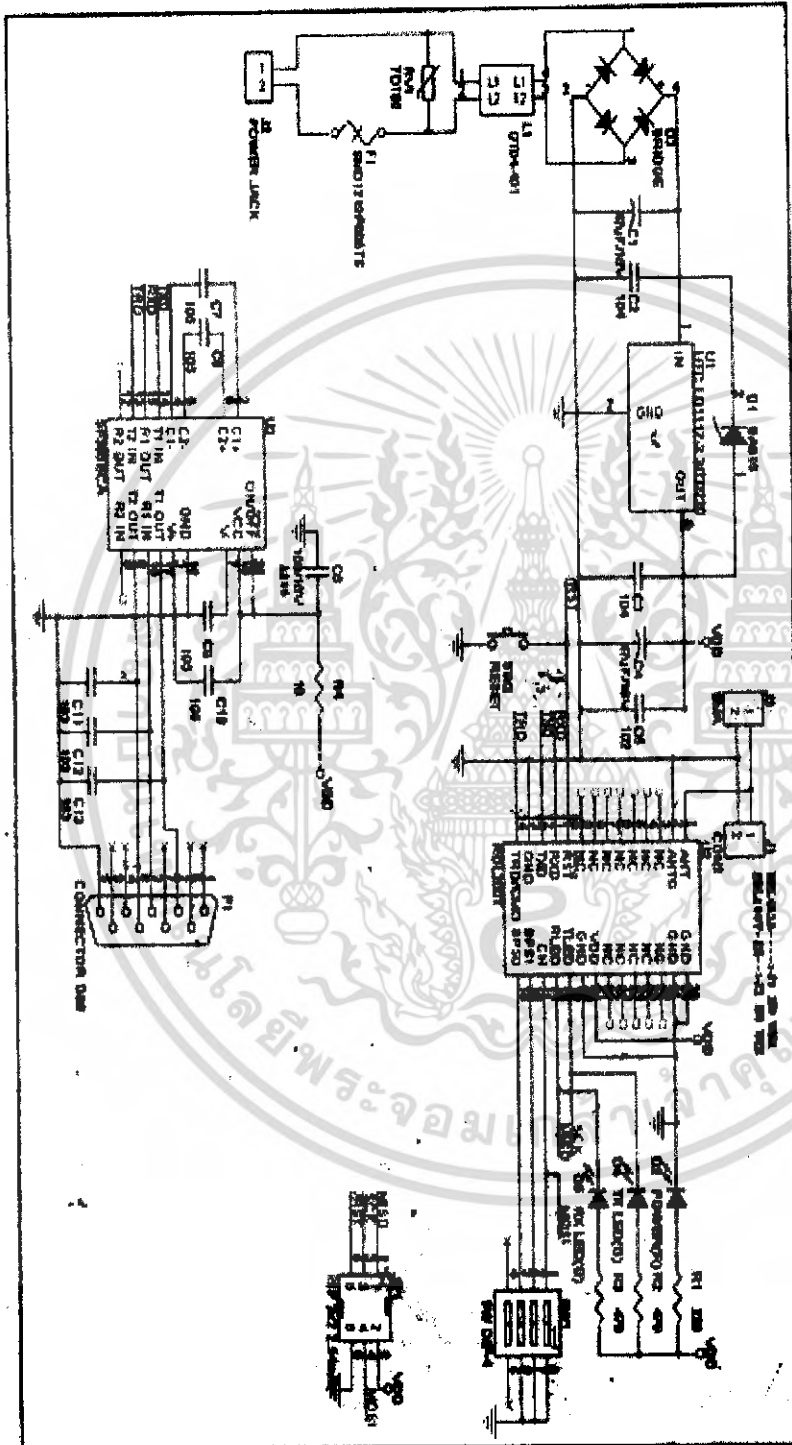
DIP SWITCH



1. ON : CH1 (433.92MHz) / OFF: CH2 (434.39MHz)
2. ON : 4800 bps DISABLE / OFF : 4800 bps ENABLE
3. ON : 9600 bps ENABLE / OFF : 19200 bps ENABLE
4. Reserved

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ASY-RDL200TSK CIRCUIT DIAGRAM



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ASY-RDL200TSK BILL OF MATERIALS

| Part Number | Description | Location | QTY |
|----------------|-------------------------------|--------------|-----|
| UTCL01117-3.3J | 3.3Vdc DC2DC Power supply | U1 | 1 |
| 7F01-027-00 | Audio Line In/Out Module | U2 | 1 |
| SP385ACA | RS232 Transceiver IC | U3 | 1 |
| BA516 | DIODE | U1 | 1 |
| DHS104G | DIODE BRIDGE SMT DHS104G | D3 | 1 |
| TR180 | Variable Resister | RV1 | 1 |
| CEC45-476016W | 47UF/16V | C1,C4 | 2 |
| CMC5X102K50 | 1000PF/50V | C5 | 1 |
| CMC5X103K50 | 0.01UF/50V | C11,C12,C13 | 3 |
| CZ012X7R1H104K | 0.1UF/50V | C2,C3 | 2 |
| CMC5X105KCH | 1UF/25V | C7,C8,C9,C10 | 4 |
| CMC6Y106Z1E | 10UF/25V | C6 | 1 |
| 0805-5-470R | 470 ohm | R2,R3 | 2 |
| 0805-5-220R | 220 ohm | R1 | 1 |
| 0805-5-10R | 10 ohm | R4 | 1 |
| ANT-300000 | Antenna 3.0GHz (Male) 50ohm | J3 | 1 |
| ANT-300000 | Antenna 3.0GHz (Female) 50ohm | J4 | 1 |

*PROVIDED BY GIGA-TMS INC.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ORDERING INFORMATION

RDL200T-00: (Radio Data Link Module)
P/N: RDL200T-00

RDL200T Start Kit Package: (ASY-RDL200TSK x 2PCS)
P/N: RDL200T-SKE

E: Include 230VAC/9VDC Adaptor
U: Include 120VAC/9VDC Adaptor

RDL401A-00 (Stand-Alone Radio Data Link)
P/N: RDL401AE

E: Include 230VAC/9VDC Adaptor
U: Include 120VAC/9VDC Adaptor



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OPTIONAL, ANTENNA SOCKET AND ANTENNA

1. Antenna Socket

P/N: **CN5-OTG0087** (Angle 180)

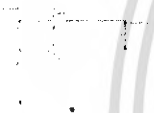


P/N: **CN5-OTG0088** (Angle 90)



2. Antenna Extension Cable

P/N: **WAS-1528**



3. Antenna

P/N: **ANT-T001** (50 ohms, 434±5 MHz)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

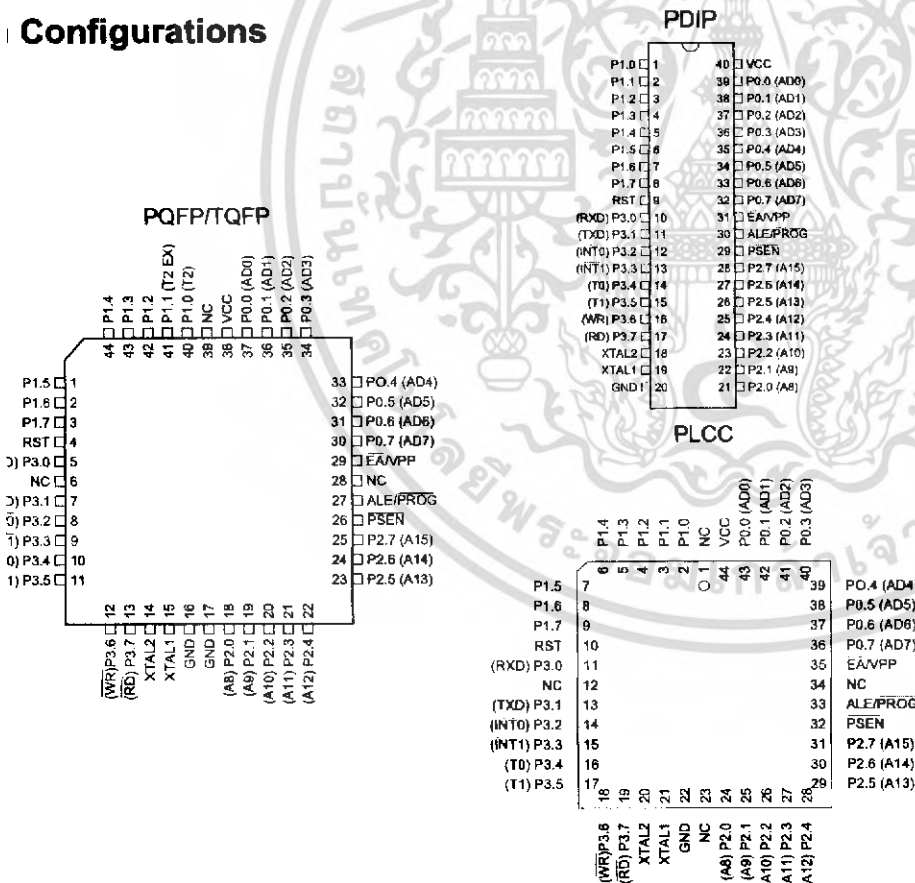
Features

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

Description

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

Configurations



8-bit Microcontroller with 4K Bytes Flash

AT89C51

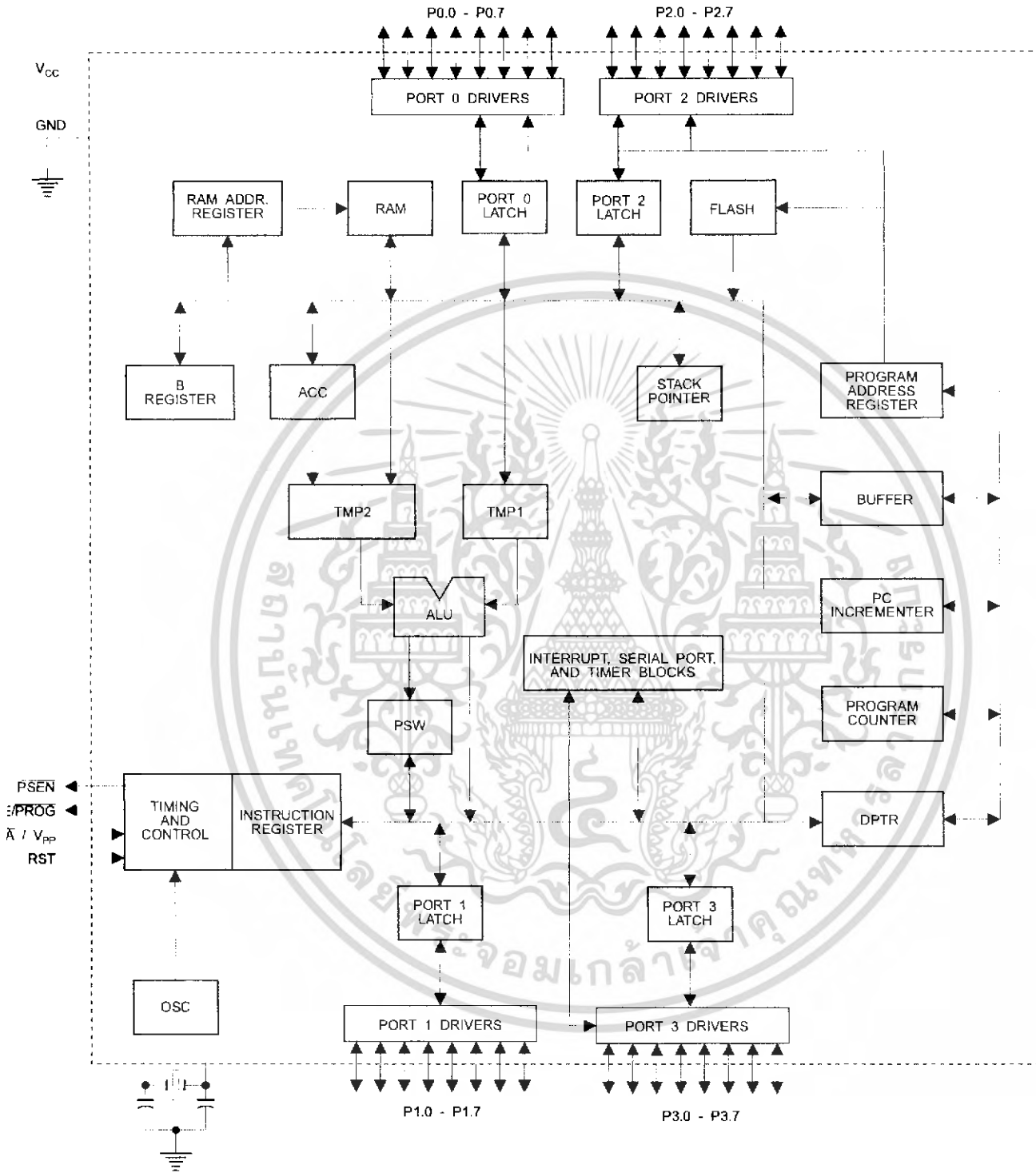
Not Recommended for New Designs. Use AT89S51.

Rev. 0265G-02/00



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Block Diagram



AT89C51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

1 Description

1.1 Supply voltage.

Supply voltage.

1.2 ID

found.

1.3 Port 0

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

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

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

1.4 Port 1

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

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

1.5 Port 2

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

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

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

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

1.6 Port 3

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

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

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

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

1.7 RST

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

1.8 ALE/PROG

Address Latch Enable output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input (PROG) during Flash programming.

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



se is skipped during each access to external Data memory.

esired, ALE operation can be disabled by setting bit 0 of R location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is actively pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

\overline{EN}

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

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

\overline{VPP}

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

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

This pin also receives the 12-volt programming enable voltage (V_{PP}) during Flash programming, for parts that require a 12-volt V_{PP} .

XTAL1

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

XTAL2

Input from the inverting oscillator amplifier.

Oscillator Characteristics

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

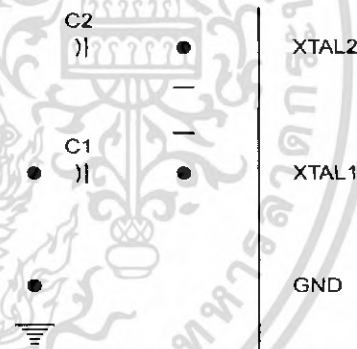
unconnected while XTAL1 is driven as shown in Figure 2. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Idle Mode

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

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

Figure 1. Oscillator Connections



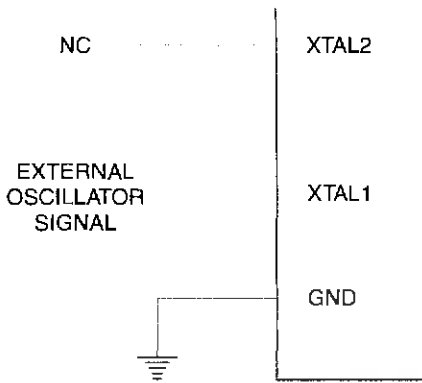
Note: C1, C2 = 30 pF \pm 10 pF for Crystals
= 40 pF \pm 10 pF for Ceramic Resonators

Status of External Pins During Idle and Power-down Modes

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

AT89C51

Figure 2. External Clock Drive Configuration



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

Program Memory Lock Bits

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

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

Power-down Mode

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

Lock Bit Protection Modes

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





Programming the Flash

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

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

| | V _{PP} = 12V | V _{PP} = 5V |
|---------------|--|--|
| Pin-side Mark | AT89C51 xxxx yyww | AT89C51 xxxx-5 yyww |
| Signature | (030H) = 1EH (031H) = 51H (032H) = FFH | (030H) = 1EH (031H) = 51H (032H) = 05H |

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

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

Input the desired memory location on the address lines.

Input the appropriate data byte on the data lines.

Activate the correct combination of control signals.

Raise \overline{EA}/V_{PP} to 12V for the high-voltage programming mode.

Pulse ALE/PROG once to program a byte in the Flash array or the lock bits. The byte-write cycle is self-timed and typically takes no more than 1.5 ms.

Repeat steps 1 through 5, changing the address

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

Data Polling: The AT89C51 features Data Polling to indicate the end of a write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written datum on PO.7. Once the write cycle has been completed, true data are valid on all outputs, and the next cycle may begin. Data Polling may begin any time after a write cycle has been initiated.

Ready/Busy: The progress of byte programming can also be monitored by the RDY/ \overline{BSY} output signal. P3.4 is pulled low after ALE goes high during programming to indicate BUSY. P3.4 is pulled high again when programming is done to indicate READY.

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

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

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

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

Programming Interface

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

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

AT89C51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

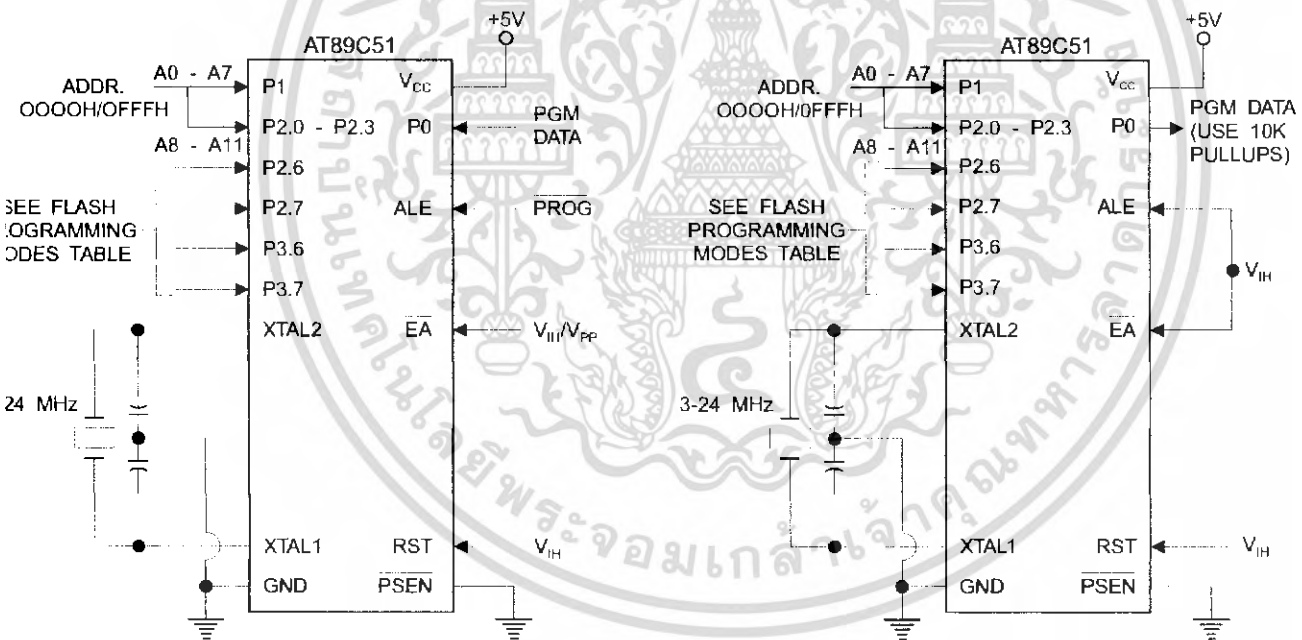
Flash Programming Modes

| Mode | RST | PSEN | ALE/PROG | EA/V _{pp} | P2.6 | P2.7 | P3.6 | P3.7 |
|---------------------|---------|------|----------|--------------------|-------|------|------|------|
| Write Code Data | H | L | | H/12V | L | H | H | H |
| Read Code Data | H | L | H | H | L | L | H | H |
| Write Lock | Bit - 1 | H | L | | H/12V | H | H | H |
| | Bit - 2 | H | L | | H/12V | H | H | L |
| | Bit - 3 | H | L | | H/12V | H | L | L |
| Chip Erase | H | L | | H/12V | H | L | L | L |
| Read Signature Byte | H | L | H | H | L | L | L | L |

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

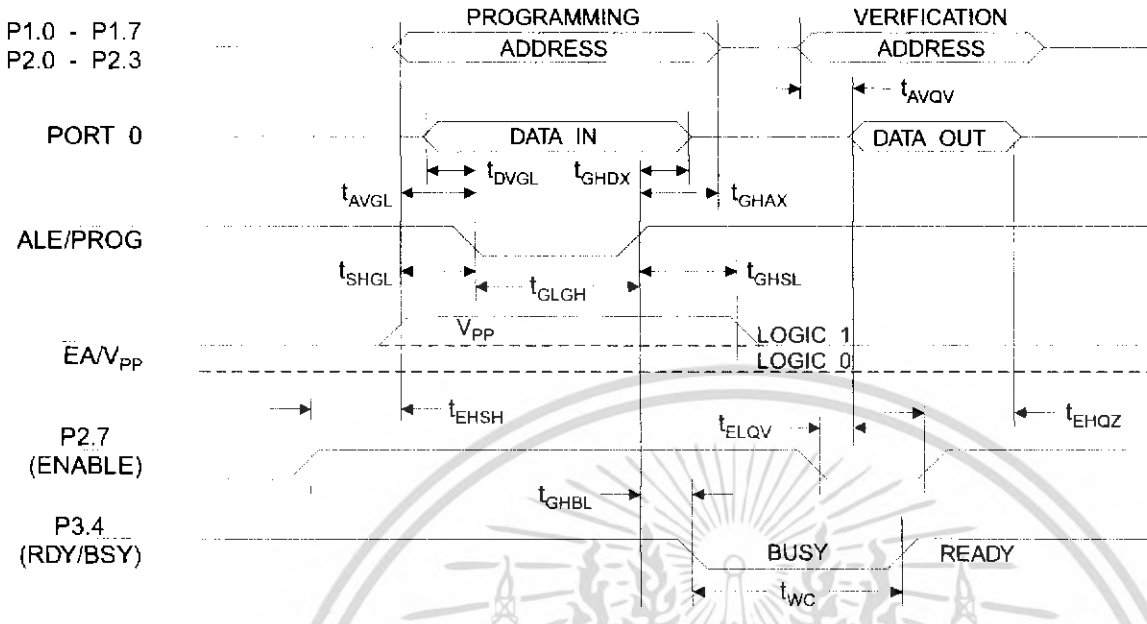
Figure 3. Programming the Flash

Figure 4. Verifying the Flash

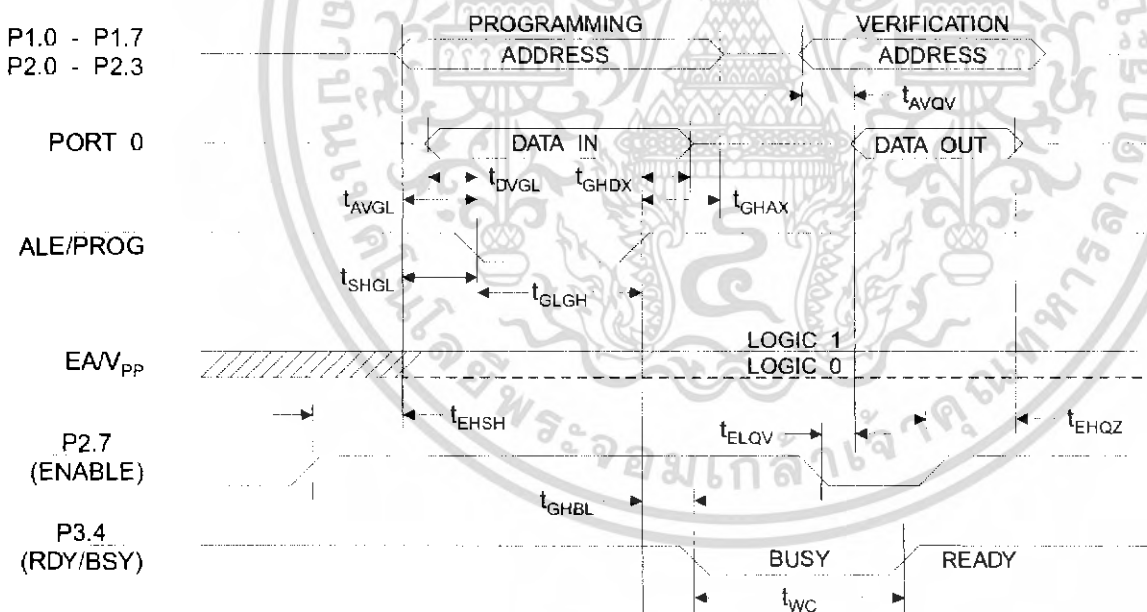




sh Programming and Verification Waveforms - High-voltage Mode ($V_{PP} = 12V$)



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



AT89C51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

sh Programming and Verification Characteristics

0°C to 70°C, V_{CC} = 5.0 ± 10%

| Symbol | Parameter | Min | Max | Units |
|--------------------------------|---|---------------------|---------------------|-------|
| (1) | Programming Enable Voltage | 11.5 | 12.5 | V |
| (1) | Programming Enable Current | | 1.0 | mA |
| f _{CL} | Oscillator Frequency | 3 | 24 | MHz |
| t _{SL} | Address Setup to $\overline{\text{PROG}}$ Low | 48t _{CLCL} | | |
| t _{SH} | Address Hold after $\overline{\text{PROG}}$ | 48t _{CLCL} | | |
| t _{DL} | Data Setup to $\overline{\text{PROG}}$ Low | 48t _{CLCL} | | |
| t _{DH} | Data Hold after $\overline{\text{PROG}}$ | 48t _{CLCL} | | |
| t _{PH} | P2.7 ($\overline{\text{ENABLE}}$) High to V _{PP} | 48t _{CLCL} | | |
| t _{PL} | V _{PP} Setup to $\overline{\text{PROG}}$ Low | 10 | | μs |
| t _{PL} ⁽¹⁾ | V _{PP} Hold after $\overline{\text{PROG}}$ | 10 | | μs |
| t _{PH} | $\overline{\text{PROG}}$ Width | 1 | 110 | μs |
| t _{AV} | Address to Data Valid | | 48t _{CLCL} | |
| t _{EV} | $\overline{\text{ENABLE}}$ Low to Data Valid | | 48t _{CLCL} | |
| t _{DZ} | Data Float after $\overline{\text{ENABLE}}$ | 0 | 48t _{CLCL} | |
| t _{HL} | $\overline{\text{PROG}}$ High to $\overline{\text{BUSY}}$ Low | | 1.0 | μs |
| | Byte Write Cycle Time | | 2.0 | ms |

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



Absolute Maximum Ratings*

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

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

Characteristics

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

| Symbol | Parameter | Condition | Min | Max | Units |
|--------|---|---|--------------------|--------------------|------------------|
| | Input Low-voltage | (Except $\bar{E}A$) | -0.5 | $0.2 V_{CC} - 0.1$ | V |
| | Input Low-voltage ($\bar{E}A$) | | -0.5 | $0.2 V_{CC} - 0.3$ | V |
| | Input High-voltage | (Except XTAL1, RST) | $0.2 V_{CC} + 0.9$ | $V_{CC} + 0.5$ | V |
| | Input High-voltage | (XTAL1, RST) | $0.7 V_{CC}$ | $V_{CC} + 0.5$ | V |
| | Output Low-voltage ⁽¹⁾ (Ports 1,2,3) | $I_{OL} = 1.6 \text{ mA}$ | | 0.45 | V |
| | Output Low-voltage ⁽¹⁾ (Port 0, ALE, PSEN) | $I_{OL} = 3.2 \text{ mA}$ | | 0.45 | V |
| | Output High-voltage (Ports 1,2,3, ALE, PSEN) | $I_{OH} = -60 \mu\text{A}, V_{CC} = 5V \pm 10\%$ | 2.4 | | V |
| | | $I_{OH} = -25 \mu\text{A}$ | $0.75 V_{CC}$ | | V |
| | | $I_{OH} = -10 \mu\text{A}$ | $0.9 V_{CC}$ | | V |
| | Output High-voltage (Port 0 in External Bus Mode) | $I_{OH} = -800 \mu\text{A}, V_{CC} = 5V \pm 10\%$ | 2.4 | | V |
| | | $I_{OH} = -300 \mu\text{A}$ | $0.75 V_{CC}$ | | V |
| | | $I_{OH} = -80 \mu\text{A}$ | $0.9 V_{CC}$ | | V |
| | Logical 0 Input Current (Ports 1,2,3) | $V_{IN} = 0.45V$ | | -50 | μA |
| | Logical 1 to 0 Transition Current (Ports 1,2,3) | $V_{IN} = 2V, V_{CC} = 5V \pm 10\%$ | | -650 | μA |
| | Input Leakage Current (Port 0, $\bar{E}A$) | $0.45 < V_{IN} < V_{CC}$ | | ± 10 | μA |
| ST | Reset Pull-down Resistor | | 50 | 300 | $\text{K}\Omega$ |
| | Pin Capacitance | Test Freq. = 1 MHz, $T_A = 25^\circ\text{C}$ | | 10 | pF |
| | Power Supply Current | Active Mode, 12 MHz | | 20 | mA |
| | | Idle Mode, 12 MHz | | 5 | mA |
| | Power-down Mode ⁽²⁾ | $V_{CC} = 6V$ | | 100 | μA |
| | | $V_{CC} = 3V$ | | 40 | μA |

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

Maximum I_{OL} per port pin: 10 mA

Maximum I_{OL} per 8-bit port: Port 0: 26 mA

Ports 1, 2, 3: 15 mA

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

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

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

AT89C51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

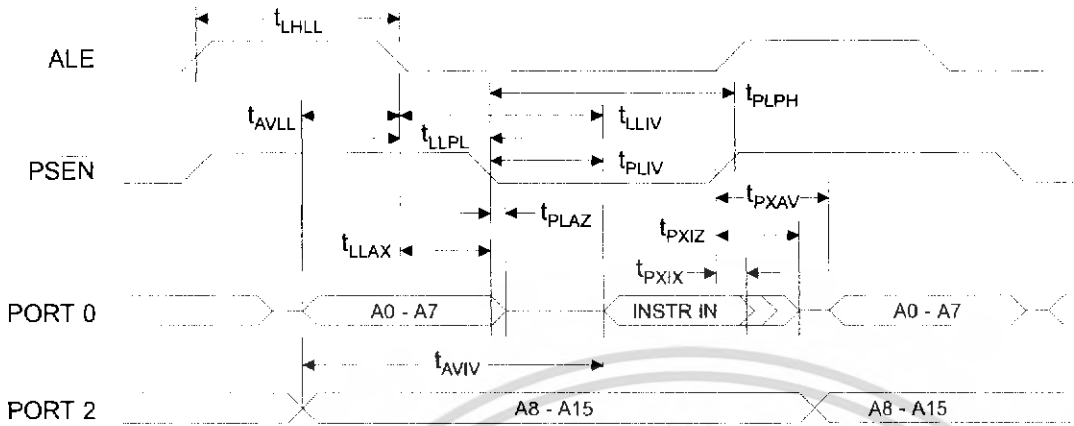
Characteristics

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

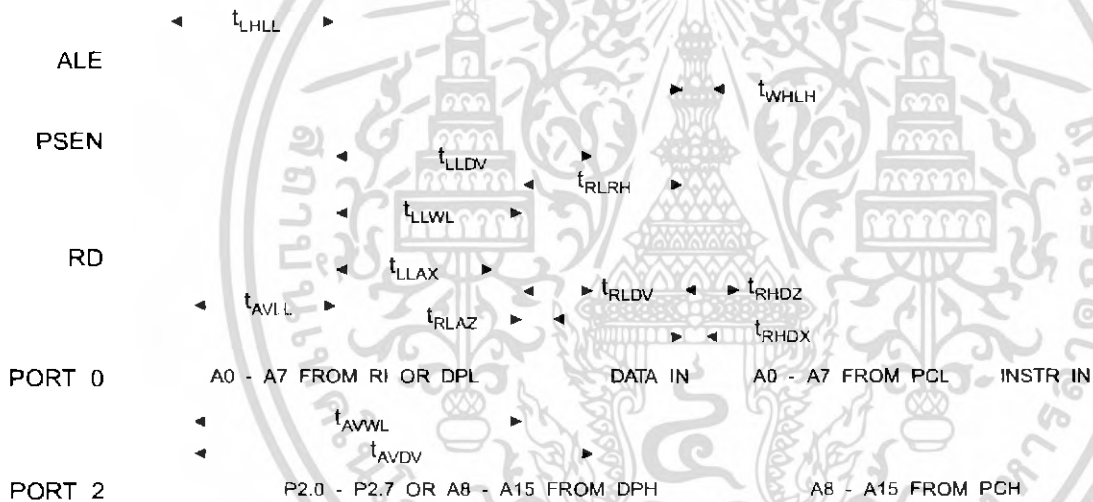
Internal Program and Data Memory Characteristics

| Symbol | Parameter | 12 MHz Oscillator | | 16 to 24 MHz Oscillator | | Units |
|-------------------|---|-------------------|-----|-------------------------|------------------------|-------|
| | | Min | Max | Min | Max | |
| t_{CLCL} | Oscillator Frequency | | | 0 | 24 | MHz |
| t_{AL} | ALE Pulse Width | 127 | | $2t_{\text{CLCL}}-40$ | | ns |
| t_{AVL} | Address Valid to ALE Low | 43 | | $t_{\text{CLCL}}-13$ | | ns |
| t_{AX} | Address Hold after ALE Low | 48 | | $t_{\text{CLCL}}-20$ | | ns |
| t_{AV} | ALE Low to Valid Instruction In | | 233 | | $4t_{\text{CLCL}}-65$ | ns |
| t_{AL} | ALE Low to $\overline{\text{PSEN}}$ Low | 43 | | $t_{\text{CLCL}}-13$ | | ns |
| t_{PH} | $\overline{\text{PSEN}}$ Pulse Width | 205 | | $3t_{\text{CLCL}}-20$ | | ns |
| t_{PV} | $\overline{\text{PSEN}}$ Low to Valid Instruction In | | 145 | | $3t_{\text{CLCL}}-45$ | ns |
| t_{IX} | Input Instruction Hold after $\overline{\text{PSEN}}$ | 0 | | 0 | | ns |
| t_{IZ} | Input Instruction Float after $\overline{\text{PSEN}}$ | | 59 | | $t_{\text{CLCL}}-10$ | ns |
| t_{AV} | $\overline{\text{PSEN}}$ to Address Valid | 75 | | $t_{\text{CLCL}}-8$ | | ns |
| t_{V} | Address to Valid Instruction In | | 312 | | $5t_{\text{CLCL}}-55$ | ns |
| t_{AZ} | $\overline{\text{PSEN}}$ Low to Address Float | | 10 | | 10 | ns |
| t_{RH} | $\overline{\text{RD}}$ Pulse Width | 400 | | $6t_{\text{CLCL}}-100$ | | ns |
| t_{WH} | $\overline{\text{WR}}$ Pulse Width | 400 | | $6t_{\text{CLCL}}-100$ | | ns |
| t_{DV} | $\overline{\text{RD}}$ Low to Valid Data In | | 252 | | $5t_{\text{CLCL}}-90$ | ns |
| t_{DX} | Data Hold after $\overline{\text{RD}}$ | 0 | | 0 | | ns |
| t_{DZ} | Data Float after $\overline{\text{RD}}$ | | 97 | | $2t_{\text{CLCL}}-28$ | ns |
| t_{DV} | ALE Low to Valid Data In | | 517 | | $8t_{\text{CLCL}}-150$ | ns |
| t_{DV} | Address to Valid Data In | | 585 | | $9t_{\text{CLCL}}-165$ | ns |
| t_{AL} | ALE Low to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ Low | 200 | 300 | $3t_{\text{CLCL}}-50$ | $3t_{\text{CLCL}}+50$ | ns |
| t_{WL} | Address to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ Low | 203 | | $4t_{\text{CLCL}}-75$ | | ns |
| t_{WX} | Data Valid to $\overline{\text{WR}}$ Transition | 23 | | $t_{\text{CLCL}}-20$ | | ns |
| t_{WH} | Data Valid to $\overline{\text{WR}}$ High | 433 | | $7t_{\text{CLCL}}-120$ | | ns |
| t_{IOX} | Data Hold after $\overline{\text{WR}}$ | 33 | | $t_{\text{CLCL}}-20$ | | ns |
| t_{AZ} | $\overline{\text{RD}}$ Low to Address Float | | 0 | | 0 | ns |
| t_{LH} | $\overline{\text{RD}}$ or $\overline{\text{WR}}$ High to ALE High | 43 | 123 | $t_{\text{CLCL}}-20$ | $t_{\text{CLCL}}+25$ | ns |

Internal Program Memory Read Cycle



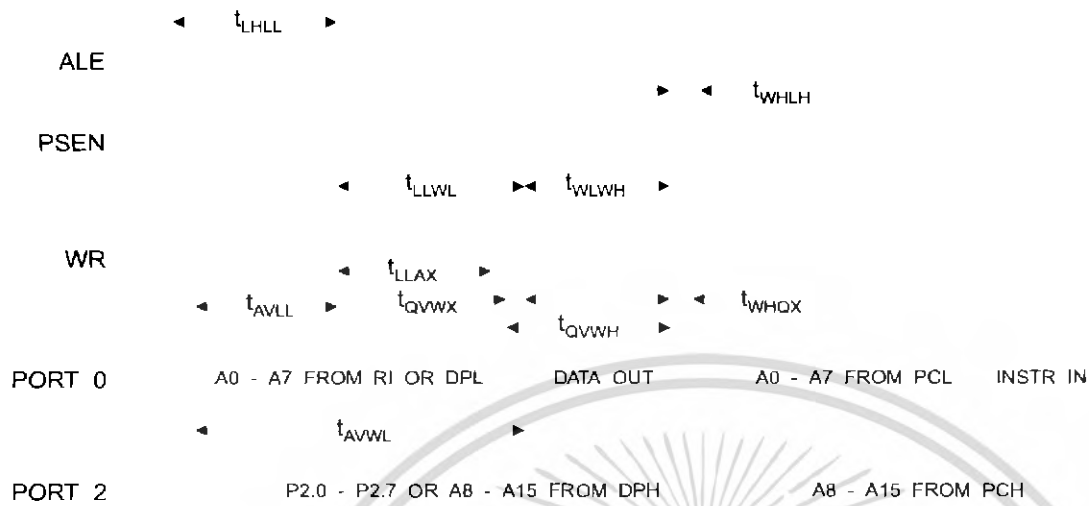
Internal Data Memory Read Cycle



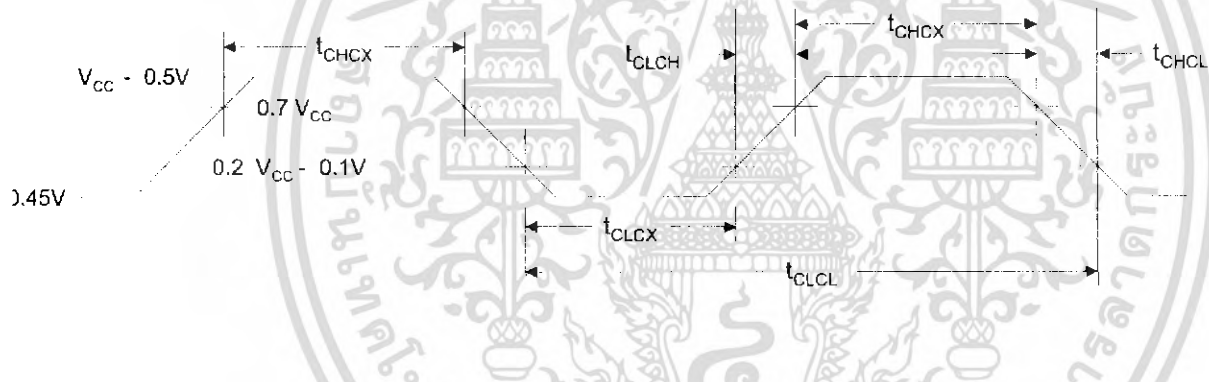
AT89C51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

External Data Memory Write Cycle



Internal Clock Drive Waveforms



Internal Clock Drive

| Symbol | Parameter | Min | Max | Units |
|-------------------|----------------------|------|-----|-------|
| f _{CLCL} | Oscillator Frequency | 0 | 24 | MHz |
| T _{CL} | Clock Period | 41.6 | | ns |
| t _{CHCX} | High Time | 15 | | ns |
| t _{CLCX} | Low Time | 15 | | ns |
| t _{CH} | Rise Time | | 20 | ns |
| t _{CL} | Fall Time | | 20 | ns |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

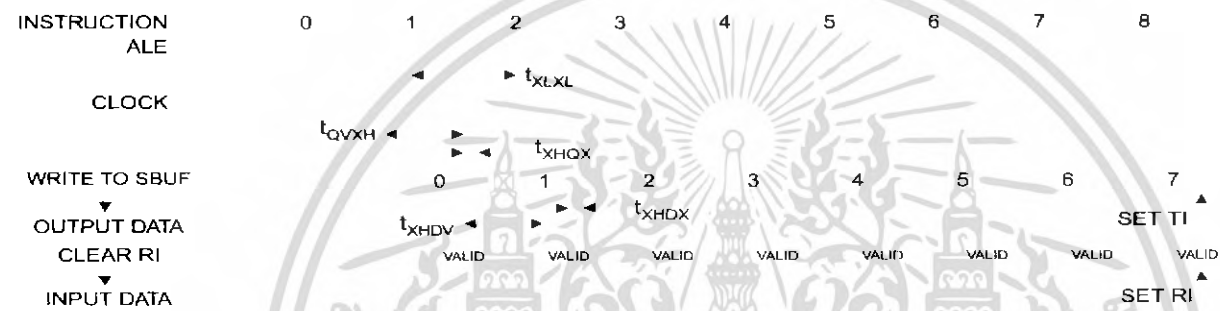


Serial Port Timing: Shift Register Mode Test Conditions

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

| Symbol | Parameter | 12 MHz Osc | | Variable Oscillator | | Units |
|-----------------|--|------------|-----|--------------------------|--------------------------|-------|
| | | Min | Max | Min | Max | |
| t _{CC} | Serial Port Clock Cycle Time | 1.0 | | 12t _{CLCL} | | μs |
| t _{OH} | Output Data Setup to Clock Rising Edge | 700 | | 10t _{CLCL} -133 | | ns |
| t _{OX} | Output Data Hold after Clock Rising Edge | 50 | | 2t _{CLCL} -117 | | ns |
| t _{IX} | Input Data Hold after Clock Rising Edge | 0 | | 0 | | ns |
| t _{IV} | Clock Rising Edge to Input Data Valid | | 700 | | 10t _{CLCL} -133 | ns |

Shift Register Mode Timing Waveforms



Testing Input/Output Waveforms⁽¹⁾

Float Waveforms⁽¹⁾



1. AC Inputs during testing are driven at V_{CC} - 0.5V for a logic 1 and 0.45V for a logic 0. Timing measurements are made at V_{IH} min. for a logic 1 and V_{IL} max. for a logic 0.

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

AT89C51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ordering Information

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

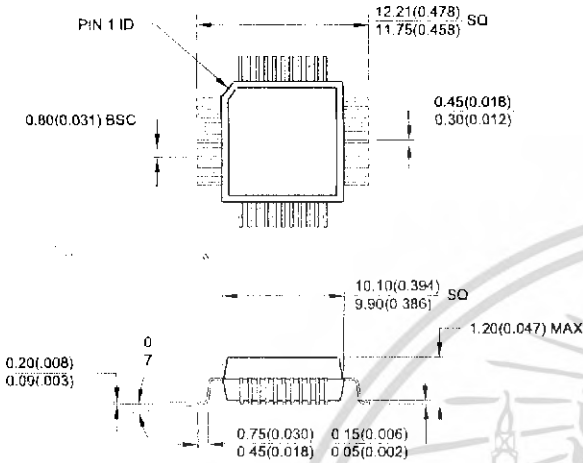
Package Type

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

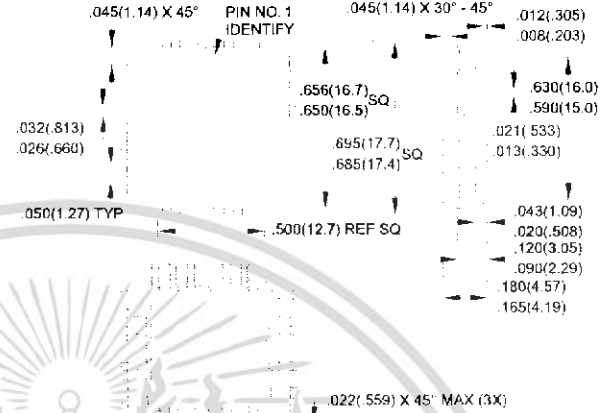


Packaging Information

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

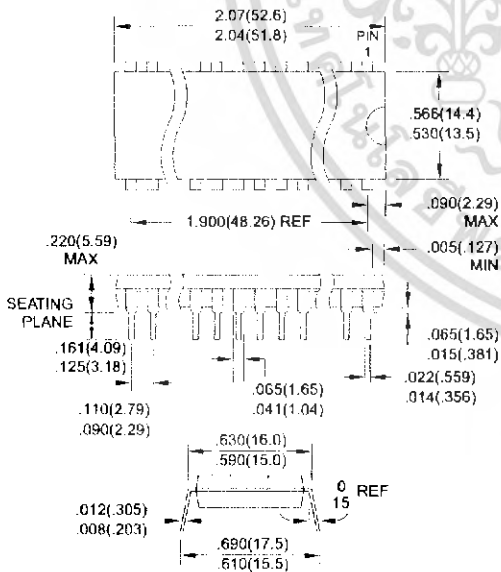


44J, 44-lead, Plastic J-leaded Chip Carrier (PLCC)
 Dimensions in Inches and (Millimeters)
 JEDEC STANDARD MS-018 AC

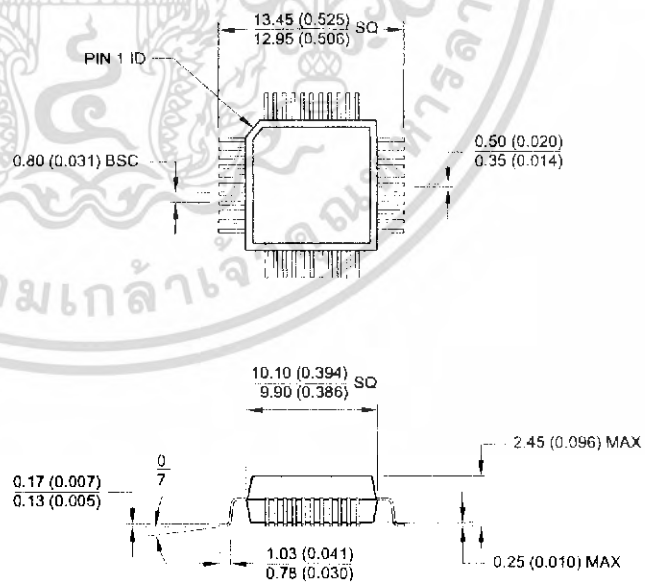


Controlling dimension: millimeters

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



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



Controlling dimension: millimeters

AT89C51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Atmel Headquarters

Corporate Headquarters

325 Orchard Parkway
Folsom, CA 95131
TEL (408) 441-0311
FAX (408) 487-2600

Europe

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

Asia

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

Japan

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

Atmel Operations

Atmel Colorado Springs

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

Atmel Rousset

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

Fax-on-Demand

North America:
1-(800) 292-8635

International:
1-(408) 441-0732

e-mail
literature@atmel.com

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

BBS
1-(408) 436-4309

Atmel Corporation 2000.

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

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

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

Printed on recycled paper.

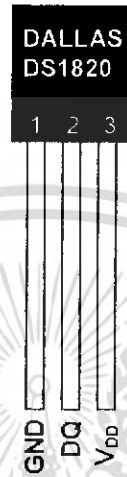
0265G-02/00/xM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FEATURES

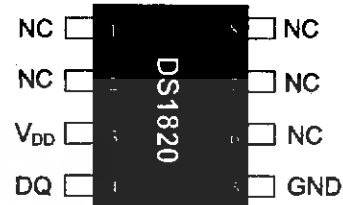
- Unique 1-wire interface requires only one port pin for communication
- Each device has a unique 64-bit serial code stored in an on-board ROM
- Multi-drop capability simplifies distributed temperature sensing applications
- Requires no external components
- Can be powered from data line. Power supply range is 3.0V to 5.5V
- Measures temperatures from -55°C to +125°C (-67°F to +257°F)
- ±0.5°C accuracy from -10°C to +85°C
- 9-bit thermometer resolution
- Converts temperature in 750 ms (max.)
- User-definable nonvolatile alarm settings
- Alarm search command identifies and addresses devices whose temperature is outside of programmed limits (temperature alarm condition)
- Applications include thermostatic controls, industrial systems, consumer products, thermometers, or any thermally sensitive system

PIN ASSIGNMENT



(BOTTOM VIEW)

TO-92
(DS18S20)



8-pin 150-mil SOIC
(DS18S20Z)

PIN DESCRIPTION

- GND - Ground
- DQ - Data In/Out
- V_{DD} - Power Supply Voltage
- NC - No Connect

DESCRIPTION

The DS18S20 Digital Thermometer provides 9-bit centigrade temperature measurements and has an alarm function with nonvolatile user-programmable upper and lower trigger points. The DS18S20 communicates over a 1-wire bus that by definition requires only one data line (and ground) for communication with a central microprocessor. It has an operating temperature range of -55°C to +125°C and is accurate to ±0.5°C over the range of -10°C to +85°C. In addition, the DS18S20 can derive power directly from the data line (“parasite power”), eliminating the need for an external power supply.

Each DS18S20 has a unique 64-bit serial code, which allows multiple DS18S20s to function on the same 1-wire bus; thus, it is simple to use one microprocessor to control many DS18S20s distributed over a large area. Applications that can benefit from this feature include HVAC environmental controls, temperature monitoring systems inside buildings, equipment or machinery, and process monitoring and control systems.

DETAILED PIN DESCRIPTIONS Table 1

| 8-PIN SOIC* | TO-92 | SYMBOL | DESCRIPTION |
|-------------|-------|-----------------|---|
| 5 | 1 | GND | Ground. |
| 4 | 2 | DQ | Data Input/Output pin. Open-drain 1-wire interface pin. Also provides power to the device when used in parasite power mode (see "Parasite Power" section.) |
| 3 | 3 | V _{DD} | Optional V_{DD} pin. V_{DD} must be grounded for operation in parasite power mode. |

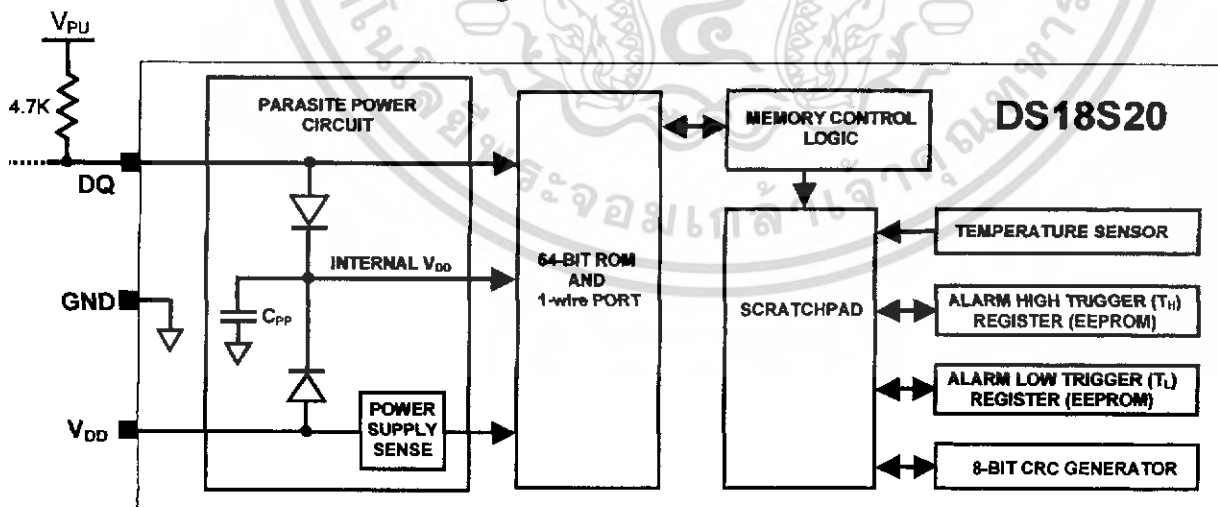
*All pins not specified in this table are "No Connect" pins.

OVERVIEW

Figure 1 shows a block diagram of the DS18S20, and pin descriptions are given in Table 1. The 64-bit ROM stores the device's unique serial code. The scratchpad memory contains the 2-byte temperature register that stores the digital output from the temperature sensor. In addition, the scratchpad provides access to the 1-byte upper and lower alarm trigger registers (T_H and T_L). The T_H and T_L registers are nonvolatile (EEPROM), so they will retain data when the device is powered down.

The DS18S20 uses Dallas' exclusive 1-wire bus protocol that implements bus communication using one control signal. The control line requires a weak pullup resistor since all devices are linked to the bus via a 3-state or open-drain port (the DQ pin in the case of the DS18S20). In this bus system, the microprocessor (the master device) identifies and addresses devices on the bus using each device's unique 64-bit code. Because each device has a unique code, the number of devices that can be addressed on one bus is virtually unlimited. The 1-wire bus protocol, including detailed explanations of the commands and "time slots," is covered in the 1-WIRE BUS SYSTEM section of this datasheet.

Another feature of the DS18S20 is the ability to operate without an external power supply. Power is instead supplied through the 1-wire pullup resistor via the DQ pin when the bus is high. The high bus signal also charges an internal capacitor (C_{PP}), which then supplies power to the device when the bus is low. This method of deriving power from the 1-wire bus is referred to as "parasite power." As an alternative, the DS18S20 may also be powered by an external supply on V_{DD}.

DS18S20 BLOCK DIAGRAM Figure 1

OPERATION – MEASURING TEMPERATURE

The core functionality of the DS18S20 is its direct-to-digital temperature sensor. The temperature sensor output has 9-bit resolution, which corresponds to 0.5°C steps. The DS18S20 powers-up in a low-power idle state; to initiate a temperature measurement and A-to-D conversion, the master must issue a Convert T [44h] command. Following the conversion, the resulting thermal data is stored in the 2-byte temperature register in the scratchpad memory and the DS18S20 returns to its idle state. If the DS18S20 is powered by an external supply, the master can issue “read time slots” (see the 1-WIRE BUS SYSTEM section) after the Convert T command and the DS18S20 will respond by transmitting 0 while the temperature conversion is in progress and 1 when the conversion is done. If the DS18S20 is powered with parasite power, this notification technique cannot be used since the bus must be pulled high by a strong pullup during the entire temperature conversion. The bus requirements for parasite power are explained in detail in the POWERING THE DS18S20 section of this datasheet.

The DS18S20 output data is calibrated in degrees centigrade; for Fahrenheit applications, a lookup table or conversion routine must be used. The temperature data is stored as a 16-bit sign-extended two’s complement number in the temperature register (see Figure 2). The sign bits (S) indicate if the temperature is positive or negative: for positive numbers S = 0 and for negative numbers S = 1. Table 2 gives examples of digital output data and the corresponding temperature reading.

Resolutions greater than 9 bits can be calculated using the data from the temperature, COUNT REMAIN and COUNT PER °C registers in the scratchpad. Note that the COUNT PER °C register is hard-wired to 16 (10h). After reading the scratchpad, the TEMP_READ value is obtained by truncating the 0.5°C bit (bit 0) from the temperature data (see Figure 2). The extended resolution temperature can then be calculated using the following equation:

$$TEMPERATURE = TEMP_READ - 0.25 + \frac{COUNT_PER_C - COUNT_REMAIN}{COUNT_PER_C}$$

Additional information about high-resolution temperature calculations can be found in Application Note 105: “High Resolution Temperature Measurement with Dallas Direct-to-Digital Temperature Sensors”.

TEMPERATURE REGISTER FORMAT Figure 2

| | | | | | | | | |
|---------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|-----------------|
| | bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 |
| LS Byte | 2 ⁶ | 2 ⁵ | 2 ⁴ | 2 ³ | 2 ² | 2 ¹ | 2 ⁰ | 2 ⁻¹ |
| | bit 15 | bit 14 | bit 13 | bit 12 | bit 11 | bit 10 | bit 9 | bit 8 |
| MS Byte | S | S | S | S | S | S | S | S |

TEMPERATURE/DATA RELATIONSHIP Table 2

| TEMPERATURE | DIGITAL OUTPUT (Binary) | DIGITAL OUTPUT (Hex) |
|-------------|----------------------------|-------------------------|
| +85.0°C* | 0000 0000 1010 1010 | 00AAh |
| +25.0°C | 0000 0000 0011 0010 | 0032h |
| +0.5°C | 0000 0000 0000 0001 | 0001h |
| 0°C | 0000 0000 0000 0000 | 0000h |
| -0.5°C | 1111 1111 1111 1111 | FFFFh |
| -25.0°C | 1111 1111 1100 1110 | FFCEh |
| -55.0°C | 1111 1111 1001 0010 | FF92h |

*The power-on reset value of the temperature register is +85°C

OPERATION – ALARM SIGNALING

After the DS18S20 performs a temperature conversion, the temperature value is compared to the user-defined two's complement alarm trigger values stored in the 1-byte T_H and T_L registers (see Figure 3). The sign bit (S) indicates if the value is positive or negative: for positive numbers $S = 0$ and for negative numbers $S = 1$. The T_H and T_L registers are nonvolatile (EEPROM) so they will retain data when the device is powered down. T_H and T_L can be accessed through bytes 2 and 3 of the scratchpad as explained in the MEMORY section of this datasheet.

T_H AND T_L REGISTER FORMAT Figure 3

| bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| S | 2^6 | 2^5 | 2^5 | 2^5 | 2^2 | 2^1 | 2^0 |

Only bits 8 through 1 of the temperature register are used in the T_H and T_L comparison since T_H and T_L are 8-bit registers. If the result of a temperature measurement is higher than T_H or lower than T_L , an alarm condition exists and an alarm flag is set inside the DS18S20. This flag is updated after every temperature measurement; therefore, if the alarm condition goes away, the flag will be turned off after the next temperature conversion.

The master device can check the alarm flag status of all DS18S20s on the bus by issuing an Alarm Search [ECh] command. Any DS18S20s with a set alarm flag will respond to the command, so the master can determine exactly which DS18S20s have experienced an alarm condition. If an alarm condition exists and the T_H or T_L settings have changed, another temperature conversion should be done to validate the alarm condition.

POWERING THE DS18S20

The DS18S20 can be powered by an external supply on the V_{DD} pin, or it can operate in “parasite power” mode, which allows the DS18S20 to function without a local external supply. Parasite power is very useful for applications that require remote temperature sensing or that are very space constrained. Figure 1 shows the DS18S20's parasite-power control circuitry, which “steals” power from the 1-wire bus via the DQ pin when the bus is high. The stolen charge powers the DS18S20 while the bus is high, and some of the charge is stored on the parasite power capacitor (C_{PP}) to provide power when the bus is low. When the DS18S20 is used in parasite power mode, the V_{DD} pin must be connected to ground.

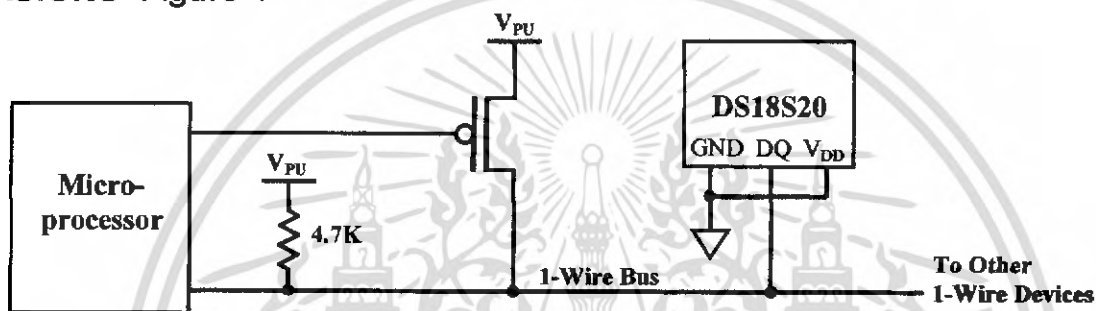
In parasite power mode, the 1-wire bus and C_{PP} can provide sufficient current to the DS18S20 for most operations as long as the specified timing and voltage requirements are met (refer to the DC ELECTRICAL CHARACTERISTICS and the AC ELECTRICAL CHARACTERISTICS sections of this data sheet). However, when the DS18S20 is performing temperature conversions or copying data from the scratchpad memory to EEPROM, the operating current can be as high as 1.5 mA. This current can cause an unacceptable voltage drop across the weak 1-wire pullup resistor and is more current than can be supplied by C_{PP} . To assure that the DS18S20 has sufficient supply current, it is necessary to provide a strong pullup on the 1-wire bus whenever temperature conversions are taking place or data is being copied from the scratchpad to EEPROM. This can be accomplished by using a MOSFET to pull the bus directly to the rail as shown in Figure 4. The 1-wire bus must be switched to the strong pullup within 10 μ s (max) after a Convert T [44h] or Copy Scratchpad [48h] command is issued, and the bus must be held high by the pullup for the duration of the conversion (t_{conv}) or data transfer ($t_{wr} = 10$ ms). No other activity can take place on the 1-wire bus while the pullup is enabled.

The DS18S20 can also be powered by the conventional method of connecting an external power supply to the V_{DD} pin, as shown in Figure 5. The advantage of this method is that the MOSFET pullup is not required, and the 1-wire bus is free to carry other traffic during the temperature conversion time.

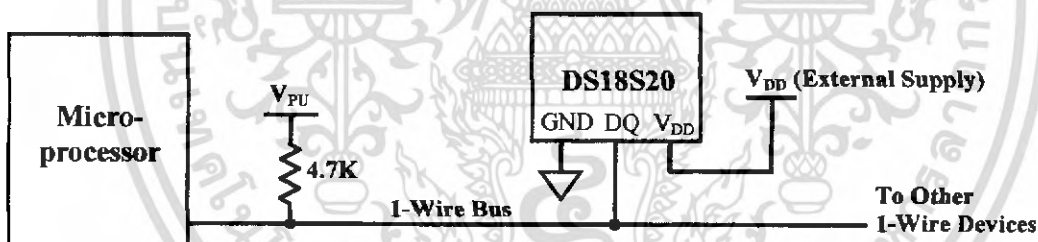
The use of parasite power is not recommended for temperatures above 100°C since the DS18S20 may not be able to sustain communications due to the higher leakage currents that can exist at these temperatures. For applications in which such temperatures are likely, it is strongly recommended that the DS18S20 be powered by an external power supply.

In some situations the bus master may not know whether the DS18S20s on the bus are parasite powered or powered by external supplies. The master needs this information to determine if the strong bus pullup should be used during temperature conversions. To get this information, the master can issue a Skip ROM [CCh] command followed by a Read Power Supply [B4h] command followed by a “read time slot”. During the read time slot, parasite powered DS18S20s will pull the bus low, and externally powered DS18S20s will let the bus remain high. If the bus is pulled low, the master knows that it must supply the strong pullup on the I-wire bus during temperature conversions.

SUPPLYING THE PARASITE-POWERED DS18S20 DURING TEMPERATURE CONVERSIONS Figure 4



POWERING THE DS18S20 WITH AN EXTERNAL SUPPLY Figure 5



64-BIT LASERED ROM CODE

Each DS18S20 contains a unique 64-bit code (see Figure 6) stored in ROM. The least significant 8 bits of the ROM code contain the DS18S20's 1-wire family code: 10h. The next 48 bits contain a unique serial number. The most significant 8 bits contain a cyclic redundancy check (CRC) byte that is calculated from the first 56 bits of the ROM code. A detailed explanation of the CRC bits is provided in the CRC GENERATION section. The 64-bit ROM code and associated ROM function control logic allow the DS18S20 to operate as a 1-wire device using the protocol detailed in the 1-WIRE BUS SYSTEM section of this datasheet.

64-BIT LASERED ROM CODE Figure 6

| 8-BIT CRC | | 48-BIT SERIAL NUMBER | | | 8-BIT FAMILY CODE (10h) | |
|-----------|-----|----------------------|-----|-----|-------------------------|-----|
| MSB | LSB | MSB | LSB | MSB | LSB | LSB |

MEMORY

The DS18S20's memory is organized as shown in Figure 7. The memory consists of an SRAM scratchpad with nonvolatile EEPROM storage for the high and low alarm trigger registers (T_H and T_L). Note that if the DS18S20 alarm function is not used, the T_H and T_L registers can serve as general-purpose memory. All memory commands are described in detail in the DS18S20 FUNCTION COMMANDS section.

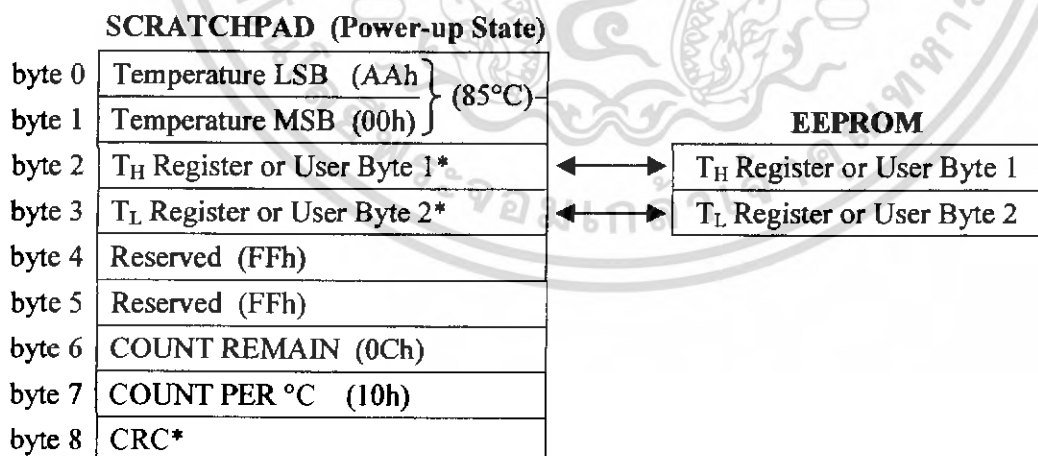
Byte 0 and byte 1 of the scratchpad contain the LSB and the MSB of the temperature register, respectively. These bytes are read-only. Bytes 2 and 3 provide access to T_H and T_L registers. Bytes 4 and 5 are reserved for internal use by the device and cannot be overwritten; these bytes will return all 1s when read. Bytes 6 and 7 contain the COUNT REMAIN and COUNT PER °C registers, which can be used to calculate extended resolution results as explained in the OPERATION – MEASURING TEMPERATURE section.

Byte 8 of the scratchpad is read-only and contains the cyclic redundancy check (CRC) code for bytes 0 through 7 of the scratchpad. The DS18S20 generates this CRC using the method described in the CRC GENERATION section.

Data is written to bytes 2 and 3 of the scratchpad using the Write Scratchpad [4Eh] command; the data must be transmitted to the DS18S20 starting with the least significant bit of byte 2. To verify data integrity, the scratchpad can be read (using the Read Scratchpad [BEh] command) after the data is written. When reading the scratchpad, data is transferred over the 1-wire bus starting with the least significant bit of byte 0. To transfer the T_H and T_L data from the scratchpad to EEPROM, the master must issue the Copy Scratchpad [48h] command.

Data in the EEPROM registers is retained when the device is powered down; at power-up the EEPROM data is reloaded into the corresponding scratchpad locations. Data can also be reloaded from EEPROM to the scratchpad at any time using the Recall E² [B8h] command. The master can issue “read time slots” (see the 1-WIRE BUS SYSTEM section) following the Recall E² command and the DS18S20 will indicate the status of the recall by transmitting 0 while the recall is in progress and 1 when the recall is done.

DS18S20 MEMORY MAP



*Power-up state depends on value(s) stored in EEPROM

CRC GENERATION

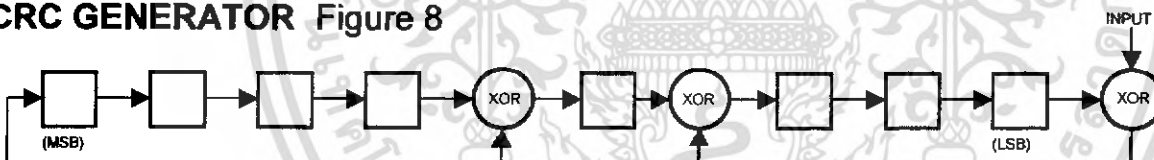
CRC bytes are provided as part of the DS18S20's 64-bit ROM code and in the 9th byte of the scratchpad memory. The ROM code CRC is calculated from the first 56 bits of the ROM code and is contained in the most significant byte of the ROM. The scratchpad CRC is calculated from the data stored in the scratchpad, and therefore it changes when the data in the scratchpad changes. The CRCs provide the bus master with a method of data validation when data is read from the DS18S20. To verify that data has been read correctly, the bus master must re-calculate the CRC from the received data and then compare this value to either the ROM code CRC (for ROM reads) or to the scratchpad CRC (for scratchpad reads). If the calculated CRC matches the read CRC, the data has been received error free. The comparison of CRC values and the decision to continue with an operation are determined entirely by the bus master. There is no circuitry inside the DS18S20 that prevents a command sequence from proceeding if the DS18S20 CRC (ROM or scratchpad) does not match the value generated by the bus master.

The equivalent polynomial function of the CRC (ROM or scratchpad) is:

$$\text{CRC} = X^8 + X^5 + X^4 + 1$$

The bus master can re-calculate the CRC and compare it to the CRC values from the DS18S20 using the polynomial generator shown in Figure 8. This circuit consists of a shift register and XOR gates, and the shift register bits are initialized to 0. Starting with the least significant bit of the ROM code or the least significant bit of byte 0 in the scratchpad, one bit at a time should be shifted into the shift register. After shifting in the 56th bit from the ROM or the most significant bit of byte 7 from the scratchpad, the polynomial generator will contain the re-calculated CRC. Next, the 8-bit ROM code or scratchpad CRC from the DS18S20 must be shifted into the circuit. At this point, if the re-calculated CRC was correct, the shift register will contain all 0s. Additional information about the Dallas 1-wire cyclic redundancy check is available in Application Note 27 entitled "Understanding and Using Cyclic Redundancy Checks with Dallas Semiconductor Touch Memory Products."

CRC GENERATOR Figure 8



1-WIRE BUS SYSTEM

The 1-wire bus system uses a single bus master to control one or more slave devices. The DS18S20 is always a slave. When there is only one slave on the bus, the system is referred to as a “single-drop” system; the system is “multi-drop” if there are multiple slaves on the bus.

All data and commands are transmitted least significant bit first over the 1-wire bus.

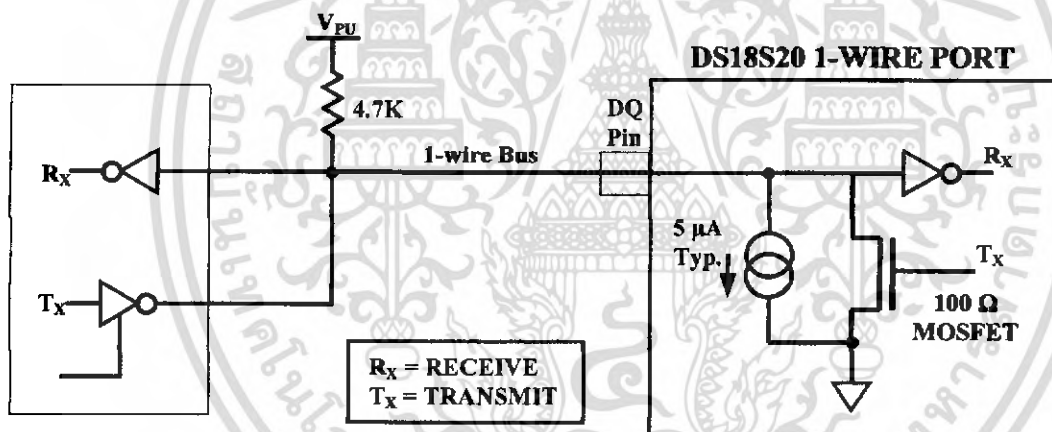
The following discussion of the 1-wire bus system is broken down into three topics: hardware configuration, transaction sequence, and 1-wire signaling (signal types and timing).

HARDWARE CONFIGURATION

The 1-wire bus has by definition only a single data line. Each device (master or slave) interfaces to the data line via an open drain or 3-state port. This allows each device to “release” the data line when the device is not transmitting data so the bus is available for use by another device. The 1-wire port of the DS18S20 (the DQ pin) is open drain with an internal circuit equivalent to that shown in Figure 9.

The 1-wire bus requires an external pullup resistor of approximately 5 k Ω ; thus, the idle state for the 1-wire bus is high. If for any reason a transaction needs to be suspended, the bus MUST be left in the idle state if the transaction is to resume. Infinite recovery time can occur between bits so long as the 1-wire bus is in the inactive (high) state during the recovery period. If the bus is held low for more than 480 μ s, all components on the bus will be reset.

HARDWARE CONFIGURATION Figure 9



TRANSACTION SEQUENCE

The transaction sequence for accessing the DS18S20 is as follows:

- Step 1. Initialization
- Step 2. ROM Command (followed by any required data exchange)
- Step 3. DS18S20 Function Command (followed by any required data exchange)

It is very important to follow this sequence every time the DS18S20 is accessed, as the DS18S20 will not respond if any steps in the sequence are missing or out of order. Exceptions to this rule are the Search ROM [F0h] and Alarm Search [ECh] commands. After issuing either of these ROM commands, the master must return to Step 1 in the sequence.

INITIALIZATION

All transactions on the 1-wire bus begin with an initialization sequence. The initialization sequence consists of a reset pulse transmitted by the bus master followed by presence pulse(s) transmitted by the slave(s). The presence pulse lets the bus master know that slave devices (such as the DS18S20) are on the bus and are ready to operate. Timing for the reset and presence pulses is detailed in the 1-WIRE SIGNALING section.

ROM COMMANDS

After the bus master has detected a presence pulse, it can issue a ROM command. These commands operate on the unique 64-bit ROM codes of each slave device and allow the master to single out a specific device if many are present on the 1-wire bus. These commands also allow the master to determine how many and what types of devices are present on the bus or if any device has experienced an alarm condition. There are five ROM commands, and each command is 8 bits long. The master device must issue an appropriate ROM command before issuing a DS18S20 function command. A flowchart for operation of the ROM commands is shown in Figure 14.

SEARCH ROM [F0h]

When a system is initially powered up, the master must identify the ROM codes of all slave devices on the bus, which allows the master to determine the number of slaves and their device types. The master learns the ROM codes through a process of elimination that requires the master to perform a Search ROM cycle (i.e., Search ROM command followed by data exchange) as many times as necessary to identify all of the slave devices. If there is only one slave on the bus, the simpler Read ROM command (see below) can be used in place of the Search ROM process. For a detailed explanation of the Search ROM procedure, refer to the iButton Book of Standards at www.ibutton.com/ibuttons/standard.pdf. After every Search ROM cycle, the bus master must return to Step 1 (Initialization) in the transaction sequence.

READ ROM [33h]

This command can only be used when there is one slave on the bus. It allows the bus master to read the slave's 64-bit ROM code without using the Search ROM procedure. If this command is used when there is more than one slave present on the bus, a data collision will occur when all the slaves attempt to respond at the same time.

MATCH ROM [55h]

The match ROM command followed by a 64-bit ROM code sequence allows the bus master to address a specific slave device on a multi-drop or single-drop bus. Only the slave that exactly matches the 64-bit ROM code sequence will respond to the function command issued by the master; all other slaves on the bus will wait for a reset pulse.

SKIP ROM [CCh]

The master can use this command to address all devices on the bus simultaneously without sending out any ROM code information. For example, the master can make all DS18S20s on the bus perform simultaneous temperature conversions by issuing a Skip ROM command followed by a Convert T [44h] command. Note, however, that the Skip ROM command can only be followed by the Read Scratchpad [BEh] command when there is one slave on the bus. This sequence saves time by allowing the master to read from the device without sending its 64-bit ROM code. This sequence will cause a data collision on the bus if there is more than one slave since multiple devices will attempt to transmit data simultaneously.

ALARM SEARCH [ECh]

The operation of this command is identical to the operation of the Search ROM command except that only slaves with a set alarm flag will respond. This command allows the master device to determine if any DS18S20s experienced an alarm condition during the most recent temperature conversion. After

every Alarm Search cycle (i.e., Alarm Search command followed by data exchange), the bus master must return to Step 1 (Initialization) in the transaction sequence. Refer to the OPERATION – ALARM SIGNALING section for an explanation of alarm flag operation.

DS18S20 FUNCTION COMMANDS

After the bus master has used a ROM command to address the DS18S20 with which it wishes to communicate, the master can issue one of the DS18S20 function commands. These commands allow the master to write to and read from the DS18S20's scratchpad memory, initiate temperature conversions and determine the power supply mode. The DS18S20 function commands, which are described below, are summarized in Table 4 and illustrated by the flowchart in Figure 15.

CONVERT T [44h]

This command initiates a single temperature conversion. Following the conversion, the resulting thermal data is stored in the 2-byte temperature register in the scratchpad memory and the DS18S20 returns to its low-power idle state. If the device is being used in parasite power mode, within 10 μ s (max) after this command is issued the master must enable a strong pullup on the 1-wire bus for the duration of the conversion (t_{conv}) as described in the POWERING THE DS18S20 section. If the DS18S20 is powered by an external supply, the master can issue read time slots after the Convert T command and the DS18S20 will respond by transmitting 0 while the temperature conversion is in progress and 1 when the conversion is done. In parasite power mode this notification technique cannot be used since the bus is pulled high by the strong pullup during the conversion.

WRITE SCRATCHPAD [4Eh]

This command allows the master to write 2 bytes of data to the DS18S20's scratchpad. The first byte is written into the T_H register (byte 2 of the scratchpad), and the second byte is written into the T_L register (byte 3 of the scratchpad). Data must be transmitted least significant bit first. Both bytes MUST be written before the master issues a reset, or the data may be corrupted.

READ SCRATCHPAD [BEh]

This command allows the master to read the contents of the scratchpad. The data transfer starts with the least significant bit of byte 0 and continues through the scratchpad until the 9th byte (byte 8 – CRC) is read. The master may issue a reset to terminate reading at any time if only part of the scratchpad data is needed.

COPY SCRATCHPAD [48h]

This command copies the contents of the scratchpad T_H and T_L registers (bytes 2 and 3) to EEPROM. If the device is being used in parasite power mode, within 10 μ s (max) after this command is issued the master must enable a strong pullup on the 1-wire bus for at least 10 ms as described in the POWERING THE DS18S20 section.

RECALL E² [B8h]

This command recalls the alarm trigger values (T_H and T_L) from EEPROM and places the data in bytes 2 and 3, respectively, in the scratchpad memory. The master device can issue read time slots following the Recall E² command and the DS18S20 will indicate the status of the recall by transmitting 0 while the recall is in progress and 1 when the recall is done. The recall operation happens automatically at power-up, so valid data is available in the scratchpad as soon as power is applied to the device.

READ POWER SUPPLY [B4h]

The master device issues this command followed by a read time slot to determine if any DS18S20s on the bus are using parasite power. During the read time slot, parasite powered DS18S20s will pull the bus low, and externally powered DS18S20s will let the bus remain high. Refer to the POWERING THE DS18S20 section for usage information for this command.

DS18S20 FUNCTION COMMAND SET Table 4

| Command | Description | Protocol | 1-Wire Bus Activity After Command is Issued | Notes |
|--|--|----------|---|-------|
| TEMPERATURE CONVERSION COMMANDS | | | | |
| Convert T | Initiates temperature conversion. | 44h | DS18S20 transmits conversion status to master (not applicable for parasite-powered DS18S20s). | 1 |
| MEMORY COMMANDS | | | | |
| Read Scratchpad | Reads the entire scratchpad including the CRC byte. | BEh | DS18S20 transmits up to 9 data bytes to master. | 2 |
| Write Scratchpad | Writes data into scratchpad bytes 2 and 3 (T_H and T_L). | 4Eh | Master transmits 2 data bytes to DS18S20. | 3 |
| Copy Scratchpad | Copies T_H and T_L data from the scratchpad to EEPROM. | 48h | None | 1 |
| Recall E ² | Recalls T_H and T_L data from EEPROM to the scratchpad. | B8h | DS18S20 transmits recall status to master. | |
| Read Power Supply | Signals DS18S20 power supply mode to the master. | B4h | DS18S20 transmits supply status to master. | |

NOTES:

1. For parasite-powered DS18S20s, the master must enable a strong pullup on the 1-wire bus during temperature conversions and copies from the scratchpad to EEPROM. No other bus activity may take place during this time.
2. The master can interrupt the transmission of data at any time by issuing a reset.
3. Both bytes must be written before a reset is issued.

1-WIRE SIGNALING

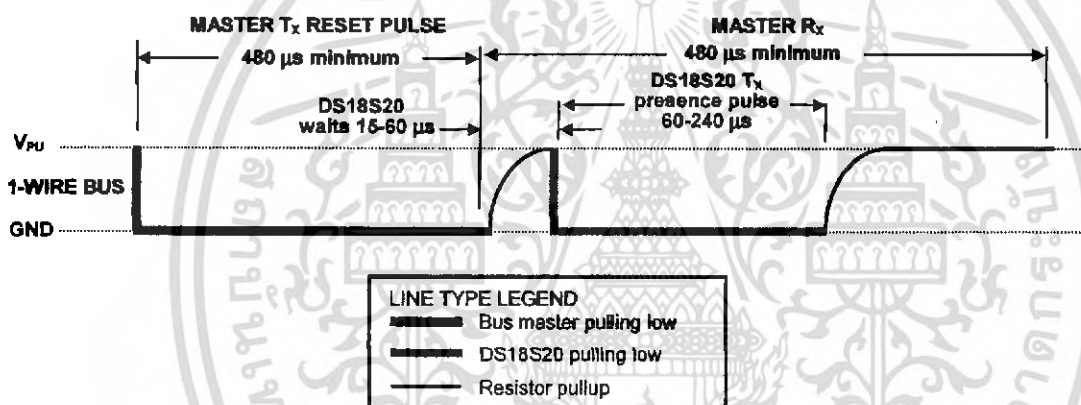
The DS18S20 uses a strict 1-wire communication protocol to insure data integrity. Several signal types are defined by this protocol: reset pulse, presence pulse, write 0, write 1, read 0, and read 1. All of these signals, with the exception of the presence pulse, are initiated by the bus master.

INITIALIZATION PROCEDURE: RESET AND PRESENCE PULSES

All communication with the DS18S20 begins with an initialization sequence that consists of a reset pulse from the master followed by a presence pulse from the DS18S20. This is illustrated in Figure 10. When the DS18S20 sends the presence pulse in response to the reset, it is indicating to the master that it is on the bus and ready to operate.

During the initialization sequence the bus master transmits (T_x) the reset pulse by pulling the 1-wire bus low for a minimum of 480 μs . The bus master then releases the bus and goes into receive mode (R_x). When the bus is released, the 5k pullup resistor pulls the 1-wire bus high. When the DS18S20 detects this rising edge, it waits 15–60 μs and then transmits a presence pulse by pulling the 1-wire bus low for 60–240 μs .

INITIALIZATION TIMING Figure 10



READ/WRITE TIME SLOTS

The bus master writes data to the DS18S20 during write time slots and reads data from the DS18S20 during read time slots. One bit of data is transmitted over the 1-wire bus per time slot.

WRITE TIME SLOTS

There are two types of write time slots: “Write 1” time slots and “Write 0” time slots. The bus master uses a Write 1 time slot to write a logic 1 to the DS18S20 and a Write 0 time slot to write a logic 0 to the DS18S20. All write time slots must be a minimum of 60 μs in duration with a minimum of a 1 μs recovery time between individual write slots. Both types of write time slots are initiated by the master pulling the 1-wire bus low (see Figure 11).

To generate a Write 1 time slot, after pulling the 1-wire bus low, the bus master must release the 1-wire bus within 15 μs . When the bus is released, the 5k pullup resistor will pull the bus high. To generate a Write 0 time slot, after pulling the 1-wire bus low, the bus master must continue to hold the bus low for the duration of the time slot (at least 60 μs). The DS18S20 samples the 1-wire bus during a window that lasts from 15 μs to 60 μs after the master initiates the write time slot. If the bus is high during the sampling window, a 1 is written to the DS18S20. If the line is low, a 0 is written to the DS18S20.

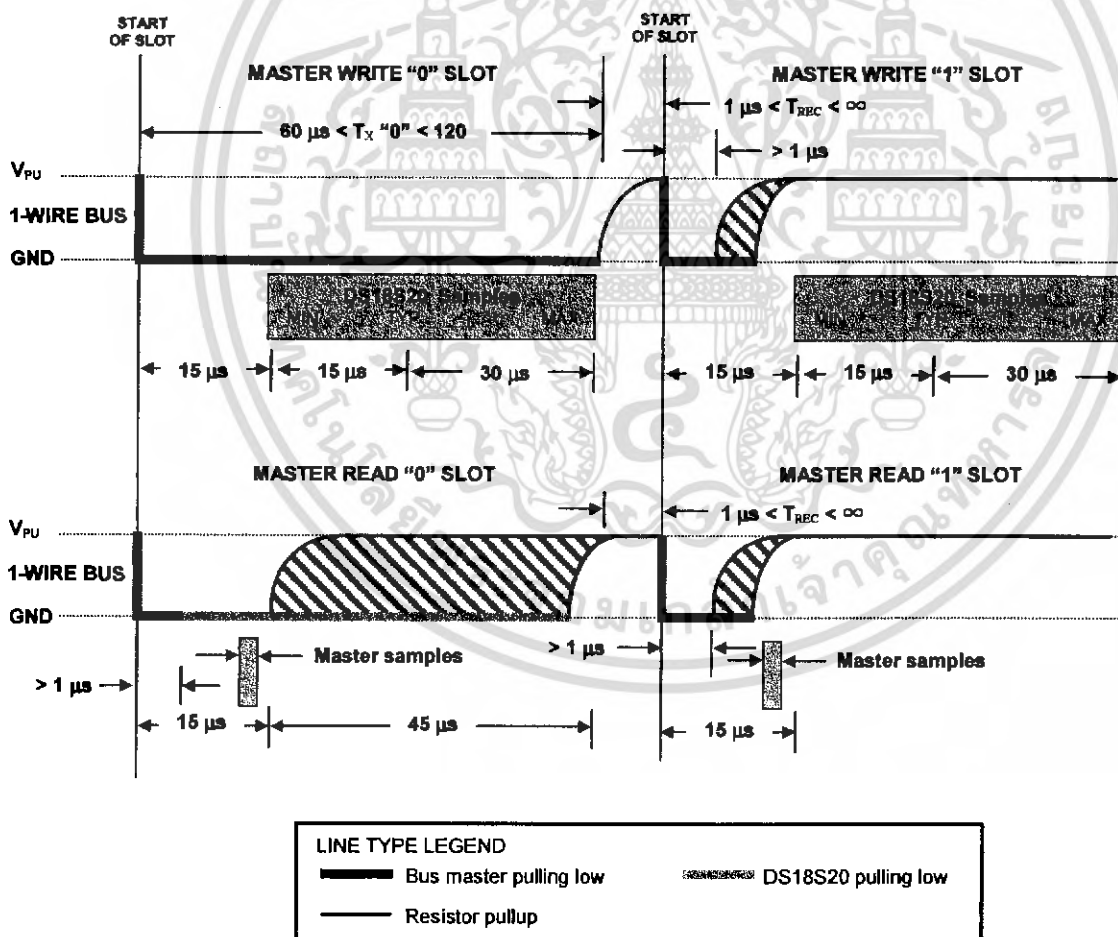
READ TIME SLOTS

The DS18S20 can only transmit data to the master when the master issues read time slots. Therefore, the master must generate read time slots immediately after issuing a Read Scratchpad [BEh] or Read Power Supply [B4h] command, so that the DS18S20 can provide the requested data. In addition, the master can generate read time slots after issuing Convert T [44h] or Recall E² [B8h] commands to find out the status of the operation as explained in the DS18S20 FUNCTION COMMAND section.

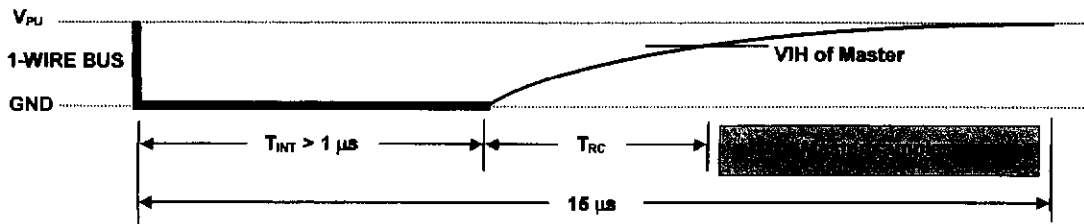
All read time slots must be a minimum of 60 μs in duration with a minimum of a 1 μs recovery time between slots. A read time slot is initiated by the master device pulling the 1-wire bus low for a minimum of 1 μs and then releasing the bus (see Figure 11). After the master initiates the read time slot, the DS18S20 will begin transmitting a 1 or 0 on bus. The DS18S20 transmits a 1 by leaving the bus high and transmits a 0 by pulling the bus low. When transmitting a 0, the DS18S20 will release the bus by the end of the time slot, and the bus will be pulled back to its high idle state by the pullup resistor. Output data from the DS18S20 is valid for 15 μs after the falling edge that initiated the read time slot. Therefore, the master must release the bus and then sample the bus state within 15 μs from the start of the slot.

Figure 12 illustrates that the sum of T_{INIT} , T_{RC} , and T_{SAMPLE} must be less than 15 μs for a read time slot. Figure 13 shows that system timing margin is maximized by keeping T_{INIT} and T_{RC} as short as possible and by locating the master sample time during read time slots towards the end of the 15 μs period.

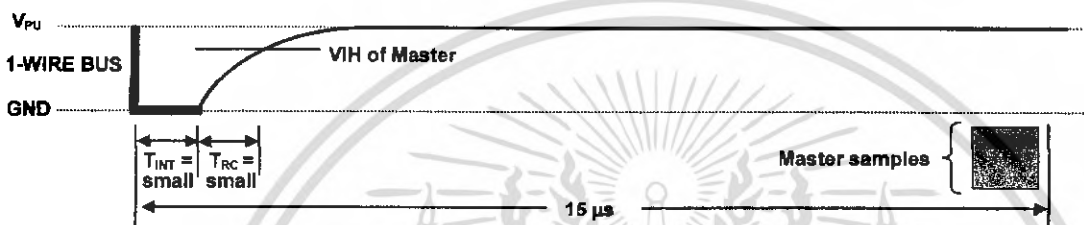
READ/WRITE TIME SLOT TIMING DIAGRAM Figure 11



DETAILED MASTER READ 1 TIMING Figure 12

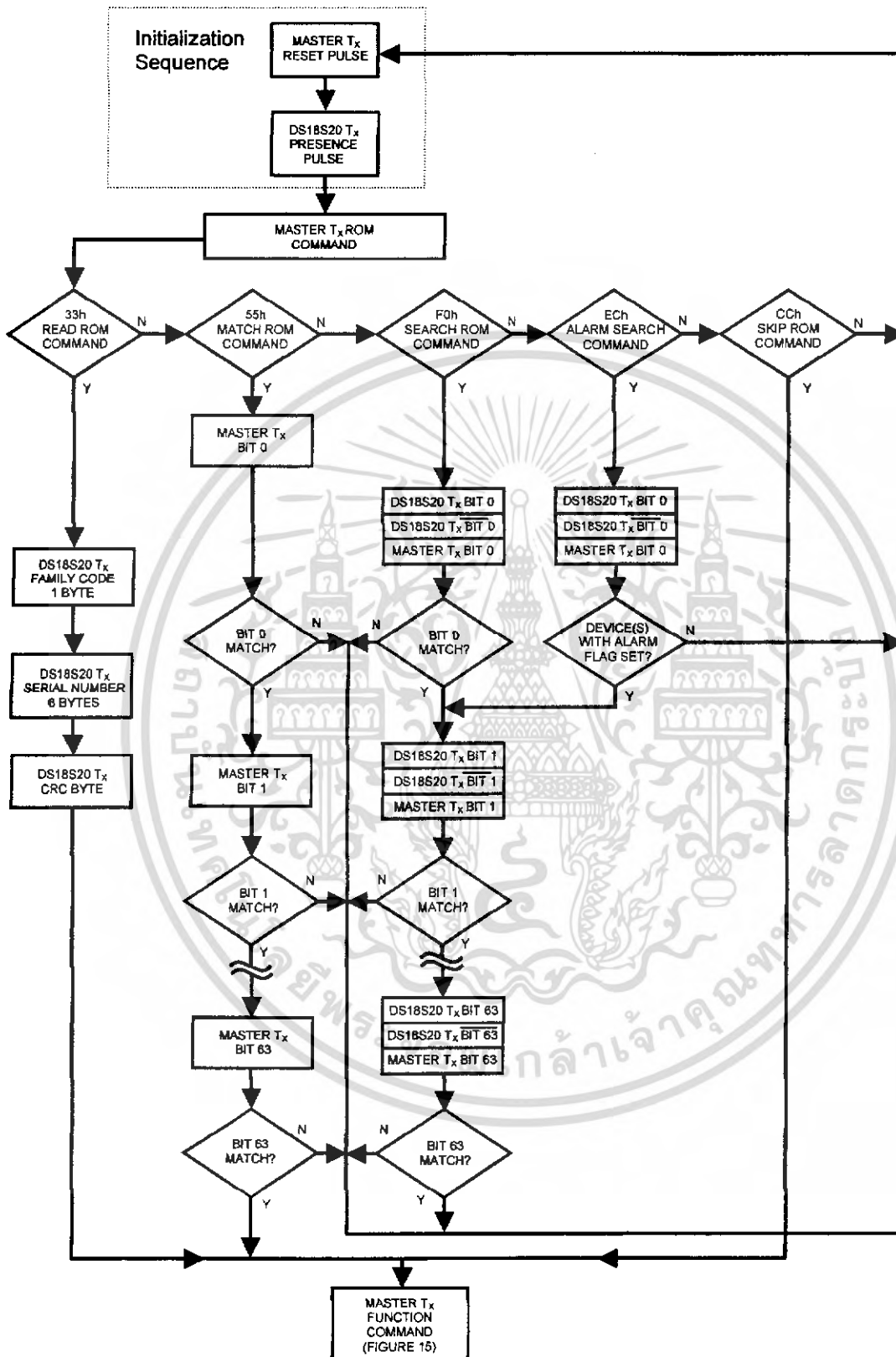


RECOMMENDED MASTER READ 1 TIMING Figure 13



LINE TYPE LEGEND
 Bus master pulling low
 Resistor pullup

ROM COMMANDS FLOW CHART Figure 14



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DS18S20 OPERATION EXAMPLE 1

In this example there are multiple DS18S20s on the bus and they are using parasite power. The bus master initiates a temperature conversion in a specific DS18S20 and then reads its scratchpad and recalculates the CRC to verify the data.

| MASTER MODE | DATA (LSB FIRST) | COMMENTS |
|-------------|------------------------------------|---|
| TX | Reset | Master issues reset pulse. |
| RX | Presence | DS18S20s respond with presence pulse. |
| TX | 55h | Master issues Match ROM command. |
| TX | 64-bit ROM code | Master sends DS18S20 ROM code. |
| TX | 44h | Master issues Convert T command. |
| TX | DQ line held high by strong pullup | Master applies strong pullup to DQ for the duration of the conversion (t_{conv}). |
| TX | Reset | Master issues reset pulse. |
| RX | Presence | DS18S20s respond with presence pulse. |
| TX | 55h | Master issues Match ROM command. |
| TX | 64-bit ROM code | Master sends DS18S20 ROM code. |
| TX | BEh | Master issues Read Scratchpad command. |
| RX | 9 data bytes | Master reads entire scratchpad including CRC. The master then recalculates the CRC of the first eight data bytes from the scratchpad and compares the calculated CRC with the read CRC (byte 9). If they match, the master continues; if not, the read operation is repeated. |

DS18S20 OPERATION EXAMPLE 2

In this example there is only one DS18S20 on the bus and it is using parasite power. The master writes to the T_H and T_L registers in the DS18S20 scratchpad and then reads the scratchpad and recalculates the CRC to verify the data. The master then copies the scratchpad contents to EEPROM.

| MASTER MODE | DATA (LSB FIRST) | COMMENTS |
|-------------|------------------------------------|---|
| TX | Reset | Master issues reset pulse. |
| RX | Presence | DS18S20 responds with presence pulse. |
| TX | CCh | Master issues Skip ROM command. |
| TX | 4Eh | Master issues Write Scratchpad command. |
| TX | 2 data bytes | Master sends two data bytes to scratchpad (T_H and T_L) |
| TX | Reset | Master issues reset pulse. |
| RX | Presence | DS18S20 responds with presence pulse. |
| TX | CCh | Master issues Skip ROM command. |
| TX | BEh | Master issues Read Scratchpad command. |
| RX | 9 data bytes | Master reads entire scratchpad including CRC. The master then recalculates the CRC of the first eight data bytes from the scratchpad and compares the calculated CRC with the read CRC (byte 9). If they match, the master continues; if not, the read operation is repeated. |
| TX | Reset | Master issues reset pulse. |
| RX | Presence | DS18S20 responds with presence pulse. |
| TX | CCh | Master issues Skip ROM command. |
| TX | 48h | Master issues Copy Scratchpad command. |
| TX | DQ line held high by strong pullup | Master applies strong pullup to DQ for at least 10 ms while copy operation is in progress. |

DS18S20 OPERATION EXAMPLE 3

In this example there is only one DS18S20 on the bus and it is using parasite power. The bus master initiates a temperature conversion then reads the DS18S20 scratchpad and calculates a higher resolution result using the data from the temperature, COUNT REMAIN and COUNT PER °C registers.

| MASTER MODE | DATA (LSB FIRST) | COMMENTS |
|-------------|------------------------------------|--|
| TX | Reset | Master issues reset pulse. |
| TR | Presence | DS18S20 responds with presence pulse. |
| TX | CCh | Master issues Skip ROM command. |
| TX | 44h | Master issues Convert T command. |
| TX | DQ line held high by strong pullup | Master applies strong pullup to DQ for the duration of the conversion (t_{conv}). |
| TX | Reset | Master issues reset pulse. |
| RX | Presence | DS18S20 responds with presence pulse. |
| TX | CCh | Master issues Skip ROM command. |
| TX | BEh | Master issues Read Scratchpad command. |
| RX | 9 data bytes | Master reads entire scratchpad including CRC. The master then recalculates the CRC of the first eight data bytes from the scratchpad and compares the calculated CRC with the read CRC (byte 9). If they match, the master continues; if not, the read operation is repeated. The master also calculates the TEMP_READ value and stores the contents of the COUNT REMAIN and COUNT PER °C registers. |
| TX | Reset | Master issues reset pulse. |
| RX | Presence | DS18S20 responds with presence pulse. |
| - | - | CPU calculates extended resolution temperature using the equation in the OPERATION - MEASURING TEMPERATURE section of this datasheet. |

RELATED APPLICATION NOTES

The following Application Notes can be applied to the DS18S20. These notes can be obtained from the Dallas Semiconductor "Application Note Book," via the Dallas website at <http://www.dalsemi.com/>, or through our faxback service at (214) 450-0441.

Application Note 27: "Understanding and Using Cyclic Redundancy Checks with Dallas Semiconductor Touch Memory Product"

Application Note 55: "Extending the Contact Range of Touch Memories"

Application Note 74: "Reading and Writing Touch Memories via Serial Interfaces"

Application Note 104: "Minimalist Temperature Control Demo"

Application Note 105: "High Resolution Temperature Measurement with Dallas Direct-to-Digital Temperature Sensors"

Application Note 106: "Complex MicroLANs"

Application Note 108: "MicroLAN – In the Long Run"

Sample 1-wire subroutines that can be used in conjunction with AN74 can be downloaded from the Dallas website or anonymous FTP Site.

ABSOLUTE MAXIMUM RATINGS*

| | |
|---------------------------------------|------------------------------|
| Voltage on any pin relative to ground | -0.5V to +6.0V |
| Operating temperature | -55°C to +125°C |
| Storage temperature | -55°C to +125°C |
| Soldering temperature | See J-STD-020A Specification |

*These are stress ratings only and functional operation of the device at these or any other conditions above those indicated in the operation sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods of time may affect reliability.

DC ELECTRICAL CHARACTERISTICS (-55°C to +125°C; $V_{DD}=3.0V$ to 5.5V)

| PARAMETER | SYMBOL | CONDITION | MIN | TYP | MAX | UNITS | NOTES |
|-----------------------|-----------|-----------------|------|------|---|-------|-------|
| Supply Voltage | V_{DD} | Local Power | +3.0 | | +5.5 | V | 1 |
| Pullup Supply Voltage | V_{PU} | Parasite Power | +3.0 | | +5.5 | V | 1,2 |
| | | Local Power | +3.0 | | V_{DD} | | |
| Thermometer Error | t_{ERR} | -10°C to +85°C | | | ±0.5 | °C | 3 |
| | | -55°C to +125°C | | | ±2 | | |
| Input Logic Low | V_{IL} | | -0.3 | | +0.8 | V | 1,4,5 |
| Input Logic High | V_{IH} | Local Power | +2.2 | | The lower of 5.5 or $V_{DD} + 0.3$ | V | 1, 6 |
| | | Parasite Power | +3.0 | | | | |
| Sink Current | I_L | $V_{IO}=0.4V$ | 4.0 | | | mA | 1 |
| Standby Current | I_{DDs} | | | 750 | 1000 | nA | 7,8 |
| Active Current | I_{DD} | $V_{DD}=5V$ | | 1 | 1.5 | mA | 9 |
| DQ Input Current | I_{DQ} | | | 5 | | µA | 10 |
| Drift | | | | ±0.2 | | °C | 11 |

NOTES:

- All voltages are referenced to ground.
- The Pullup Supply Voltage specification assumes that the pullup device is ideal, and therefore the high level of the pullup is equal to V_{PU} . In order to meet the V_{IH} spec of the DS18S20, the actual supply rail for the strong pullup transistor must include margin for the voltage drop across the transistor when it is turned on; thus: $V_{PU_ACTUAL} = V_{PU_IDEAL} + V_{TRANSISTOR}$.
- See typical performance curve in Figure 16
- Logic low voltages are specified at a sink current of 4 mA.
- To guarantee a presence pulse under low voltage parasite power conditions, V_{ILMAX} may have to be reduced to as low as 0.5V.
- Logic high voltages are specified at a source current of 1 mA.
- Standby current specified up to 70°C. Standby current typically is 3 µA at 125°C.
- To minimize I_{DDs} , DQ should be within the following ranges: $GND \leq DQ \leq GND + 0.3V$ or $V_{DD} - 0.3V \leq DQ \leq V_{DD}$.
- Active current refers to supply current during active temperature conversions or EEPROM writes.
- DQ line is high ("hi-Z" state).
- Drift data is based on a 1000 hour stress test at 125°C with $V_{DD} = 5.5V$.

AC ELECTRICAL CHARACTERISTICS: NV MEMORY(-55°C to +100°C; $V_{DD}=3.0V$ to 5.5V)

| PARAMETER | SYMBOL | CONDITION | MIN | TYP | MAX | UNITS |
|-----------------------|------------|----------------|-----|-----|-----|--------|
| NV Write Cycle Time | t_{WT} | | | 2 | 10 | ms |
| EEPROM Writes | N_{EEWR} | -55°C to +55°C | 50k | | | writes |
| EEPROM Data Retention | t_{EEDR} | -55°C to +55°C | 10 | | | years |

AC ELECTRICAL CHARACTERISTICS (-55°C to +125°C; $V_{DD}=3.0V$ to 5.5V)

| PARAMETER | SYMBOL | CONDITION | MIN | TYP | MAX | UNITS | NOTES |
|-----------------------------|--------------|--------------------------------|-----|-----|-----|---------|-------|
| Temperature Conversion Time | t_{CONV} | | | | 750 | ms | 1 |
| Time to Strong Pullup On | t_{SPON} | Start Convert T Command Issued | | | 10 | μs | |
| Time Slot | t_{SLOT} | | 60 | | 120 | μs | 1 |
| Recovery Time | t_{REC} | | 1 | | | μs | 1 |
| Write 0 Low Time | t_{LOW0} | | 60 | | 120 | μs | 1 |
| Write 1 Low Time | t_{LOW1} | | 1 | | 15 | μs | 1 |
| Read Data Valid | t_{RDV} | | | | 15 | μs | 1 |
| Reset Time High | t_{RSTH} | | 480 | | | μs | 1 |
| Reset Time Low | t_{RSTL} | | 480 | | | μs | 1,2 |
| Presence Detect High | t_{PDHIGH} | | 15 | | 60 | μs | 1 |
| Presence Detect Low | t_{PDLOW} | | 60 | | 240 | μs | 1 |
| Capacitance | $C_{IN/OUT}$ | | | | 25 | pF | |

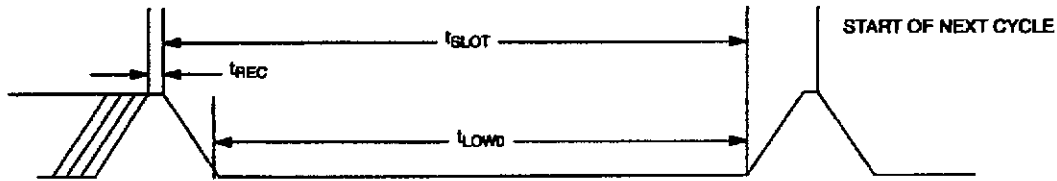
NOTES:

1. Refer to timing diagrams in Figure 17.
2. Under parasite power, if $t_{RSTL} > 960 \mu s$, a power on reset may occur.

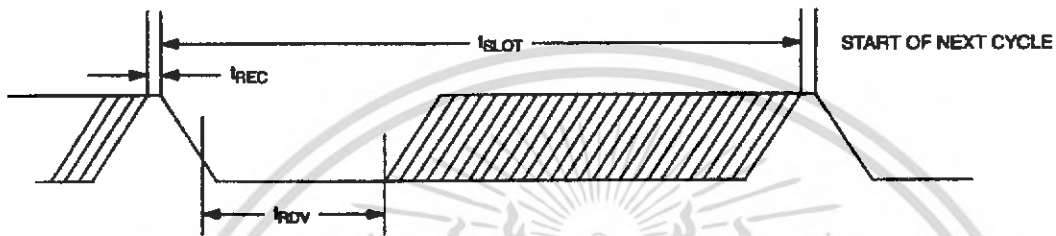
TYPICAL PERFORMANCE CURVE Figure 16

TIMING DIAGRAMS Figure 17

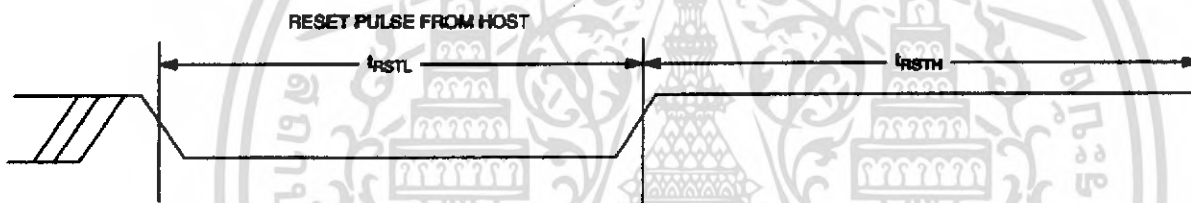
1-WIRE WRITE ZERO TIME SLOT



1-WIRE READ ZERO TIME SLOT



1-WIRE RESET PULSE



1-WIRE PRESENCE DETECT

