สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบต้อนรับผู้มาเยือน

VISITOR RECEPTION SYSTEM

โดย

นายศิริพงษ์ จันทรี

นางสาวสุนิสา สุวรรณวัฒนะ

นางสาวสุมณฑารถ ชาวเรือ

ปพ.
ส 462 ร
2547
เลขหมู่.....
เลขทะเบียน..... 61967
วัน,เดือน,ปี..... 25 ม.ค. 2549
b.....
i.....

อาจารย์ที่ปรึกษา

ดร.พิพัฒน์ พรหมมี

รศ.สมยศ จุณณะปิยะ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

ภาควิชา
วิศวกรรมโทรคมนาคม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบต้อนรับผู้มาเยือน

VISITOR RECEPTION SYSTEM

โดย

นายศิริพงษ์ จันทรี 44010485

นางสาวสุนิสา สุวรรณวัฒนะ 44010541

นางสาวสุมณฑารถ ชาวเรือ 44010547

อาจารย์ที่ปรึกษา

ดร.พิพัฒน์ พรหมมี

รศ.สมยศ จุณณะปิยะ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2547

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง **ระบบต้อนรับผู้มาเยือน**

VISITOR RECEPTION SYSTEM

ผู้จัดทำ

1.นาย ศิริพงษ์ จันทรี 44010485

2.นางสาว สุนิสา สุวรรณวัฒนะ 44010541

3.นางสาว สุมณมารถ ชาวเรือ 44010547

_____ อาจารย์ที่ปรึกษา

(ดร.พิพัฒน์ พรหมมี)

_____ อาจารย์ที่ปรึกษา

(รศ.สมยศ จุณณะปิยะ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบต้อนรับผู้มาเยือน

Visitor Reception System

โดย นาย ศิริพงษ์ จันทรี 44010485

นางสาว สุนิสา สุวรรณวัฒนะ 44010541

นางสาว สุมณมารถ ชาวเรือ 44010547

อาจารย์ที่ปรึกษา ดร.พิพัฒน์ พรหมมี

รศ.สมยศ จุณณะปิยะ

บทคัดย่อ

โครงงานนี้เป็นการนำเสนอระบบอำนวยความสะดวกภายในบ้าน กรณีมีผู้มาติดต่อขณะที่เจ้าของบ้านไม่อยู่ เมื่อกดกริ่งหน้าบ้าน กล้องโทรทัศน์วงจรปิดที่ติดตั้งไว้จะถ่ายรูปแล้วส่งข้อมูลผ่านคอมพิวเตอร์ไปยังโทรศัพท์เคลื่อนที่ของเจ้าของบ้านพร้อมกันกับหมายเลขโทรศัพท์สำหรับติดต่อกลับ โดยใช้หลักการของ MMS

Abstract

This project presents a visitor reception system while a visitor comes home and nobody's around. After the visitor pressing the ring bell, the camera which has been installed will capture his picture to the computer and send it to the host's mobile phone, using cellular MMS function. Moreover, the visitor can also leave his telephone number to the system and it will be sent together with the picture file.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทคัดย่อ

ABTRACT

หน้า

บทที่ 1 บทนำ 1

บทที่ 2 ทฤษฎีและหลักการ 2

2.1 ส่วนของการตอบรับและการฝากหมายเลขโทรศัพท์ 2

2.1.1 คุณสมบัติของ ISD-2560 2

2.1.2 โครงสร้างภายในของ ISD-2560 2

2.1.3 การบันทึกข้อมูล (Recording) 4

2.1.4 การเล่นกลับ (Play Back) 5

2.1.5 รายละเอียดของขาอุปกรณ์ของ ISD-2560 6

2.2 ทฤษฎีของไมโครคอนโทรลเลอร์ตระกูล MCS-51 9

2.2.1 คุณสมบัติทั่วไปของไมโครคอนโทรลเลอร์ตระกูล MCS-51 9

2.2.2 โครงสร้างภายนอกไมโครคอนโทรลเลอร์ตระกูลของ MCS-51 11

2.2.3 โครงสร้างภายในของ MCS-51 14

2.2.4 การจัดหน่วยความจำ 15

2.2.4.1 หน่วยความจำโปรแกรม 16

2.2.4.2 หน่วยความจำข้อมูล 16

2.2.4.3 รีจิสเตอร์ใช้งานทั่วไป 16

2.2.4.4 รีจิสเตอร์หน้าที่พิเศษ (SFR) 16

2.2.5 การใช้งานรีจิสเตอร์ 18

2.2.6 การเชื่อมต่อไมโครคอนโทรลเลอร์ตระกูล 8051 19

กับคอมพิวเตอร์ผ่านพอร์ตอนุกรม

2.2.6.1 การสื่อสารแบบอนุกรม 19

2.2.6.2 การสื่อสารข้อมูลแบบอะซิงโครนัส 20

2.2.6.3 มาตรฐานพอร์ตอนุกรมแบบ RS-232 22

2.2.6.4 คอนเน็กเตอร์สำหรับพอร์ต RS-232 และการเชื่อมต่อ 22

2.2.6.5 UART 25

2.2.6.6 MCS-51 กับการรับส่งข้อมูลแบบอนุกรม 26

2.3 การเชื่อมต่อกับคีย์แพด (Keypad) 33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.4 VC++.NET 34

2.4.1 การจัดเก็บข้อมูล 35

2.5 ความรู้เบื้องต้นเกี่ยวกับจดหมายอิเล็กทรอนิกส์ 36

2.6 ความรู้เบื้องต้นของโปรโตคอล SMTP (Simple Mail Transfer Protocol) 37

2.6.1 วิธีการทำงานของโปรโตคอล SMTP 37

2.6.2 การรับส่งจดหมายอิเล็กทรอนิกส์ระหว่างเครื่องโดยใช้โปรโตคอล SMTP 37

2.7 ความรู้เบื้องต้นของ MIME (Multipurpose Internet Mail Extensions) 38

2.7.1 วิธีการทำงานของ MIME 38

2.7.2 การเข้ารหัส MIME 39

2.8 MMS (Multimedia Messaging Services) 41

2.8.1 รูปแบบการให้บริการ MMS 41

2.8.2 โครงสร้างและการทำงานของเทคโนโลยี MMS 43

บทที่ 3 การคำนวณและการสร้าง **46**

3.1 บล็อกไดอะแกรมการทำงานของระบบต้อนรับผู้มาเยือน 46

3.2 การทำงานของระบบต้อนรับผู้มาเยือน 47

3.2.1 การทำงานในส่วนที่1 47

อุปกรณ์ฮาร์ดแวร์และไมโครคอนโทรลเลอร์ MCS-51

3.2.2 การทำงานในส่วนที่ 2 49

โปรแกรมที่พัฒนาโดยใช้ VC++.NET และฐานข้อมูล

3.2.3 การทำงานในส่วนที่ 3 เซิร์ฟเวอร์ของผู้ให้บริการ 50

3.3 โฟลว์ชาร์ตการทำงานของไมโครคอนโทรลเลอร์ 50

3.4 โฟลว์ชาร์ตการทำงานของ VC++.NET 52

บทที่4 ผลการทดลอง **53**

4.1 ผลการทดลองส่วนของอุปกรณ์ฮาร์ดแวร์ 53

4.2 ผลการทดลองในส่วนของไมโครคอนโทรลเลอร์ 54

4.2.1 ผลการทดลองในส่วนของการแสดงผลที่ได้จากพอร์ตอนุกรม 54

4.2.2 ผลการทดลองในส่วนระบบฐานข้อมูล 54

4.2.2.1 การทำงานในส่วนของการรับและบันทึกข้อมูล 55

4.2.2.2 การทำงานในส่วนของการเรียกดูรายละเอียดของข้อมูล 55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

4.2.3 ผลการทดลองในส่วนของการจับภาพ 56

4.3 การทำงานในส่วนของการส่งจดหมายอิเล็กทรอนิกส์ 57

4.3.1 ผลการทดลองส่วนของผู้ให้บริการและการรับข้อมูลของโทรศัพท์เคลื่อนที่ 57

4.3.1.1 ผลการทดลองเมื่อส่งไปยังเซิร์ฟเวอร์ของ 57
บริษัท แอดวานซ์อินโฟเซอร์วิส จำกัด

4.3.1.2 ผลการทดลองเมื่อส่งไปยังเซิร์ฟเวอร์ของ 58
บริษัท โทเทิลแอคเซสคอมมูนิเคชั่น จำกัด

บทที่ 5 บทวิจารณ์และสรุปผล **59**

5.1 สรุปผลการทดลอง 59

5.1.1 การทำงานในส่วนของอุปกรณ์ฮาร์ดแวร์ 59

5.1.2 การทำงานในส่วนของ VC++.NET 59

5.1.3 การทำงานในส่วนของผู้ให้บริการและการรับข้อมูลของโทรศัพท์เคลื่อนที่ 59

5.2 ปัญหาที่พบในการทดลอง 59

ภาคผนวก

หนังสืออ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

หน้า

รูปที่ 2.1 บล็อกไดอะแกรมภายในของ ISD-2560 3
รูปที่ 2.2 การจัดตำแหน่งขาต่างๆ ของไอซี ISD-2560 9
รูปที่ 2.3 การจัดตำแหน่งขาต่างๆ ของไมโครคอนโทรลเลอร์ตระกูล MCS-51 11
รูปที่ 2.4 โครงสร้างภายในของไมโครคอนโทรลเลอร์ตระกูล MCS-51 14
รูปที่ 2.5 การจัดโครงสร้างของหน่วยความจำทั้งในส่วนของ 15
หน่วยความจำโปรแกรมและหน่วยความจำข้อมูล
รูปที่ 2.6 การจัดหน่วยความจำและตำแหน่งรีจิสเตอร์หน้าที่พิเศษต่างๆ 17
รูปที่ 2.7 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรม 19
รูปที่ 2.8 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรมแบบอะซิงโครนัส 20
รูปที่ 2.9 การจัดขาของคอนเน็กเตอร์พอร์ตอนุกรมตามมาตรฐาน RS-232 22
ทั้งแบบ DB-9 และ DB-25
รูปที่ 2.10 การต่ออุปกรณ์ภายนอกกับพอร์ตอนุกรมของคอมพิวเตอร์ในลักษณะต่างๆ 24
รูปที่ 2.11 รีจิสเตอร์ควบคุมการทำงานของพอร์ตอนุกรม 27
รูปที่ 2.12 ผังการทำงานในโหมด 1 ของพอร์ตอนุกรมภายในไมโครคอนโทรลเลอร์ MCS-51 29
รูปที่ 2.13 รายละเอียดเบื้องต้นของไอซีแปลงสัญญาณเพื่อเชื่อมต่อ 32
กับพอร์ตอนุกรมของคอมพิวเตอร์
รูปที่ 2.14 วงจรเชื่อมต่อ MAX232 หรือ ICL232 33
เข้ากับพอร์ตอนุกรมของคอมพิวเตอร์และไมโครคอนโทรลเลอร์ MCS-51
รูปที่ 2.15 วงจรของสวิตช์แบบเมตริกซ์หรือคีย์แพด 34
รูปที่ 2.16 แสดงโครงสร้างข้อมูลที่นำมาประมวลผลด้วยคอมพิวเตอร์ 35
รูปที่ 2.17 ส่วนประกอบของการให้บริการแบบ MMS 43

รูปที่ 3.1 บล็อกไดอะแกรมรวมของโครงงานระบบต้อนรับผู้มาเยือน 46
รูปที่ 3.2 การทำงานของระบบต้อนรับผู้มาเยือน 47
รูปที่ 3.3 วงจรบันทึกเสียง 48
รูปที่ 3.4 วงจรขยายกำลัง 48
รูปที่ 3.5 วงจรรวมของการเชื่อมต่อของอุปกรณ์ต่างๆกับไมโครคอนโทรลเลอร์ 49
และไมโครคอนโทรลเลอร์กับพอร์ตอนุกรม
รูปที่ 3.6 โฟลว์ชาร์ตแสดงการทำงานของอุปกรณ์ฮาร์ดแวร์ 51
ซึ่งควบคุมโดยไมโครคอนโทรลเลอร์ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

หน้า

รูปที่ 3.7 โฟลว์ชาร์ตแสดงการทำงานในส่วนของการถ่ายภาพ และการเก็บหมายเลขโทรศัพท์ลงในฐานข้อมูล 52

รูปที่ 4.1 การเชื่อมต่ออุปกรณ์กับไมโครคอนโทรลเลอร์ 53
รูปที่ 4.2 อุปกรณ์พร้อมติดตั้งเพื่อใช้งาน 53
รูปที่ 4.3 รูปแบบหน้าจอแสดงผลในการรับและแสดงข้อมูลของผู้มาเยือน 54
รูปที่ 4.4 การรับข้อมูลของระบบฐานข้อมูลและบันทึกข้อมูล 55
รูปที่ 4.5 การค้นหาดูข้อมูลโดยหมายเลขโทรศัพท์ 55
รูปที่ 4.6 การค้นหาดูข้อมูลโดยชื่อภาพ 56
รูปที่ 4.7 การถ่ายภาพของกล้องโทรทัศน์วงจรปิด 56
รูปที่ 4.8 รูปภาพที่ได้จากการจับภาพของการถ่ายภาพจากกล้องโทรทัศน์วงจรปิด โดยการกดปุ่ม Capture ภาพที่ได้จะถูกเก็บไว้ที่ Local Disk (C:) 57
รูปที่ 4.9 การแสดงข้อความมัลติมีเดียที่โทรศัพท์เคลื่อนที่ 57
รูปที่ 4.10 การแสดงข้อความสั้นแจ้งเตือนที่โทรศัพท์เคลื่อนที่ 58
รูปที่ 4.11 ผลการรับจดหมายอิเล็กทรอนิกส์ของ http://dtac.co.th 58

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 2.1 วิธีการควบคุมการทำงานของ ISD2560 ในการทำงานแบบการเล่นกลับโดยไม่เรียงตามลำดับการบันทึก 6

ตารางที่ 2.2 คุณสมบัติของไมโครคอนโทรลเลอร์แต่ละเบอร์ในตระกูล MCS-51 11

ตารางที่ 2.3 หน้าที่พิเศษของแต่ละขาของพอร์ต P_3 12

ตารางที่ 2.4 แสดงบิตพาริตี้ของข้อมูล 21

ตารางที่ 2.5 การเลือกอัตราพอร์ตของวงจรพอร์ตอนุกรมภายในไมโครคอนโทรลเลอร์ MCS-51 31

ตารางที่ 2.6 ตัวอย่างแฟ้มข้อมูลชื่อ นามสกุล ที่อยู่และหมายเลขโทรศัพท์ 35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่1
บทนำ

เทคโนโลยีและศาสตร์ต่างๆ ในโลกล้วนถูกคิดค้นขึ้นมาเพื่อตอบสนองความต้องการของมนุษย์ ไม่ว่าจะเป็นเพื่อตอบสนองความต้องการพื้นฐาน หรือเพื่ออำนวยความสะดวกสบายในชีวิตประจำวัน รวมไปถึงเพื่อผลประโยชน์ทางด้านธุรกิจ ด้วยเหตุนี้เอง เทคโนโลยีจึงถูกพัฒนาไปอย่างรวดเร็ว ในโลกที่การติดต่อสื่อสารเป็นสิ่งจำเป็น มนุษย์ต้องกระทำเกือบตลอดเวลา เป็นเหตุให้มีการพัฒนาเทคโนโลยีด้านการสื่อสารให้ก้าวหน้าไปอย่างรวดเร็วด้วยเช่นกัน

โครงงานนี้ เป็นการนำเทคโนโลยีทางด้านการสื่อสารข้อมูลมาใช้พัฒนาระบบสื่อสารอีกรูปแบบหนึ่งเพื่อนำมาประยุกต์ใช้อำนวยความสะดวกในการติดต่อสื่อสารระหว่างเจ้าบ้านและแขกผู้มาติดต่อ โดยจะเป็นรูปแบบระบบรับฝากข้อมูลของแขกผู้มาติดต่อ และแจ้งเตือนให้เจ้าบ้านได้รับทราบผ่านทางระบบโทรศัพท์เคลื่อนที่ เพื่อจะได้ทำการติดต่อกลับไปยังผู้มาเยือนต่อไป

โดยโครงงานนี้ได้ประยุกต์ใช้กล้องโทรทัศน์วงจรปิด, ไมโครคอนโทรลเลอร์ตระกูล MCS-51, วงจรบันทึกเสียง, การพัฒนาโปรแกรมโดยใช้ภาษา Visual C++.NET รวมไปถึงการรับส่งข้อมูลแบบข้อความมัลติมีเดีย (MMS) มาทำงานร่วมกันอย่างเป็นระบบ การทำงานของระบบจะเริ่มเมื่อมีการกดกริ่งไฟฟ้าขึ้น โปรแกรมควบคุมจะสั่งให้กล้องโทรทัศน์วงจรปิดที่ถูกติดตั้งไว้ทำการจับภาพนิ่งของแขกผู้มาติดต่อในขณะนั้นและส่งข้อมูลภาพไปเก็บไว้ยังฐานข้อมูล ในขณะเดียวกัน ส่วนที่ติดต่อกับแขกผู้มาติดต่อจะรับข้อมูลหมายเลขโทรศัพท์จากคีย์แพด (keypad) แล้วส่งไปเก็บไว้ยังฐานข้อมูล โดยกำหนดให้มีการจัดเก็บให้มีความสัมพันธ์กับข้อมูลรูปภาพ หลังจากนั้น ระบบจะส่งข้อมูลรูปภาพและข้อมูลหมายเลขโทรศัพท์ไปยังโทรศัพท์ของเจ้าบ้านโดยผ่านทางระบบ MMS

ในโลกที่กำลังพัฒนาอย่างไม่หยุดยั้ง การติดต่อสื่อสารมีความสำคัญขึ้นทุกขณะ ทุกคนย่อมไม่อยากพลาดการติดต่อสื่อสารไป โดยเฉพาะหน่วยงานทางธุรกิจต่างๆที่หากพลาดการติดต่อสื่อสารอาจหมายถึงการพลาดโอกาสทางธุรกิจไปอย่างน่าเสียดาย โครงงาน “ระบบต้อนรับผู้มาเยือน” (Visitor Reception System) จึงเป็นเครื่องมือหนึ่งที่สามารถรองรับความต้องการด้านนี้ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่2
ทฤษฎีและหลักการ

2.1 ส่วนของการตอบรับและการฝากหมายเลขโทรศัพท์

ในส่วนของการตอบรับและฝากหมายเลขโทรศัพท์นั้น ได้นำเอาไอซีบันทึกเสียงที่มีชื่อว่า ISD-2560 เข้ามาประยุกต์ใช้ โดยนำผ่านการควบคุมโดยไมโครคอนโทรลเลอร์

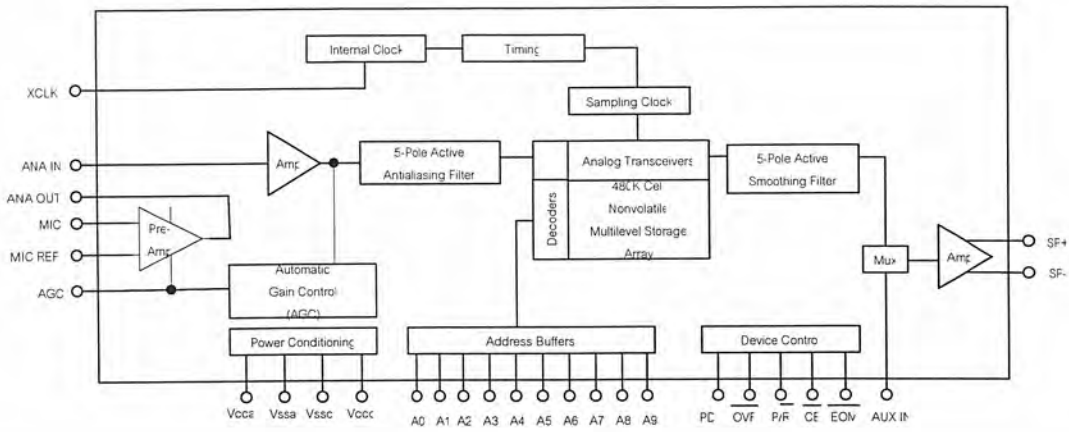
2.1.1 คุณสมบัติของ ISD-2560

- ใช้ไอซีเพียงตัวเดียวก็สามารถบันทึกและเล่นกลับได้
- ไม่ต้องมีอุปกรณ์ประเภทไอซีอื่นๆ ประกอบร่วมภายนอก
- ไม่ต้องพัฒนาระบบอื่นขึ้นมาเสริมเพื่อให้ใช้งานได้
- มีประสิทธิภาพในการบันทึกและเล่นกลับโดยให้เสียงได้เหมือนต้นกำเนิดเสียง
- สามารถควบคุมการบันทึกและการเล่นกลับได้ด้วยไมโครคอนโทรลเลอร์ได้
- สามารถต่อคาสเคด (cascade) กันเพื่อเพิ่มหน่วยความจำในการเก็บข้อมูลให้ได้มากขึ้น
- การทำงานโดยอัตโนมัติเมื่อไม่มีการบันทึกหรือเล่นกลับนานเกินไป
- สามารถเก็บความจำไว้ได้นาน 100 ปี ไม่ต้องมีแบตเตอรี่สำรอง
- สามารถบันทึกซ้ำได้ถึง 100,000 ครั้ง
- วงจรกำเนิดสัญญาณนาฬิกาภายในตัว
- สามารถโปรแกรมควบคุมการเล่นกลับเพียงอย่างเดียวเพื่อพัฒนารูปแบบใช้งาน

2.1.2 โครงสร้างภายในของ ISD-2560

จากคุณสมบัติต่างๆที่รวมอยู่ภายในไอซีเพียงตัวเดียวจึงทำให้ง่ายต่อการใช้งาน ตั้งแต่วงจรขยายสัญญาณจากไมโครโฟน จนถึงหน่วยจัดเก็บข้อมูลที่ทำการบันทึกและขับออกลำโพงก็ถูกรวมไว้ในไอซีเพียงตัวเดียวกันนี้ ในโหมดการบันทึกจะจัดเก็บข้อมูลต่างๆไว้ในหน่วยความจำที่เป็นเซลล์ (cell) แบบไม่ต้องการแรงดันสำรองเพื่อเก็บรักษาข้อมูลไม่ให้เสียหาย (Non-volatile memory cells) สัญญาณเสียงที่อยู่ในรูปแบบของสัญญาณอนาลอกจะถูกบันทึกไว้ในหน่วยจัดเก็บความจำโดยตรง และการจัดเก็บความจำก็จะจัดเก็บในลักษณะที่เป็นสัญญาณอนาลอกอยู่เช่นเดิม จึงทำให้การเล่นกลับสามารถให้สัญญาณเสียงที่เหมือนกับต้นกำเนิดเสียงมาก เพราะไม่มีกระบวนการเปลี่ยนสัญญาณอนาลอกเป็นดิจิตอลเข้ามาเกี่ยวข้อง ซึ่งสามารถแสดงโครงสร้างได้ดังรูปที่ 2.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 บล็อกไดอะแกรมภายในของ ISD-2560

ขณะทำการบันทึก ISD-2560 จะเกิดขึ้นได้นั้น สัญญาณที่เข้ามาจะต้องตรงตามเงื่อนไข ซึ่งการทำสัญญาณให้ตรงตามเงื่อนไขที่จะสามารถทำการบันทึกได้นั้นจะต้องผ่านหลายขั้นตอนด้วยกัน ขั้นแรกคือการขยายสัญญาณอินพุตให้มีขนาดมากพอสำหรับวงจรบันทึกข้อมูล ซึ่งกระบวนการนี้จะถูกกระทำโดยส่วนปรีแอมปลิไฟเออร์และวงจรควบคุมอัตราการขยาย AGC

ปรีแอมปลิไฟเออร์ (Preamplifier) เป็นส่วนที่ต่อกับไมโครโฟน โดยมีตัวเก็บประจุ (DC blocking capacitor) ทำหน้าที่แยกองค์ประกอบทางดีซี (DC component) ออกจากสัญญาณเอซี (AC signal) ที่มีระดับต่ำมากๆ ประมาณ 2-20 mV

วงจรควบคุมอัตราการขยาย (AGC circuit) จะทำหน้าที่ควบคุมระดับของสัญญาณที่ออกมาจากปรีแอมปลิไฟเออร์ และส่งค่าโวลเตจที่ถูกควบคุมอัตราการขยายแล้วไปยังแอมปลิไฟเออร์ อัตราการขยายของปรีแอมปลิไฟเออร์จะถูกปรับค่าอย่างอัตโนมัติ เพื่อที่จะคงขนาดของสัญญาณให้เหมาะสมเพียงพอที่จะส่งต่อไปในส่วนของวงจรกรองสัญญาณ (filter)

วงจรกรองความถี่ (5-Pole Active Antialiasing Filter) จะทำหน้าที่ปรับสัญญาณให้เหมาะสมและลดทอนส่วนที่ไม่สำคัญออกไป จึงได้เฉพาะเสียงที่มีคุณภาพดีในการบันทึกเป็นลำดับไป

สัญญาณที่ผ่านการปรับจนเหมาะสมแล้วจะถูกเขียนลงอย่างต่อเนื่องลงในหน่วยความจำอนาลอกในรูปการรวมสัญญาณ

การเล่นกลับ (Playback) อนาลอกโวลเตจที่ถูกบันทึกไว้จะถูกอ่านออกมาจากหน่วยความจำอย่างต่อเนื่อง จะมีส่วนทำการแยกสัญญาณให้กลับมาอยู่ในสภาพเดิม สัญญาณอนาลอกก็จะผ่านลำโพงออกมาได้ กรณีที่ใช้ ISD-2560 ต่อคาสเคดกันหลายๆตัว เราสามารถให้ลำโพงตัวเดียวกันได้โดยผ่านเข้า AUX IN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 การบันทึกข้อมูล (Recording)

กระบวนการบันทึกข้อมูลจะเกิดขึ้นได้เมื่อขา CE เป็นลอจิก "0" เพื่อให้ไอซีทำงานได้ ขา P/R จะเป็นลอจิก "0" เพื่อเข้าสู่โหมดการบันทึกและขา EOM จะเป็นลอจิก "1" เพื่อแสดงว่าตอนนี้ข้อมูลกำลังถูกบันทึก เมื่อทุกอย่างเรียบร้อยแล้วขาอนาลอกอินพุตจะรับสัญญาณอนาลอกจากภายนอกเข้ามาขยายด้วยวงจรขยายที่มีอยู่ภายในไอซี นอกจากนี้ก็ยังมีปรีแอมปลิไฟเออร์ไว้ให้ขยายสัญญาณที่มีแอมปลิจูดต่ำๆ เช่น สัญญาณจากไมโครโฟน โดยแยกออกมาเป็นอินพุตอีกขาหนึ่งต่างหาก และมีเอาต์พุตออกมายังภายนอกเพื่อที่จะต่อกลับเข้าไปที่ขาอนาลอกอินพุตเพื่อขยายด้วยวงจรขยายอีกครั้ง ทำให้สามารถที่จะเลือกใช้ขาอินพุตได้ตามความแรงของสัญญาณอนาลอก

จากบล็อกไดอะแกรมแสดงส่วนต่างๆภายในไอซี จะเห็นว่ามีวงจรควบคุมอัตราขยายอัตโนมัติ (AGC) ที่คอยควบคุมอัตราขยายของปรีแอมปลิไฟเออร์ โดยนำเอาเอาต์พุตของวงจรขยายนั้นป้อนกลับมาเปรียบเทียบเพื่อปรับความแรงของสัญญาณให้เหมาะสมโดยอัตโนมัติ สัญญาณที่ถูกขยายแล้วจะถูกกรองด้วยฟิลเตอร์ก่อนที่จะผ่านไปยังบล็อกของตัวรับ/ส่ง ข้อมูลอนาลอก (analog transceiver) ภายในบล็อกจะประกอบด้วยวงจร A/D และ D/A สัญญาณอนาลอกก็จะถูกแปลงให้อยู่ในรูปแบบของดิจิตอล โดยมีลักษณะการทำงานดังนี้

ลักษณะของสัญญาณออดิโอเป็นอนาลอกนั้น มีสิ่งสำคัญที่จะต้องคำนึงถึงอยู่สองอย่างนั่นคือขนาดของแรงดันหรือแอมปลิจูดและเวลาที่เปลี่ยนแปลงไปในการทำ A/D จะเป็นการเปรียบเทียบระดับแรงดันในขณะเวลาใดๆ ให้เป็นรหัสดิจิตอลจำนวน 10 บิต จากสูตร 2^n เมื่อ n คือจำนวนบิต จะสามารถเปลี่ยนแปลงเป็นรหัสที่ไม่ซ้ำกันได้ $2^{10} = 1024$ ค่านั้นก็คือ เราสามารถที่จะแทนระดับแรงดันที่แตกต่างกันได้ 1024 ระดับนั่นเอง รหัสดิจิตอลนี้จะถูกเก็บบันทึกลงในหน่วยความจำ EEPROM ซึ่งใน ISD-2560 นี้มีเนื้อที่ในการเก็บข้อมูลจำนวน 480 K เซล มีความสามารถในการเก็บและคงสภาวะข้อมูลได้นานถึง 10 ปี ส่วนความผิดพลาดที่อาจเกิดขึ้นได้นั้นมีสาเหตุมาจากผลของอุณหภูมิ การปรับแต่งในการใช้งานหรือเสื่อมสภาพตามอายุการใช้งาน โดยไอซีนี้สามารถจะทำการอ่านหรือเขียนข้อมูลได้กว่า 100,000 ครั้ง

ลักษณะการสุ่มบันทึกของไอซีจะใช้สัญญาณนาฬิกาเพื่อกำหนดจังหวะในการเปรียบเทียบ โดยคอมพาราเตอร์ที่อยู่ภายในบล็อกของตัวรับ/ส่งข้อมูลอนาลอก และเก็บรหัสดิจิตอลที่ได้ลงในหน่วยความจำ จากบล็อกไดอะแกรมของส่วนต่างๆภายในไอซี ส่วนที่ใช้ในการสร้างสัญญาณนาฬิกาก็คือ วงจรกำเนิดสัญญาณนาฬิกาภายใน (internal clock)

วงจรกำเนิดเวลา (Timing) และวงจรสุ่มสัญญาณนาฬิกา (sampling clock) ซึ่งความถี่ในการสุ่ม (sampling frequency) ที่สร้างขึ้นนี้จะต้องมีความเที่ยงตรงสูง เพราะผลต่างเพียง 2% ของความถี่ที่ใช้ในการบันทึกกับความถี่ที่ใช้ในขณะเล่นกลับก็จะให้เสียงที่ได้นั้นผิดเพี้ยนไปจนสามารถสังเกตได้ด้วยหูเปล่า สำหรับไอซีของ ISD นี้จะมีความผิดพลาดของวงจรสัญญาณสุ่มนี้เพียง $\pm 1\%$ เท่านั้น

ซึ่งการบันทึกข้อความในโครงงานนี้จะเป็นการบันทึกข้อความเรียงทีละข้อความ ตั้งแต่ข้อความที่ 1 จนถึงข้อความสุดท้าย วิธีการเขียนข้อความให้ทำตามลำดับขั้นตอนดังนี้

- ป้อนสัญญาณพัลส์บวกที่ขา PD เพื่อรีเซ็ตการทำงาน
- ป้อนไฟลบ (ลอจิก "0") ให้ขา P/R เพื่อเข้าสู่การบันทึกข้อความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ป้อนสัญญาณพัลส์ลบที่ขา $\overline{CE}$ เพื่อเป็นสัญญาณเริ่มต้นบันทึกข้อความที่ 1
- พูดข้อความที่ 1 ลงไปจนจบ
- ป้อนสัญญาณพัลส์ลบที่ขา $\overline{CE}$ เพื่อเป็นสัญญาณสิ้นสุดการบันทึกข้อความที่ 1
- ป้อนสัญญาณพัลส์ลบที่ขา $\overline{CE}$ เพื่อเป็นสัญญาณเริ่มต้นบันทึกข้อความที่ 2
- พูดข้อความที่ 2 ลงไปจนจบ
- ป้อนสัญญาณพัลส์ลบที่ขา $\overline{CE}$ เพื่อเป็นสัญญาณสิ้นสุดการบันทึกข้อความที่ 2

เมื่อต้องการบันทึกข้อความที่ 3, 4, 5 ... ก็ให้ทำซ้ำไปเรื่อยๆ

2.1.4 การเล่นกลับ (Play Back)

ในโครงงานนี้ การเล่นกลับจะเป็นการเล่นกลับข้อความแบบไม่เรียงลำดับ คือเป็นการเล่นกลับแบบให้เล่นข้อความไหนก่อนก็ได้ โดยไม่ต้องเรียงตามลำดับข้อความที่ทำการบันทึกไว้อยู่ก่อนแล้ว ซึ่งสามารถทำได้ตามขั้นตอนดังต่อไปนี้

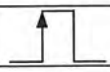
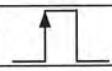
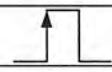
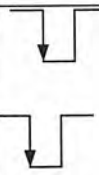
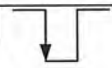
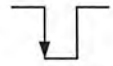
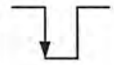
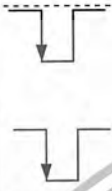
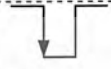
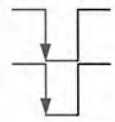
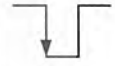
- ป้อนสัญญาณพัลส์บวกที่ขา PD เพื่อรีเซ็ตการทำงาน
- ป้อนไฟบวก (ลอจิก "1") ให้ขา P/R เพื่อเข้าสู่การเล่นกลับข้อความ
- ป้อนไฟบวก (ลอจิก "1") ที่ขา SHIFT (ในที่นี้เลือกขา A0 ซึ่งจะใช้เป็นขาอินพุตสำหรับทำหน้าที่ควบคุมการเลื่อนข้อความไปข้างหน้าทีละข้อความ) เพื่อเข้าสู่การเลื่อนข้อความ
- ป้อนสัญญาณพัลส์ลบที่ขา $\overline{CE}$ 1 ครั้ง เพื่อสั่งเลื่อนข้อความไปข้างหน้า 1 ข้อความ
- ป้อนไฟลบ (ลอจิก "0") ที่ขา SHIFT เพื่อเสร็จสิ้นการเลื่อนข้อความ
- ป้อนสัญญาณพัลส์ลบที่ขา $\overline{CE}$ ข้อความที่ 2 ก็จะทำการเล่นกลับ

จากที่กล่าวมาเป็นการเล่นกลับข้อความที่ 2 ก่อน แต่ถ้าต้องการเล่นกลับข้อความที่ 5 เป็นข้อความถัดไป จะต้องทำดังนี้

- ป้อนไฟบวก (ลอจิก "1") ที่ขา SHIFT เพื่อเข้าสู่การเลื่อนข้อความ
- ป้อนสัญญาณพัลส์ลบที่ขา $\overline{CE}$ 2 ครั้งเพื่อสั่งเลื่อนข้อความไป 2 ช่อง

ซึ่งสามารถแสดงตัวอย่างวิธีควบคุมการทำงานของ ISD-2560 ในโหมดการเล่นกลับแบบไม่เรียงลำดับของข้อความดังในตารางที่ 2.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การบันทึกข้อความ		การเล่นกลับแบบเรียงลำดับ		การเล่นกลับแบบไม่เรียงลำดับ	
ขาอินพุต	สภาวะ	ขาอินพุต	สภาวะ	ขาอินพุต	สภาวะ
ขา PD		ขา PD		ขา PD	
ขา P/R	"0"	ขา P/R	"1"	ขา P/R	"1"
ขา $\overline{CE}$ ใส่ข้อความที่ 1 ขา $\overline{CE}$		ขา $\overline{CE}$ ฟังข้อความที่ 1		ขา SHIFT ขา $\overline{CE}$ ขา SHIFT ขา $\overline{CE}$ ฟังข้อความที่ 2	"1"  "0" 
ขา $\overline{CE}$ ใส่ข้อความที่ 2 ขา $\overline{CE}$		ขา $\overline{CE}$ ฟังข้อความที่ 2		ขา SHIFT ขา $\overline{CE}$ ขา $\overline{CE}$ -ขา SHIFT ขา $\overline{CE}$ ฟังข้อความที่ 5	"1"  "0" 
ขา $\overline{CE}$ ใส่ข้อความที่ 3 ขา $\overline{CE}$		ขา $\overline{CE}$ ฟังข้อความที่ 3			

ตารางที่ 2.1 วิธีการควบคุมการทำงานของ ISD2560
ในการทำงานแบบการเล่นกลับโดยไม่เรียงตามลำดับการบันทึก

2.1.5 รายละเอียดของขาอุปกรณ์ของ ISD-2560

- Microphone Input (MIC)

ขา 17 จะรับสัญญาณอินพุตที่ผ่านเข้ามายังไมโครโฟน แล้วส่งผ่านสัญญาณเข้าสู่วงจรปรีแอมปลิไฟเออร์ที่ประกอบอยู่ภายในตัวไอซี ภายในประกอบด้วยวงจรควบคุมอัตราการขยายอัตโนมัติ (AGC) โดยวงจรนี้จะทำหน้าที่ควบคุมอัตราการขยายวงจรปรีแอมปลิไฟเออร์ให้มีการขยายในช่วง –15 ถึง 24 เดซิเบล ไมโครโฟนจากภายนอกจะถูกคับปลิ้งผ่านตัวเก็บประจุในลักษณะอนุกรมกับขา 17 ค่าความจุของตัวเก็บประจุและค่าความต้านทานภายในไอซี (10kΩ) จะกำหนดช่วงความถี่คัตออฟด้านต่ำของ passband

- Microphone Reference Input (MIC REF)

ขา18 นี้จะต่อกับกราวด์อนาลอก (V_{ssa}) โดยต่ออนุกรมกับตัวเก็บประจุเพื่อทำหน้าที่กำจัดสัญญาณรบกวนทางอินพุตขา 17 และเพื่อให้การชดเชยทางด้านสัญญาณรบกวนให้ดีกว่า 10 เดซิเบล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Analog Input (ANA IN)

ขา 20 เป็นขาอินพุตของวงจรแอมปลิไฟเออร์ที่จะนำสัญญาณเสียงเข้าไปบันทึกในหน่วยความจำ ถ้ารับอินพุตจากไมโครโฟน ระหว่างขา ANA IN และ ANA OUT ต้องมีตัวเก็บประจุภายนอกเชื่อมต่อไว้ด้วย ตัวเก็บประจุคับปลิ้งภายนอกพร้อมด้วยค่าความต้านทานภายในของ ANA IN (3 kΩ) นี้ เป็นค่าสำหรับแสดงความถี่คัตออฟด้านต่ำของ passband ถ้ารับอินพุตจากแหล่งอื่นสัญญาณสามารถถูกป้อนไปยังขา ANA IN ได้โดยตรง

- Analog Output (ANA OUT)

ขา 21 เป็นขาเอาต์พุตของวงจรปรีแอมปลิไฟเออร์ และอัตราขยายแรงดันของวงจรปรีแอมปลิไฟเออร์จะขึ้นอยู่กับระดับแรงดันที่ขา AGC

- Automatic Gain Control Input (AGC)

ขา 19 เป็นอินพุตเพื่อควบคุมอัตราการขยายของปรีแอมปลิไฟเออร์ไมโครโฟนทางด้านไดนามิก เพื่อทำให้เกิดความเหมาะสมเมื่อเทียบกับระดับสัญญาณ ที่มีย่านความถี่กว้างมากของสัญญาณจากทางด้านอินพุตจากไมโครโฟน และเพื่อให้ระดับสัญญาณที่ทำการบันทึกมีความผิดเพี้ยนน้อยที่สุด ขา AGC นี้จะต่อรวมกับอุปกรณ์ RC เพื่อกำหนดค่าเวลาคงที่ โดยมีค่าความต้านทานภายใน 5kΩ และจะต่อกับตัวเก็บประจุภายนอกอีกตัวหนึ่งเพื่อผ่านลงกราวด์อนาลอก ค่าที่เหมาะสมบางครั้งกำหนดไว้ที่ R = 470kΩ , C = 4.7μF

- Speaker Outputs (SP+, SP-)

ขา 14, 15 เป็นขาเอาต์พุตต่อออกมาลำโพง โดยในไอซีจะมีวงจรสัญญาณความแตกต่างออกสู่ลำโพง ซึ่งความสามารถในการขับลำโพงเอาต์พุตได้ 50 mW ที่โหลด 16Ω ขาเอาต์พุตนี้ไม่สามารถต่อขนานกันได้หลายตัว ในกรณีที่ต่อคาสเคดกันหลายตัว

- Power Down Input (PD)

ขา 24 ในขณะที่ไม่มีการบันทึกหรือเล่นกลับ ที่ขา PD จะมีสภาวะเป็น “1” ก็จะเป็นการรักษาระดับการสิ้นเปลืองกำลังงานในระดับต่ำมากๆ แต่เมื่อขา $\overline{OVF}$ มีสภาวะเป็น “0” ที่แสดงถึงการเล่นกลับสิ้นสุดลงปรากฏขึ้น ขา PD ปกติจะเป็น “1” อยู่ในขณะนี้ ก็จะถูกรีเซ็ตและจะเริ่มขบวนการบันทึกหรือเล่นกลับใหม่อีกครั้ง

- Chip Enable Input ($\overline{CE}$)

ขา 23 ขา $\overline{CE}$ จะต้องได้รับสัญญาณพัลส์ “0” เพื่อเกิดการเปลี่ยนแปลงระหว่างการเล่นกลับและการบันทึกที่ขาแอดเดรสอินพุต และขา P / $\overline{R}$ จะถูกแลตช์จากพัลส์ขอบขาลงของพัลส์ที่ขา $\overline{CE}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Playback/Record Input (P / $\overline{R}$)

ขา 27 เมื่อขาอินพุตควบคุมการเล่นกลับและบันทึกได้รับพัลส์ "1" จะเป็นวงรอบของการเล่นกลับและถ้าเป็นพัลส์ "0" จะเป็นการเลือกวงรอบการบันทึก ถ้าหากได้รับพัลส์ที่ขอบขาลงของขา $\overline{CE}$ จะเป็นการแลตซ์อินพุตที่ขา P / $\overline{R}$

- Address/Mode Input (A0-A9/M0-M6)

ขา 1-10 ขาแอดเดรสและโหมดอินพุตจะมีอยู่สองฟังค์ชั่นที่ขึ้นอยู่กับระดับของ MSB สองบิตของแอดเดรส ถ้ามีบิตใดบิตหนึ่งของสองบิต MSB เป็น "0" อินพุตก็จะมาปรากฏที่แอดเดรสบิตทั้งหมดและใช้เป็นแอดเดรสเริ่มต้นสำหรับวงรอบการบันทึกและการเล่นกลับ และขาแอดเดรสจะเกิดการแลตซ์โดยขอบขาลงของพัลส์ที่ขา $\overline{CE}$ และถ้า MSB ทั้งสองบิตมีสถานะเป็น "1" ขาแอดเดรสของโหมดอินพุตจะมาขึ้นอยู่ที่โหมดบิตทั้งหมดและเกิดการแลตซ์เมื่อพัลส์ขอบขาลงปรากฏที่ขา $\overline{CE}$ ซึ่งในโครงงานนี้จะต่อขา A1 ถึง A5 และ A7 ลงกราวด์ไว้ เพื่อให้วงรอบของการบันทึกแต่ละครั้งกลับไปเริ่มต้นใหม่ และต่อขา A6, A8 และ A9 กับไฟเลี้ยง ส่วนขา A0 จะใช้เป็นขารับอินพุตสำหรับทำหน้าที่ควบคุมการเลื่อนข้อความไปข้างหน้าทีละข้อความ

ขา 22 สัญญาณพัลส์ "0" จะปรากฏออกทางขาเอาต์พุตนี้เพื่อเป็นการแสดงว่าสิ้นสุดการเล่นกลับหรือหน่วยความจำภายในไอซีถูกอ่านออกมาจนหมดแล้ว และจะแสดงเป็นสภาวะเล่นกลับ พัลส์จากขานี้จะจ่ายให้กับขาอินพุตจนกว่าจะได้รับพัลส์เพื่อทำการรีเซท และเริ่มวงรอบการเล่นกลับใหม่อีกครั้ง พัลส์ที่ขานี้จะสามารถใช้เริ่มต้นการทำงานของ ISD-2560 ในตัวถัดไปได้เมื่อมีการต่อค่าคาสเคดกันอยู่หลายตัว

- Auxiliary Input (AUX IN)

ขา 11 จะเป็นขารับอินพุตจากภายนอกเพื่อทำการมัลติเพล็กซ์สัญญาณผ่านออกไปทางเอาต์พุตลำโพง โดยขั้นตอนการทำงานนี้จะเกิดขึ้นเมื่อขามีสถานะเป็น "1" วงรอบของการเล่นกลับก็จะสิ้นสุดลงหรือเมื่อสัญญาณที่บันทึกไว้ถูกเล่นกลับจนหมดแล้ว การต่อคาสเคด ISD-2560 กันหลายๆขา AUX IN จะถูกใช้เมื่อต่อเข้ากับสัญญาณเล่นกลับที่ออกมาจากขาเอาต์พุตลำโพงของตัวก่อนหน้านี้ หรือจากตัวอันดับแรก

- Voltage Input (V_{cca} , V_{ccd})

ขา 16 และ 28 เป็นขารับแรงดันที่จะต้องแยกกันระหว่างขารับแรงดันของวงจรอนาลอกและวงจรดิจิตอลที่ปรากฏอยู่ในตัวไอซีแล้ว ขารับแรงดันต้องการแรงดันไฟเลี้ยง +5 โวลต์ และต้องเป็นแรงดันไฟเลี้ยงที่มีสัญญาณรบกวนต่ำมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Ground Input (V_{ssa} , V_{ssd})

ขา 12 และ 13 โดยคุณสมบัติของไอซีในตระกูล ISD-25XX จะมีการแยกกราวด์ของสัญญาณอนาลอกและกราวด์ของสัญญาณดิจิตอลออกจากกัน ขากราวด์ทั้งสองข้างนี้จะถูกต่อและปิดไว้ในตัวถังบรรจุของไอซี การใช้งานของกราวด์ทั้งสองโดยจะเลือกต่อกับกราวด์ของพาวเวอร์ซัพพลายในส่วนที่มีค่าอิมพีแดนซ์ต่ำ เพื่อไม่ต้องการให้เกิดค่าแรงดันที่แตกต่างกันระหว่างกราวด์ทั้งสอง

- End-Of-Message/Run Output ($\overline{EOM}$)

ขา 25 เป็นขาที่แสดงการทำงานว่ากำลังบันทึกข้อความหรืออ่านข้อความอยู่โดยที่ขณะที่ทำการบันทึกข้อความหรืออ่านข้อความอยู่นั้นขานี้จะมีสถานะเป็น “1” และพอสิ้นสุดการทำงานขานี้จะมีสถานะเป็น “0”

ตำแหน่งของขาต่าง ๆ ของไอซี ISD-2560 แสดงดังรูปที่ 2.2

รูปที่ 2.2 การจัดตำแหน่งขาต่างๆ ของไอซี ISD-2560

2.2 ทฤษฎีของไมโครคอนโทรลเลอร์ตระกูล MCS-51

โครงสร้างและพื้นฐานทั่วไปของไมโครคอนโทรลเลอร์ที่จำเป็นต้องทำการศึกษาเพื่อนำไปใช้ในโครงงานระบบต้อนรับผู้มาเยือน มีดังต่อไปนี้

2.2.1 คุณสมบัติทั่วไปของไมโครคอนโทรลเลอร์ตระกูล MCS-51

คุณสมบัติทั่วไปของไมโครคอนโทรลเลอร์ตระกูล MCS-51 มีดังต่อไปนี้

- เป็นไมโครคอนโทรลเลอร์ขนาด 8 บิต
- มีวงจรออสซิลเลเตอร์และวงจรผลิตสัญญาณนาฬิกาภายในไอซี
- มีขาสัญญาณอินพุตและเอาต์พุตจำนวน 32 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สามารถเชื่อมหน่วยความจำข้อมูลภายนอก (External Data Memory) โดยการอ้างตำแหน่งแอดเดรสได้ถึง 64 K
- สามารถเชื่อมต่อหน่วยความจำโปรแกรมภายนอก (External Program Memory) โดยการอ้างตำแหน่งแอดเดรสได้ถึง 64 K
- มีหน่วยความจำข้อมูลภายในตัว (On-Chip Data Memory) ขนาด 128 ไบต์ โดยเฉพาะหมายเลข 8032 และ M8052 จะมีหน่วยความจำในส่วนนี้ถึง 256 ไบต์
- มีหน่วยความจำโปรแกรมภายในตัว (On-Chip Program Memory) ขนาด 4 K โดยเฉพาะหมายเลข 8052 จะมีหน่วยความจำในส่วนนี้ถึง 8 K สำหรับหมายเลข 8031 และ M8032 จะไม่มีหน่วยความจำในส่วนนี้
- หน่วยความจำข้อมูลภายในบางส่วนสามารถเข้าถึงข้อมูลระดับบิตได้ด้วย ทำให้สามารถควบคุมหรือตรวจสอบสถานะบิตได้ง่าย ส่งผลให้สามารถเขียนโปรแกรมได้ง่ายขึ้น
- มีไทเมอร์/เคาน์เตอร์ (Timer/Counter) ขนาด 16 บิตจำนวน 2 ตัว โดยเฉพาะหมายเลข 8032 หรือ 8052 จะมีไทเมอร์/เคาน์เตอร์จำนวน 3 ตัว
- สามารถทำการอินเตอร์รัปได้จากแหล่งกำเนิด 5 แหล่ง โดยเฉพาะหมายเลข 8032 และ 8052 จะทำการอินเตอร์รัปได้จากแหล่งกำเนิด 6 แหล่ง พร้อมทั้งยังสามารถจัดลำดับความสำคัญของการอินเตอร์รัปได้ 2 ระดับ
- มีพอร์ตสื่อสารอนุกรมภายในตัวเอง ซึ่งการทำงานจะเป็นแบบฟูลดูเพล็กซ์ (Full Duplex)
- มีคำสั่งในการคำนวณทางคณิตศาสตร์และทางตรรกศาสตร์
- คำสั่งส่วนใหญ่จะใช้เวลาในการประมวลผลเพียง 1 ไมโครวินาที เมื่อใช้คริสตอลออสซิลเลเตอร์ความถี่ 12 เมกะเฮิรตซ์
- ต้องการแหล่งจ่ายไฟกระแสตรงแรงดัน 5 โวลต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหมายเลข	หน่วยความจำภายใน		จำนวนไทเมอร์/ เคาน์เตอร์	จำนวน อินเตอรัปต์
	เก็บโปรแกรม	เก็บข้อมูล		
8052AH	8K x 8 ROM	256 x 8 RAM	3 x 16-Bit	6
8051AH	4K x 8 ROM	128 x 8 RAM	2 x 16-Bit	5
8051	4K x 8 ROM	128 x 8 RAM	2 x 16-Bit	5
8032AH	ไม่มี	256 x 8 RAM	2 x 16-Bit	6
8031AH	ไม่มี	128 x 8 RAM	2 x 16-Bit	5
8031	ไม่มี	128 x 8 RAM	2 x 16-Bit	5
8751H	4K x 8 EPROM	128 x 8 RAM	2 x 16-Bit	5
8751H-12	4K x 8 EPROM	128 x 8 RAM	2 x 16-Bit	5

ตารางที่ 2.2 คุณสมบัติของไมโครคอนโทรลเลอร์แต่ละหมายเลขในตระกูล MCS-51

2.2.2 โครงสร้างภายนอกไมโครคอนโทรลเลอร์ตระกูลของ MCS-51

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ทุกหมายเลขจะมีตำแหน่งขาพื้นฐานที่เหมือนกันดังแสดงในรูปที่ 2.3 สำหรับหน้าที่การใช้งานของแต่ละขามีดังนี้

รูปที่ 2.3 การจัดตำแหน่งขาต่างๆ ของไมโครคอนโทรลเลอร์ตระกูล MCS-51

- ขา V_{CC} เป็นขาสำหรับป้อนแรงดันไฟเลี้ยง +5 โวลต์
- ขา V_{SS} เป็นขากราวด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ขาพอร์ต 0 (Port 0) มี 8 ขาได้แก่ขา $P_{0.0}$-$P_{0.7}$ เป็นขาพอร์ตอินพุตและเอาต์พุตแบบ 2 ทิศทางสำหรับใช้งานทั่วไป โดยถ้าใช้งานเป็นอินพุตพอร์ตต้องทำการเขียนค่า 1 ไปยังแต่ละบิตของพอร์ต เพื่อกำหนดให้ขาพอร์ตเหล่านั้นอยู่ในสภาวะปล่อยลอย ซึ่งในสถานะนี้เองที่สามารถนำใช้เป็นพอร์ตอินพุตอิมพีแดนซ์สูงได้ นอกจากพอร์ตนี้จะใช้งานเป็นพอร์ตอินพุตและเอาต์พุตแล้ว ยังถูกใช้งานในการติดต่อกับหน่วยความจำภายนอกด้วย โดยทำหน้าที่ในการกำหนดแอดเดรสไบต์ต่ำ (A_0–A_7) ซึ่งจะใช้งานเป็นแบบมัลติเพล็กซ์กับการรับส่งข้อมูลขนาด 8 บิต (D_0–D_7)
- ขาพอร์ต 1 (Port 1) มี 8 ขาได้แก่ขา $P_{1.0}$-$P_{1.7}$ เป็นขาพอร์ตอินพุตและเอาต์พุตแบบ 2 ทิศทางสำหรับใช้งานทั่วไป โดยถ้าใช้งานเป็นอินพุตพอร์ตต้องทำการเขียนค่า 1 ไปยังแต่ละบิตของพอร์ต เพื่อกำหนดให้เป็นพอร์ตอินพุต นอกจากนี้สำหรับหมายเลข 8032 และ 8052 ขาพอร์ต $P_{1.0}$ และ $P_{1.1}$ จะถูกนำมาใช้งานเป็นขา T2 และ T2EX ตามลำดับด้วย
- ขาพอร์ต 2 (Port 2) มี 8 ขาได้แก่ขา $P_{2.0}$-$P_{2.7}$ เป็นขาพอร์ตอินพุตและเอาต์พุตแบบ 2 ทิศทางสำหรับใช้งานทั่วไป โดยถ้าใช้งานเป็นอินพุตพอร์ตต้องทำการเขียนค่า 1 ไปยังแต่ละบิตของพอร์ต เพื่อกำหนดให้เป็นพอร์ตอินพุต นอกจากพอร์ตนี้จะใช้งานเป็นพอร์ตอินพุตและเอาต์พุตแล้ว ยังถูกใช้งานในการติดต่อกับหน่วยความจำภายนอกด้วย โดยทำหน้าที่ในการกำหนดแอดเดรสไบต์สูง (A_8–A_{15})
- ขาพอร์ต 3 (Port 3) มี 8 ขาได้แก่ขา $P_{3.0}$-$P_{3.7}$ เป็นขาพอร์ตอินพุตและเอาต์พุตแบบ 2 ทิศทางสำหรับใช้งานทั่วไป โดยถ้าใช้งานเป็นอินพุตพอร์ตต้องทำการเขียนค่า 1 ไปยังแต่ละบิตของพอร์ต เพื่อกำหนดให้เป็นพอร์ตอินพุต นอกจากพอร์ตนี้จะใช้งานเป็นพอร์ตอินพุตและเอาต์พุตแล้ว ยังถูกใช้งานในหน้าที่พิเศษต่างๆ ดังแสดงในตารางที่ 2.3 ด้วย

ขาพอร์ต	หน้าที่พิเศษ
$P_{3.0}$	RXD (serial input port)
$P_{3.1}$	TXD (serial output port)
$P_{3.2}$	INT0 (external interrupt 0)
$P_{3.3}$	INT1 (external interrupt 1)
$P_{3.4}$	T0 (timer 0 external input)
$P_{3.5}$	T1 (timer 1 external input)
$P_{3.6}$	WR (external data memory write strobe)
$P_{3.7}$	RD (external data memory read strobe)

ตารางที่ 2.3 หน้าที่พิเศษของแต่ละขาของพอร์ต P_3

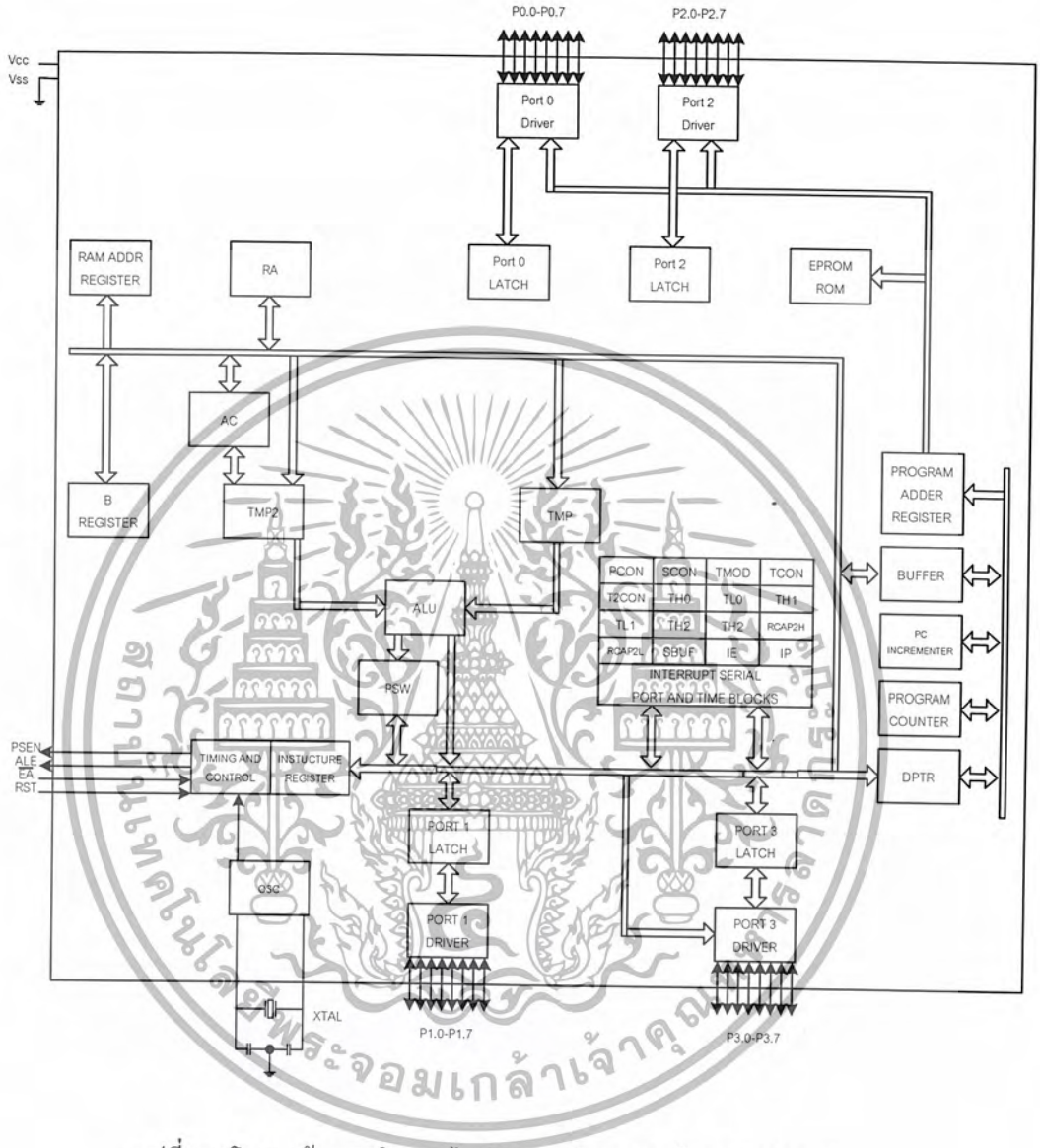
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ขารีเซ็ต (RST) ใช้สำหรับทำการรีเซ็ตการทำงานของไมโครคอนโทรลเลอร์ โดยการรีเซ็ตต้องคงสถานะเป็น 1 นานอย่างน้อย 2 แมชีนไซเคิล ในขณะที่ออสซิลเลเตอร์ยังคงทำงานอยู่
- ขา $\overline{\text{ALE/PROG}}$ เป็นขาสัญญาณเพื่อทำหน้าที่ควบคุมการแลตช์ (Latch) ค่าตำแหน่งแอดเดรสไบต์ต่ำ (Address Latch Enable) เมื่อต้องการติดต่อกับหน่วยความจำภายนอก นอกจากนี้ขานี้ยังทำหน้าที่เป็นอินพุตรับพัลส์ในการโปรแกรม (Program Pulse Input) ในส่วนของหน่วยความจำ EPROM สำหรับไมโครคอนโทรลเลอร์ตระกูล MCS-51 ที่มีหน่วยความจำโปรแกรมภายในเป็น EPROM
- ขา $\overline{\text{PSEN}}$ (Program Store Enable) ทำหน้าที่เป็นสัญญาณสโตรบเพื่ออ่านคำสั่งจากหน่วยความจำโปรแกรมภายนอก เมื่อไมโครคอนโทรลเลอร์ทำการประมวลผลคำสั่งจากหน่วยความจำภายนอก ขานี้จะส่งสัญญาณสโตรบจำนวน 2 ครั้งในแต่ละแมชีนไซเคิล แต่ในขณะที่ติดต่อกับหน่วยความจำภายนอกจะไม่มีการส่งสัญญาณสโตรบแต่อย่างใด
- ขา $\overline{\text{EA}}$/VPP (External Access Enable/VPP) เป็นขาสำหรับการเลือกใช้หน่วยความจำโปรแกรมจากภายในหรือจากภายนอก โดยถ้าขานี้มีสถานะเป็น “0” จะหมายถึง ให้ไมโครคอนโทรลเลอร์รับคำสั่งจากหน่วยความจำภายนอกที่ตำแหน่งแอดเดรส 0-0FFFH (0-1FFFH ถ้าเป็นหมายเลข 8052) อย่างไรก็ตามถ้าบิตป้องกัน (Security Bit) ในหน่วยความจำ EPROM ถูกโปรแกรมไว้ ไมโครคอนโทรลเลอร์จะไม่รับคำสั่งจากหน่วยความจำภายนอกเลย นอกจากนี้ ขานี้ยังทำหน้าที่รับแรงดันไฟสำหรับการโปรแกรม (V_{pp}) ขนาด 21 โวลต์ เพื่อใช้ระหว่างการโปรแกรม EPROM
- ขา XTAL1 และขา XTAL2 เป็นขาอินพุตและเอาต์พุตของวงจรอินเวอร์ติ้งออสซิลเลเตอร์แอมปลิไฟเออร์ (Inverting Oscillator Amplifier) สำหรับใช้ต่อร่วมกับคริสตอลออสซิลเลเตอร์ภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 โครงสร้างภายในของ MCS-51

โครงสร้างภายในของไมโครคอนโทรลเลอร์ตระกูล MCS-51 แสดงดังรูปที่ 2.4



รูปที่ 2.4 โครงสร้างภายในของไมโครคอนโทรลเลอร์ตระกูล MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.4 การจัดหน่วยความจำ

ในไมโครคอนโทรลเลอร์ตระกูล MCS-51 แบ่งชนิดหรือหน้าที่ของหน่วยความจำออกเป็น 2 ส่วนคือ หน่วยความจำโปรแกรม (Program Memory) และ หน่วยความจำข้อมูล (Data Memory)

หน่วยความจำโปรแกรมใช้สำหรับเก็บโปรแกรมควบคุมการทำงานของไมโครคอนโทรลเลอร์ ซึ่งบางหมายเลขจะมีหน่วยความจำนี้อยู่ในตัว โดยอาจจะมีขนาดไม่เท่ากันหรือเป็นหน่วยความจำต่างชนิดกัน เช่น บางหมายเลขเป็น ROM แต่บางหมายเลขอาจเป็น EPROM และบางหมายเลขอาจไม่มีหน่วยความจำในส่วนนี้เลย โปรแกรมการทำงานจะถูกเก็บไว้ยังหน่วยความจำโปรแกรมภายนอกทั้งหมด

สำหรับหน่วยความจำข้อมูลจะใช้สำหรับเก็บข้อมูลหรือค่าตัวแปรต่างๆ จากการทำงานของโปรแกรมซึ่งใน MCS-51 ทุกหมายเลขจะมีหน่วยความจำส่วนนี้อยู่จำนวนหนึ่ง แต่อาจมีขนาดมากน้อยต่างกันไปในแต่ละหมายเลข สำหรับการจัดโครงสร้างของหน่วยความจำทั้งในส่วนของหน่วยความจำโปรแกรมและหน่วยความจำข้อมูลแสดงไว้ในรูปที่ 2.5

รูปที่ 2.5 การจัดโครงสร้างของหน่วยความจำทั้งในส่วนของ
หน่วยความจำโปรแกรมและหน่วยความจำข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.4.1 หน่วยความจำโปรแกรม

หน่วยความจำโปรแกรมสามารถแบ่งออกได้เป็น 2 ส่วนคือ หน่วยความจำโปรแกรมภายในและหน่วยความจำโปรแกรมภายนอก หน่วยความจำภายในจะถูกเลือกใช้งานถ้าขาสัญญาณ EA มีค่าเป็น 1 โดยจะถูกใช้งานในช่วงแอดเดรส 0-0FFFH (หรือช่วงแอดเดรส 0-1FFFH ในหมายเลข 8052) ในกรณีตรงข้ามถ้าขาสัญญาณ EA มีค่าเป็น 0 ในช่วงแอดเดรส 0-0FFFH (หรือช่วงแอดเดรส 0-1FFFH ในหมายเลข 8052) จะถูกใช้จากหน่วยความจำภายนอก หรือกล่าวได้ว่าถ้าขาสัญญาณ EA มีค่าเป็น 0 จะเป็นการเลือกใช้หน่วยความจำโปรแกรมภายนอกทั้งหมดตลอดช่วงแอดเดรส

2.2.4.2 หน่วยความจำข้อมูล

หน่วยความจำข้อมูลสามารถแบ่งออกได้เป็น 2 ส่วนคือ หน่วยความจำข้อมูลภายในและหน่วยความจำข้อมูลภายนอก สำหรับหน่วยความจำข้อมูลภายในยังแบ่งออกได้เป็น 2 ส่วนย่อยคือ ส่วนที่ใช้เก็บข้อมูลทั่วไปและส่วนที่ใช้เป็นรีจิสเตอร์หน้าที่พิเศษหรือ SFR (Special Function Register) โดยส่วนที่ทำหน้าที่เก็บข้อมูลทั่วไปจะถูกใช้สำหรับเก็บข้อมูลหรือค่าตัวแปรต่างๆ จากการทำงานของโปรแกรม ส่วนรีจิสเตอร์หน้าที่พิเศษจะถูกใช้งานเป็นรีจิสเตอร์ควบคุมการทำงาน และแสดงสถานะการทำงานของไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ทุกหมายเลขจะมีหน่วยความจำข้อมูลภายในขนาด 128 ไบต์เป็นอย่างน้อย และบางหมายเลขอาจมีถึง 256 ไบต์

2.2.4.3 รีจิสเตอร์ใช้งานทั่วไป

รีจิสเตอร์ใช้งานทั่วไปมีไว้สำหรับให้ผู้เขียนโปรแกรมสามารถนำข้อมูลไปพักไว้ชั่วคราว หรือใช้งานทั่วไปได้ตามต้องการ ซึ่งรีจิสเตอร์ใช้งานทั่วไปนี้มีอยู่ด้วยกัน 8 ตัวคือ รีจิสเตอร์ R0-R7 โดยรีจิสเตอร์ทั้ง 8 ตัวถูกจัดให้อยู่รวมกันและมีให้เลือกใช้ถึง 4 แบงก์ (bank) นั่นคือ มีรีจิสเตอร์ใช้งานทั่วไปถึง 32 ตัวให้ใช้งาน เพียงแต่การเลือกใช้รีจิสเตอร์ R0-R7 ในแบงก์ใดแบงก์หนึ่งจะถูกกำหนดจากบิต RS0, RS1 ในรีจิสเตอร์หน้าที่พิเศษ PSW ดังนั้นการเลือกใช้จึงเลือกได้เพียงแบงก์เดียวในขณะใดขณะหนึ่ง อย่างไรก็ตาม ค่าของข้อมูลที่เก็บไว้ในรีจิสเตอร์มีชื่อเดียวกันแต่อยู่คนละแบงก์จะไม่มีผลซึ่งกันและกันเลย ทำให้ผู้เขียนโปรแกรมสามารถใช้งานรีจิสเตอร์ทั่วไปได้ทั้ง 32 ตัวอย่างเต็มที่และไม่ยุ่งยากในการเขียนโปรแกรม

2.2.4.4 รีจิสเตอร์หน้าที่พิเศษ (SFR)

รีจิสเตอร์หน้าที่พิเศษมีบทบาทอย่างมาก ในการควบคุมการทำงานของไมโครคอนโทรลเลอร์และทำให้การเขียนโปรแกรมสามารถทำได้สะดวกขึ้น รีจิสเตอร์หน้าที่พิเศษนี้ทำหน้าที่สำคัญคือควบคุมการทำงานในส่วนต่างๆ ภายในไมโครคอนโทรลเลอร์และยังทำหน้าที่แสดงสถานะการทำงาน ซึ่งในรีจิสเตอร์หน้าที่พิเศษบางตัวยังสามารถเข้าถึงได้ในระดับบิต (bit addressable) ด้วย ดังแสดงการจัดหน่วยความจำและตำแหน่งรีจิสเตอร์หน้าที่พิเศษต่างๆ ในรูปที่ 2.6

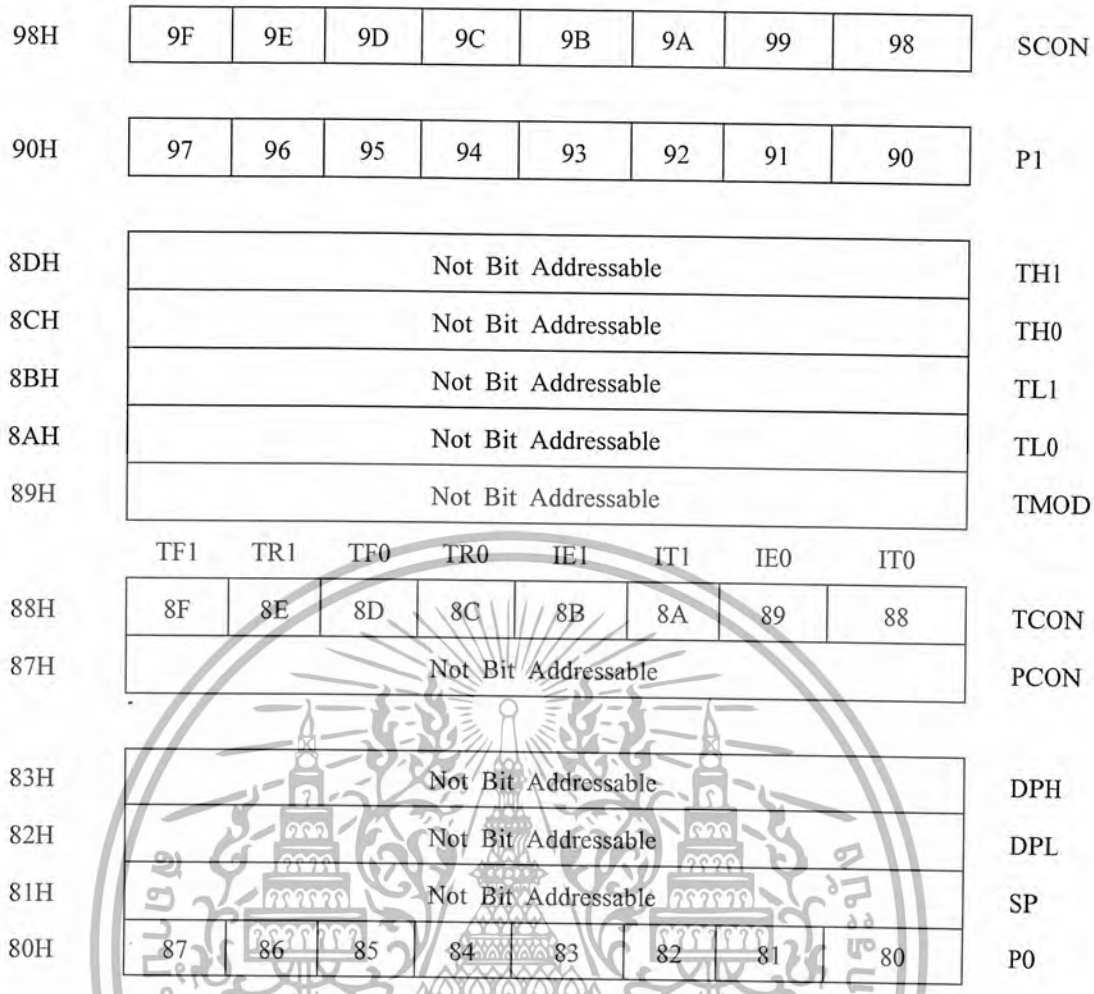
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

DIRECT Byte Address	Bit Address (MSB)							(LSB)	Special Function Register Symbol
	WDT	T32	SERR	IZC	P3HZ	P2HZ	P1HZ	ALF	
0F8H	FF	FE	FD	FC	FB	FA	F9	F8	IOCON
0F0H	F7	F6	F5	F4	F3	F2	F1	F0	B
0E0H	E7	E6	E5	E4	E3	E2	E1	E0	ACC
	CY	AC	F0	RS1	RS0	0V	F1	P	
0D0H	D7	D6	D5	D4	D3	D2	D1	D0	PSW
0CDH	Not Bit Addressable								TH2
0CCH	Not Bit Addressable								TL2
0CBH	Not Bit Addressable								RCAP2H
0CAH	Not Bit Addressable								RCAP2L
	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	$C/\overline{T}$	$CP/\overline{RL}$	
0C8H	CF	CE	CD	CC	CB	CA	C9	C8	T2CON
	PCT		PT2	PS	PT1	PX1	PT0	PX0	
0B8H	BF	–	BD	BC	BB	BA	B9	B8	IP
0B0H	B7	B6	B5	B4	B3	B2	B1	B0	P3
	EA		ET2	ES	ET1	EX1	ET0	EX0	
0A8H	AF	–	AD	AC	AB	AA	A9	A8	IE
0A0H	A7	A6	A5	A4	A3	A2	A1	A0	P2
99H	Not Bit Addressable								SBUF
	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

61967



รูปที่ 2.6 การจัดหน่วยความจำและตำแหน่งรีจิสเตอร์หน้าที่พิเศษต่างๆ

2.2.5 การใช้งานรีจิสเตอร์

โดยปกติไมโครคอนโทรลเลอร์ตระกูล MCS-51 จะทำการประมวลผลข้อมูลครั้งละ 1 ไบต์ ซึ่งกระทำกับรีจิสเตอร์ภายในโดยที่รีจิสเตอร์แต่ละตัวเก็บข้อมูลได้ขนาด 1 ไบต์เช่นกัน เช่น รีจิสเตอร์ A ซึ่งเป็นแอคคิวมูเลเตอร์ (accumulator) ทำหน้าที่เป็นรีจิสเตอร์กลางสำหรับการคำนวณทางคณิตศาสตร์หรือทางลอจิกของตัวกระทำ 2 ตัว ตัวอย่างเช่น ถ้าต้องการบวกค่า 10 กับข้อมูลตัวหนึ่ง ให้ทำการโหลดข้อมูลไปเก็บไว้ในรีจิสเตอร์ A ก่อน จากนั้นให้ใช้คำสั่งนำค่า 10 ไปบวกกับ A ผลที่ได้จากการบวกข้อมูลและค่า 10 จะถูกเก็บไว้ในรีจิสเตอร์ A นอกจากรีจิสเตอร์ A ทำการบวกด้วยการกำหนดค่าโดยตรงแล้ว มันยังทำการคำนวณร่วมกับรีจิสเตอร์ขนาด 8 บิตตัวอื่นๆ ได้อีกด้วย

ทั้งไมโครโปรเซสเซอร์ และไมโครคอนโทรลเลอร์จะมีรีจิสเตอร์สำหรับใช้งานในคำสั่งพิเศษโดยผู้เขียนโปรแกรมอาจกำหนดขึ้นเองได้ โดยที่กำหนดให้อยู่ในตำแหน่งแอดเดรสพิเศษซึ่งในที่นี้มีค่ามากกว่า 07FH ขึ้นไป ตัวอย่างเช่น แอคคิวมูเลเตอร์ถูกกำหนดให้ใช้หน่วยความจำภายในที่ 0E0H รีจิสเตอร์เหล่านี้เรียกว่า รีจิสเตอร์หน้าที่พิเศษ (Special Function Registers : SFRs) จำนวนของรีจิสเตอร์หน้าที่พิเศษอาจจะมีไม่เท่ากันในไมโครคอนโทรลเลอร์แต่ละหมายเลขในตระกูล MCS-51 ขึ้นอยู่กับคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ได้ทำการตั้งค่าไว้ เพราะรีจิสเตอร์หน้าที่พิเศษเหล่านี้ถูกรวม หรือใช้พื้นที่ในหน่วยความจำภายในของไมโครคอนโทรลเลอร์ ซึ่งหน่วยความจำภายในหรือแรมภายในจะมีขนาดไม่เท่ากันในแต่ละหมายเลข

นอกจากรีจิสเตอร์หน้าที่พิเศษหรือ SFR แล้วยังมีรีจิสเตอร์สำหรับใช้งานทั่วไปอีก 8 ตัวคือ รีจิสเตอร์ R0-R7 รีจิสเตอร์ทั้ง 8 ตัวนี้ถูกบรรจุอยู่ภายในแรมภายในไมโครคอนโทรลเลอร์ในรูปของแบงก์ และใช้สำหรับเก็บข้อมูลชั่วคราวระหว่างการประมวลผล

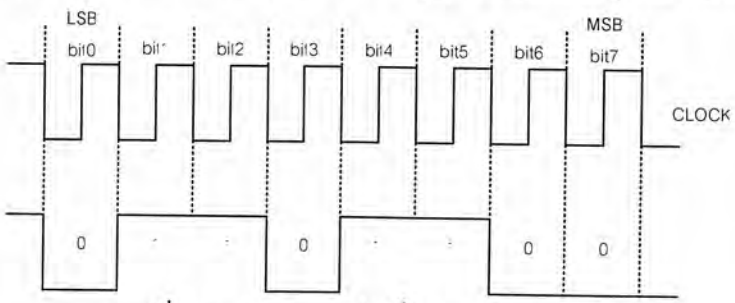
2.2.6 การเชื่อมต่อไมโครคอนโทรลเลอร์ตระกูล 8051 กับคอมพิวเตอร์ผ่านพอร์ตอนุกรม

การเคลื่อนย้ายข้อมูลจากคอมพิวเตอร์ไปยังอุปกรณ์ต่อพ่วงอื่นๆ หรือคอมพิวเตอร์ด้วยกันนั้น มี 2 วิธี คือ การรับส่งข้อมูลแบบขนานและการรับส่งข้อมูลแบบอนุกรม การรับส่งข้อมูลแบบขนานจะเป็นการรับส่งข้อมูลคราวละ 4 หรือ 8 บิต ในเวลาเดียวกันซึ่งจะทำให้การรับและส่งข้อมูลทำได้ที่ความเร็วสูง ซึ่งหมายความว่า จำนวนสายที่ใช้ในการส่งจะต้องมีมากเท่ากับจำนวนบิตของข้อมูลที่จะส่งด้วย นอกจากนี้ยังจะต้องรวมถึงสายที่ใช้สำหรับควบคุมและการตรวจสอบการรับส่งข้อมูลด้วย ซึ่งอาจจะต้องใช้สายมากเป็น 2 เท่าของจำนวนบิตข้อมูลที่จะส่งก็ได้ ซึ่งก็เป็นปัญหาในเรื่องของราคาของสายที่ใช้ในการเชื่อมต่อแบบขนานมักจะมีราคาแพง

ในขณะที่การรับส่งข้อมูลแบบอนุกรมเป็นการรับส่งข้อมูลครั้งละ 1 บิต แต่ก็สามารถรับส่งข้อมูลได้คราวละหลายๆ บิตได้ หากแต่จะต้องมีการตกลงกันระหว่างตัวส่งและตัวรับว่า จะรับส่งข้อมูลคราวละกี่บิต ตัวรับจะต้องรอข้อมูลมาให้ครบทุกบิตเสียก่อนจึงจะทำการประมวลผล ส่งผลให้การสื่อสารข้อมูลแบบอนุกรมอาจมีความเร็วต่ำกว่าแบบขนาน อย่างไรก็ตาม การรับส่งข้อมูลแบบอนุกรมนั้นสามารถใช้สายสัญญาณที่มีความยาวมากกว่าแบบขนาน ทำให้ระยะทางในการสื่อสารข้อมูลแบบอนุกรมสามารถทำได้มากกว่า

2.2.6.1 การสื่อสารแบบอนุกรม

การสื่อสารแบบอนุกรมนั้นจะแบ่งออกได้เป็น 2 แบบ คือ การสื่อสารอนุกรมแบบซิงโครนัสและการสื่อสารอนุกรมแบบอะซิงโครนัส การสื่อสารแบบซิงโครนัสจะมีสัญญาณนาฬิการ่วมอยู่กับการรับและส่งสัญญาณด้วย ตัวอย่างการส่งข้อมูลแบบซิงโครนัสคือคีย์บอร์ดของคอมพิวเตอร์ ซึ่งสายเส้นหนึ่งจะเป็นสายของสัญญาณนาฬิกา ส่วนสายอีกเส้นหนึ่งจะเป็นสายของข้อมูล ดังนั้นการติดต่อแบบซิงโครนัสนี้จะต้องใช้สายในการเชื่อมต่ออย่างน้อยที่สุด 3 เส้น คือ สัญญาณนาฬิกา, ข้อมูล และกราวด์



รูปที่ 2.7 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรม

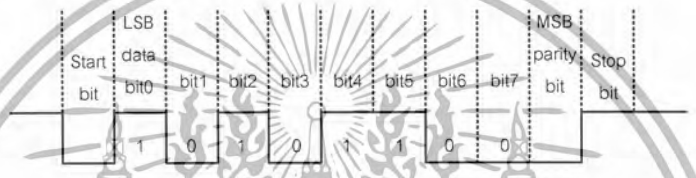
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.6.2 การสื่อสารข้อมูลแบบอะซิงโครนัส

การสื่อสารข้อมูลแบบอะซิงโครนัส คือการรับส่งข้อมูลไปในสายสัญญาณ โดยไม่จำเป็นต้องส่งสัญญาณนาฬิการ่วมด้วยเหมือนกับการรับส่งข้อมูลแบบซิงโครนัส แต่จะใช้การกำหนดค่าสัญญาณนาฬิกาทั้งภาครับและภาคส่งให้มีค่าเท่ากัน ซึ่งเรียกสัญญาณนาฬิกาที่ใช้ในการกำหนดค่าให้ภาครับและภาคส่งนี้ว่า อัตราการถ่ายเทข้อมูล หรือ อัตราบอด (baudrate) ที่มีหน่วยเป็น บิตต่อวินาที (bit per second : bps)

รูปแบบของข้อมูลที่ใช้ในการรับส่งแบบอะซิงโครนัสประกอบด้วย 4 ส่วนด้วยกัน คือ

1. บิตเริ่มต้น (Start Bit) ซึ่งจะมีขนาด 1 บิต
2. บิตข้อมูลอนุกรมจะมีขนาด 5, 6, 7 หรือ 8 บิต
3. บิตตรวจสอบพาริตี้ (Parity Bit) จะมีขนาด 1 บิตหรือไม่มี
4. บิตปิดท้าย (Stop Bit) จะมีขนาด 1, 1.5 หรือ 2 บิต



รูปที่ 2.8 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรมแบบอะซิงโครนัส

รูปที่ 2.8 แสดงรูปแบบของข้อมูลอนุกรมแบบอะซิงโครนัส ซึ่งเมื่อไม่มีข้อมูลที่จะส่ง ขาดาต้าจะมีสถานะลอจิก "1" ซึ่งจะเรียกสถานะนี้ว่าสถานะว่าง (idle stage) การเริ่มต้นส่งข้อมูลจะเริ่มจากการให้ขาดาต้ามีลอจิก "0" ด้วยช่วงระยะเวลา 1 บิต ซึ่งจะเรียกบิตนี้ว่าบิตเริ่มต้น จากนั้นบิตข้อมูลจะถูกส่งออกไปโดยเริ่มจากบิตที่มีนัยสำคัญต่ำสุด (LSB) ก่อน ซึ่งข้อมูลในไบต์ที่จะส่งอาจจะมีจำนวนบิต 5, 6, 7 หรือ 8 บิตก็ได้ จากนั้นจะตามด้วยบิตพาริตี้ ซึ่งใช้เพื่อตรวจสอบความผิดพลาดที่เกิดขึ้นจากการส่งข้อมูล บิตสุดท้ายที่จะส่งคือบิตปิดท้าย ซึ่งจะให้ขาดาต้ามีสถานะลอจิก 1 อีกครั้งด้วยระยะเวลาอย่างน้อย 1 บิต, 1.5 บิต หรือ 2 บิต เพื่อเป็นการแสดงว่าสิ้นสุดข้อมูลแล้ว

อุปกรณ์พิเศษที่ได้รับการออกแบบมาสำหรับการรับและการส่งข้อมูลแบบอะซิงโครนัส เรียกว่า Universal Asynchronous Receiver / Transmitter หรือ UART อัตราความเร็วในการรับและส่งข้อมูลของการรับส่งข้อมูลแบบอะซิงโครนัส คือ ค่าอัตราบอด ซึ่งก็คือค่าอัตราการเข้ารหัสที่ใช้ในการรับและส่งข้อมูล อัตราบอดมาตรฐานที่ใช้สำหรับพอร์ตอนุกรม RS-232 ได้แก่ 110, 150, 300, 600, 1200, 2400, 4800, 9600 และ 19200 บิตต่อวินาที และมีค่าเพิ่มมากขึ้นตามเทคโนโลยีของคอมพิวเตอร์ ซึ่งการรับส่งแบบอนุกรมโดยไม่ผ่านโมเด็มอาจจะสามารถกำหนดค่าอัตราบอดได้สูงถึง 115200 บิตต่อวินาที เนื่องจากอัตราบอดคือจำนวนบิตของข้อมูลที่สามารถถ่ายทอดได้ภายใน 1 วินาที ยกตัวอย่าง ข้อมูลอนุกรมถูกส่งในลักษณะ 8 บิต ไม่มีการตรวจสอบพาริตี้ มีบิตเริ่มต้น 1 บิตและบิตปิดท้าย 1 บิตความยาวของข้อมูลที่รับส่งนี้เท่ากับ 10 บิต ถ้าใช้อัตราบอดในการส่งข้อมูลเท่ากับ 9600 บิตต่อวินาที ก็จะสามารถรับส่งข้อมูลได้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความเร็ว 960 ไบต์ต่อวินาที และถ้ามีการใช้พาริตี้ความเร็วในการรับส่งข้อมูลจะเหลือเป็น 872 ไบต์ต่อวินาที

การตรวจสอบพาริตี้สามารถกำหนดให้เป็นแบบคี่ (odd), แบบคู่ (even) หรืออาจไม่มีการตรวจสอบพาริตี้ก็ได้ การตรวจสอบพาริตี้เป็นการตรวจสอบจำนวนรวมของบิตที่เป็นลอจิก "1" ภายในข้อมูลที่ส่งไป 1 ไบต์ว่ามีจำนวนรวมเป็นเลขคู่หรือเลขคี่โดยต้องรวมบิตพาริตี้เข้าไปด้วย ยกตัวอย่างข้อมูลที่จะทำการส่งมีขนาด 8 บิตและมีค่าเท่ากับ 99 ฐานสิบหก หรือ 10011001 ฐานสอง จะเห็นว่าข้อมูลในไบต์นี้มีลอจิก "1" จำนวน 4 ตัวซึ่งเป็นเลขคู่ ดังนั้น ถ้ากำหนดค่าพาริตี้เป็นคู่ ค่าในบิตพาริตี้จะต้องมีลอจิก "0" แต่ถ้าพาริตี้เป็นคี่ ค่าที่บิตพาริตี้จะต้องเป็น "1" เพื่อให้ข้อมูล 1 ไบต์ รวมทั้งบิตพาริตี้มีจำนวนบิตที่เป็นลอจิก "1" มีจำนวนรวมกันเป็นเลขคี่ ตารางที่ 2.4 แสดงตัวอย่างของบิตพาริตี้ในการรับส่งข้อมูลอนุกรม

ข้อมูล	บิตพาริตี้คู่	บิตพาริตี้คี่
00000000	0	1
00000001	1	0
00000010	1	0
00000011	0	1
00000100	1	0
11111110	1	0
11111111	0	1

ตารางที่ 2.4 แสดงบิตพาริตี้ของข้อมูล

บิตพาริตี้ถูกสร้างขึ้นจากภาคส่งข้อมูลของ UART ซึ่งทางภาครับจะต้องทำการกำหนดคุณสมบัติการตรวจสอบพาริตี้ให้ตรงกันว่าจะตรวจสอบพาริตี้คี่หรือพาริตี้คู่ จากนั้นภาครับของ UART จะทำการตรวจสอบค่าพาริตี้ที่เกิดขึ้นว่าเป็นคู่หรือเป็นคี่ โดยการนับจำนวนลอจิก "1" ทั้งหมดรวมทั้งบิตพาริตี้ด้วย ถ้ากำหนดพาริตี้ไว้เป็นคู่แต่อ่านค่าตัวเลขในการนับออกมาได้ตัวเลขเป็นคี่ ทางภาครับจะแสดงข้อผิดพลาดออกมาให้ผู้ใช้ทราบ นับเป็นการตรวจสอบความผิดพลาดที่เกิดขึ้นในการถ่ายทอดข้อมูลที่ง่ายที่สุด แต่จะเชื่อถือได้เมื่อมีบิตข้อมูลที่ทำการส่งผิดพลาดแค่เพียงบิตเดียวเท่านั้น ถ้าข้อมูลที่ทำการส่งมีบิตผิดพลาดมากกว่า 1 บิต การตรวจสอบด้วยวิธีนี้จะไม่ได้ผล สำหรับการตั้งพาริตี้บิตเป็น NONE นั้นทั้งภาครับและภาคส่งจะไม่มีการตรวจสอบพาริตี้

คอมพิวเตอร์ในรุ่น AT เกือบทั้งหมดจะใช้ UART หมายเลข 16450 และ 16550 ส่วนคอมพิวเตอร์ในรุ่น XT ใช้ UART หมายเลข 8250 UART ชิปเหล่านี้มีระดับแรงดันเป็นแบบทีทีแอล (0 และ +5V) แต่เพื่อให้มีแรงดันเป็นไปตามมาตรฐาน RS-232 และเพื่อให้การรับส่งข้อมูลสามารถทำได้ที่ระยะทางไกลขึ้นระดับแรงดันทีทีแอลจะถูกแปลงเป็นระดับแรงดันที่สูงขึ้น โดยลอจิก "0" มีระดับแรงดัน +3V ถึง +12V ในขณะที่ลอจิก "1" มีระดับแรงดัน -3V จนถึง -12V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.6.3 มาตรฐานพอร์ตอนุกรมแบบ RS-232

มาตรฐานการเชื่อมต่อแบบอนุกรม RS-232 เป็นมาตรฐานอุตสาหกรรมที่ออกแบบมาเพื่อใช้ในการส่งข้อมูลอนุกรมแบบอะซิงโครนัส 2 ทิศทาง โดยมาตรฐาน RS-232 ในอดีตนั้นถูกออกแบบมาเพื่อการส่งผ่านข้อมูลจากคอมพิวเตอร์ไปยังโมเด็มเพียงอย่างเดียว เพื่อที่จะนำข้อมูลจากโมเด็มนี้สื่อสารผ่านสายโทรศัพท์ไปยังคอมพิวเตอร์อีกชุดที่อยู่ห่างไกลกัน โดยคณะกรรมการที่เรียกว่า สมาคมอุตสาหกรรมอิเล็กทรอนิกส์ (Electronics Industries Association : EIA) ได้วางมาตรฐานที่มีชื่อเรียกกันว่า EIA RS-232 มาตรฐานนี้ในช่วงแรกจะใช้คอนเน็กเตอร์เป็นแบบ DB-25 โดยกำหนดความยาวสูงสุดของสายสัญญาณไว้ที่ 50 ฟุต มีระดับสัญญาณตั้งแต่ -3 ถึง -12V แสดงว่ามีข้อมูล (Mark) และ +3 ถึง +12V แสดงว่าเป็นช่องว่าง (Space)

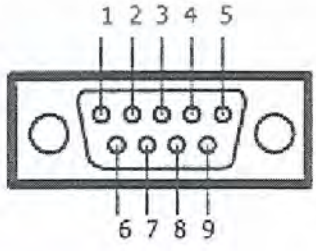
มาตรฐาน RS-232 ได้กำหนดรูปแบบของอุปกรณ์เชื่อมต่อข้อมูล (Data Terminal Equipment: DTE) กับวงจรข้อมูลปลายทาง (Data Circuit Terminating: DCE) ไว้ว่า อุปกรณ์ DTE จะต้องเป็นอุปกรณ์ที่มีการประมวลผลในตัว เช่น ไมโครคอนโทรลเลอร์หรือไมโครคอมพิวเตอร์ ซึ่งมีความสามารถในการสร้างบิตข้อมูลแบบอนุกรมได้ ส่วนอุปกรณ์ DCE จะทำหน้าที่เป็นเพียงตัวรับข้อมูลที่ส่งมาจาก DTE เท่านั้น โดยการรับส่งข้อมูลระหว่างอุปกรณ์ทั้งสองจะกระทำผ่านมาตรฐาน RS-232

ข้อแตกต่างของอุปกรณ์ DTE และอุปกรณ์ DCE อย่างหนึ่งที่เห็นได้ชัด คือ คอนเน็กเตอร์ของ DTE จะเป็นตัวผู้ ส่วนคอนเน็กเตอร์ของ DCE จะเป็นตัวเมีย ซึ่งพอร์ตอนุกรมของคอมพิวเตอร์ที่ใช้กันอยู่ทั่วไปจะเป็นแบบ DTE ส่วนคอนเน็กเตอร์ที่อยู่กับโมเด็มจะเป็นแบบ DCE

สำหรับการใช้งานบนคอมพิวเตอร์ พอร์ตอนุกรม RS-232 มันถูกใช้เชื่อมต่อกับโมเด็มหรือเมาส์ โดยสามารถรับส่งข้อมูลได้ที่ความยาวของสายสัญญาณสูงสุดถึง 20 เมตร

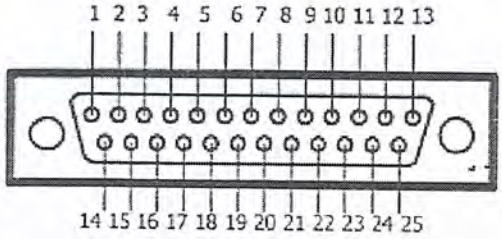
2.2.6.4 คอนเน็กเตอร์สำหรับพอร์ต RS-232 และการเชื่อมต่อ

มาตรฐานการเชื่อมต่อแบบ RS-232 จะใช้คอนเน็กเตอร์แบบ DB-25 ตัวผู้หรือ DB-9 ตัวผู้ ซึ่งคอนเน็กเตอร์แบบ DB-25 จะมีขาต่อใช้งานเพียง 9 เส้น เช่นเดียวกับคอนเน็กเตอร์แบบ DB-9 เนื่องจากขาอื่นๆ ที่เคยใช้งานในอดีต ปัจจุบันมีการใช้งานไม่มากนักจึงถูกยกเลิกไป โดยแสดงดังรูปร่างและตำแหน่งขาในรูปที่ 2.9



(ก) คอนเน็กเตอร์ 9 ขาหรือแบบ DB-9 (มองจากด้านหลังคอมพิวเตอร์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ข) คอนเน็กเตอร์อนุกรม 25 ขาหรือแบบ DB-25 (มองจากด้านหลังคอมพิวเตอร์)

คอนเน็กเตอร์ DB-9	คอนเน็กเตอร์ DB-25	ชื่อของสายสัญญาณ	ชนิดของสายสัญญาณ
1	8	Data Carrier Detect:DCD	อินพุต
2	3	Received Data:RxD	อินพุต
3	2	Transmitted Data:TxD	เอาต์พุต
4	20	Data Terminal Ready:DTR	เอาต์พุต
5	7	Signal Ground:GND	-
6	6	Data Set Ready:DSR	อินพุต
7	4	Request To Send:RTS	เอาต์พุต
8	5	Clear To Send:CTS	อินพุต
9	22	Ring Indicator:RI	อินพุต

รูปที่ 2.9 การจัดขาของคอนเน็กเตอร์พอร์ตอนุกรมตามมาตรฐาน RS-232 ทั้งแบบ DB-9 และ DB-25

รายละเอียดของสายสัญญาณ

1. Transmit Data : TD ใช้สำหรับส่งข้อมูลแบบอนุกรมออกจากคอมพิวเตอร์
2. Receive Data : RD ใช้สำหรับรับข้อมูลแบบอนุกรมออกจากคอมพิวเตอร์
3. Request To Send : RTS ใช้สำหรับส่งข้อมูลไปยังอุปกรณ์ปลายทาง เพื่อร้องขอให้อุปกรณ์ปลายทางส่งข้อมูลกลับมา
4. Clear To Send : CTS ใช้สำหรับตรวจสอบว่าอุปกรณ์ที่เชื่อมต่อพร้อมที่จะรับข้อมูลหรือไม่ โดยจะคอยรับสัญญาณ RTS เมื่อทุกอย่างพร้อมก็จะทำการส่งข้อมูลออกทาง TD
5. Data Set Ready : DSR ใช้สำหรับตรวจสอบการเชื่อมต่อกันระหว่างคอมพิวเตอร์กับอุปกรณ์ปลายทาง จะใช้คู่ขากับ DTR
6. Signal Ground : SG เป็นกราวด์ของระบบ
7. Carrier Detect : CD ขานี้จะแอกทีฟเมื่อมีการส่งสัญญาณพาห์จากโมเด็ม
8. Data Terminal Ready : DTR ใช้สำหรับบอกให้อุปกรณ์ปลายทางรับรู้ว่าต้องการติดต่อด้วยขา DTR นี้ต้องการเชื่อมต่อกับขา DSR ของอุปกรณ์ปลายทาง
9. Ring Indicator : RI ขานี้จะ Active เมื่อโมเด็มได้รับสัญญาณเรียกเข้าจากสายโทรศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกแสดงดังในรูปที่ 2.10 ลูกศรในรูปแสดงถึงทิศทางของข้อมูล ในรูปที่ 2.10 (ก) เป็นการเชื่อมต่อแบบโดยตรงโดยไม่ต้องผ่านโมเด็ม (Null modem) โดยมีการตรวจสอบหรือแฮนด์เช็กเต็มรูปแบบ ส่วนในรูปที่ 2.10 (ข) เป็นการเชื่อมต่อแบบโดยตรงโดยไม่ต้องผ่านโมเด็ม ในลักษณะใช้สายสัญญาณเพียง 3 เส้น โดยเส้นหนึ่งสำหรับส่งข้อมูล อีกเส้นสำหรับข้อมูลและเส้นสุดท้ายเป็นกราวด์ สำหรับรายละเอียดหน้าที่การทำงานในแต่ละขาของพอร์ตอนุกรม RS-232 มีดังนี้

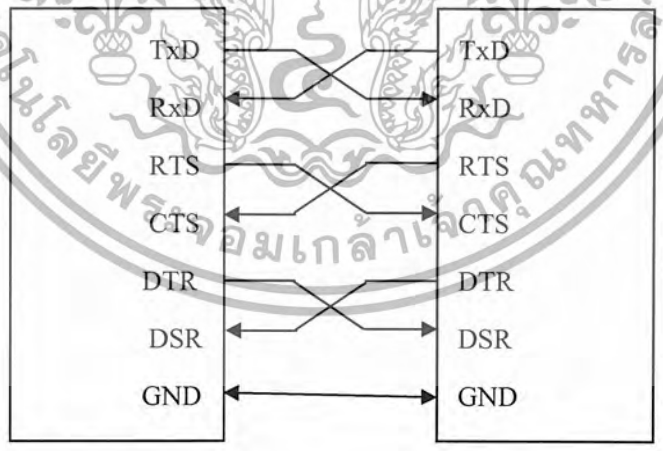
- Data Carrier Detect : DCD หรืออาจเรียกว่า Carrier Detect : CD ขานี้จะแอกทีฟเมื่อมีการส่งสัญญาณพาห์จากอุปกรณ์สื่อสารข้อมูล เช่น โมเด็ม สำหรับการใช้งานปกติ ขานี้จะไม่ได้ถูกใช้งานมากนัก

- Receive Data : RD หรือ RxD ขานี้ใช้เพื่อรับสัญญาณอนุกรมเข้ามายังคอมพิวเตอร์โดยนำข้อมูลที่อ่านได้เก็บไว้ในรีจิสเตอร์บัฟเฟอร์

- Transmitted Data : TD หรือ TxD ขานี้ใช้เพื่อส่งข้อมูลออกจากคอมพิวเตอร์ โดยนำข้อมูลที่เก็บอยู่ในบัฟเฟอร์สำหรับส่งข้อมูลส่งออกไป

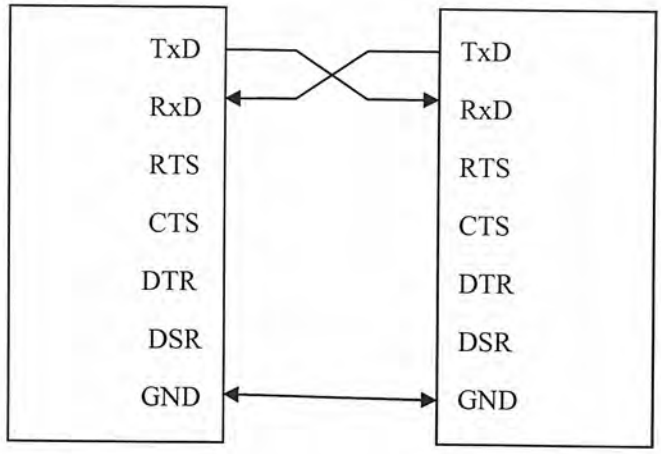
- Data Terminal Ready : DTR เป็นขาสัญญาณที่ส่งออกจากคอมพิวเตอร์เพื่อให้อุปกรณ์ปลายทางรับรู้ว่าต้องการติดต่อด้วย โดยขา DTR นี้จะต้องเชื่อมต่อกับขา DSR ของอุปกรณ์ปลายทางและขา DTR ของอุปกรณ์ปลายทางจะต้องเชื่อมต่อกับขา DSR ของคอมพิวเตอร์ ถ้าใช้การเชื่อมต่อเป็นแบบโดยตรงโดยไม่ต้องผ่านโมเด็ม ซึ่งใช้สายในการเชื่อมต่อเพียง 3 เส้น จะต้องต่อขา DTR และ DSR ของตัวมันเองเข้าด้วยกันและต้องต่อกับขา DCD ด้วยในกรณีที่โปรแกรมสื่อสารที่ใช้มีการตรวจจับสัญญาณพาห์

- Signal Ground : GND ขากราวด์ของระบบ



(ก) การต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์ แบบ Null modem

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ข) การต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์ แบบ RS-232 โดยใช้สายสัญญาณเพียง 3 เส้น

รูปที่ 2.10 การต่ออุปกรณ์ภายนอกกับพอร์ตอนุกรมของคอมพิวเตอร์ในลักษณะต่างๆ

- Data Set Ready : DSR ขานี้จะใช้คู่กับขา DTR เพื่อการตรวจสอบการเชื่อมต่อกันระหว่างคอมพิวเตอร์กับอุปกรณ์ปลายทาง ซึ่งขา DSR นี้จะเป็นขาสำหรับรับข้อมูลจากภายนอกซึ่งถูกส่งมาจากขา DTR

- Request To Send : RTS เป็นขาสำหรับส่งสัญญาณร้องขอให้ทางอุปกรณ์ปลายทางส่งข้อมูลกลับมายังคอมพิวเตอร์ โดยขาที่รับสัญญาณ RTS ก็คือขา CTS ในกรณีที่ใช้การเชื่อมต่อแบบโดยตรงโดยไม่ต้องผ่านโมเด็ม 3 สาย จะต้องเชื่อมต่อขา RTS และ CTS ของตัวมันเองเข้าด้วยกันเพื่อจะให้การรับและส่งข้อมูลสามารถเกิดขึ้นได้ตลอดเวลา

- Clear To Send : CTS ขานี้จะคอยรับสัญญาณจากขา RTS เมื่อรับสัญญาณได้ ข้อมูลที่ขา TxD จะถูกส่งออกไป ดังนั้นขานี้จึงถูกใช้เพื่อตรวจสอบอุปกรณ์ต่อพ่วงว่าพร้อมที่จะรับข้อมูลหรือไม่

- Ring Indicator : RI ใช้แสดงสถานะสัญญาณเรียกจากสายโทรศัพท์ ปกติในการสื่อสารโดยทั่วไปสายนี้จะถูกใช้งานก็ต่อเมื่อมีการเชื่อมต่อกับโมเด็มและโปรแกรมมีการตรวจสอบสัญญาณนี้เท่านั้น

2.2.6.5 UART

UART มาจากคำว่า Universal Asynchronous Receiver Transmitter ซึ่งหมายถึงอุปกรณ์ที่ทำหน้าที่รับและส่งข้อมูลแบบอะซิงโครนัสนั่นเอง สำหรับการสื่อสารอนุกรมบนคอมพิวเตอร์แล้ว UART ถือเป็นหัวใจสำคัญของการสื่อสารอนุกรม

หน้าที่หลักของ UART คือทำหน้าที่แปลงข้อมูลที่อยู่ในรูปแบบขนานจากคอมพิวเตอร์ให้อยู่ในรูปแบบอนุกรมแบบอะซิงโครนัส แล้วส่งออกไป และทำหน้าที่แปลงสัญญาณอนุกรมแบบอะซิงโครนัสที่ป้อนเข้ามายัง UART ให้เป็นแบบขนานก่อนที่จะส่งเข้าสู่คอมพิวเตอร์ ซึ่งนอกจาก UART จะส่งข้อมูลไปยังคอมพิวเตอร์แล้ว ยังแจ้งข้อมูลอื่นๆให้คอมพิวเตอร์รับทราบด้วย เช่นอัตราเร็วในการรับส่งข้อมูล (อัตรา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บอด), รูปแบบการส่งข้อมูล, ความผิดพลาดที่เกิดขึ้นระหว่างการถ่ายทอดข้อมูล (ผิดพลาดจากพาริตี้, เฟรมข้อมูล, โอเวอร์รัน) เป็นต้น

ภายใน UART จะมีส่วนของวงจรสร้างอัตราบอดแบบโปรแกรมได้ (Programmable Baudrate Generator) โดยการกำหนดค่าตัวหารให้กับสัญญาณนาฬิกาของ UART โดยตัวหารนี้มีขนาด 16 บิต ดังนั้นจึงสามารถกำหนดตัวหารอยู่ในช่วง 1–65,535 UART สามารถรับส่งข้อมูลได้ทั้งแบบฮาล์ฟดูเพล็กซ์ (half duplex) และฟูลดูเพล็กซ์ (full duplex) โดยการส่งแบบฮาล์ฟดูเพล็กซ์เป็นการส่งแบบทิศทางเดียว ส่วนการส่งแบบฟูลดูเพล็กซ์นั้นสามารถรับและส่งข้อมูลได้ในคราวเดียวกัน

ชนิดของ UART

ในเครื่องคอมพิวเตอร์ทั่วไปมี UART ที่ใช้งานกันอยู่ 2 หมายเลข คือ 8250 ซึ่งเป็น UART มาตรฐานที่มีใช้กันมายาวนาน UART หมายเลขนี้จะมีบัฟเฟอร์สำหรับรับและส่งข้อมูลตำแหน่งเดียวกัน ทำให้การรับและส่งข้อมูลถูกจำกัดความเร็วอยู่ที่ 57.6 กิโลบิตต่อวินาทีเท่านั้น แต่ UART หมายเลขนี้ก็ถือเป็นต้นแบบของ UART ที่ใช้ในคอมพิวเตอร์โดยคอมพิวเตอร์ทุกๆรุ่นจะต้องสนับสนุนการทำงานตามรูปแบบของ UART หมายเลขนี้

UART อีกหมายเลขหนึ่ง คือ 16450 มีความสามารถรับส่งข้อมูลได้ที่ความเร็ว 115,200 บิตต่อวินาทีและเพิ่มรีจิสเตอร์สำหรับพักข้อมูลสำหรับ UART นอกจากนี้ยังเพิ่มส่วนของชิฟต์รีจิสเตอร์แบบ FIFO (First In First Out) ขนาด 16 ไบต์เข้าไป ทำให้สามารถสนับสนุนความเร็วในการรับส่งข้อมูลที่ 256 กิโลบิตต่อวินาทีได้ โดยคอมพิวเตอร์ในปัจจุบันใช้ UART หมายเลขนี้หรือใหม่กว่า เช่น หมายเลข TL16C750 ซึ่งมีรีจิสเตอร์แบบ FIFO ขนาด 64 ไบต์ ทำงานได้ที่ระดับแรงดัน +5V และ +3V มีโหมดประหยัดพลังงานสามารถรับส่งข้อมูลได้ที่ความเร็ว 1 เมกะบิตต่อวินาที เมื่อใช้สัญญาณนาฬิกา 16 MHz

อย่างไรก็ตาม ความเร็วในการส่งข้อมูลที่มากมายของ UART หมายเลขใหม่ๆก็ไม่ได้ช่วยให้การรับส่งข้อมูลของคอมพิวเตอร์เร็วขึ้น เนื่องจากว่าคอมพิวเตอร์ยังใช้ความถี่ของสัญญาณนาฬิกาในการแปลงข้อมูลเพียง 1.8432 MHz เท่านั้น

2.2.6.6 MCS-51 กับการรับส่งข้อมูลแบบอนุกรม

พอร์ตอนุกรมของ MCS-51 จะใช้ขา TxD และ RxD ในการรับส่งข้อมูลโดยขาทั้งสองจะอยู่ในพอร์ต 3 คือ P3.1 เป็น TxD และ P3.0 เป็น RxD พอร์ตอนุกรมของ MCS-51 สามารถทำงานแบบฟูลดูเพล็กซ์ได้ คือสามารถส่งและรับข้อมูลในเวลาเดียวกันได้ โดยในการรับและส่งข้อมูลจะมีบัฟเฟอร์สำหรับเก็บข้อมูลให้ใช้

รีจิสเตอร์ที่เกี่ยวข้องกับการทำงานของพอร์ตอนุกรมใน MCS-51

ในการทำงานของพอร์ตอนุกรมในไมโครคอนโทรลเลอร์ MCS-51 มีรีจิสเตอร์ที่เกี่ยวข้องอยู่ 2 ตัว ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- รีจิสเตอร์บัฟเฟอร์ของพอร์ตอนุกรมหรือ SBUF (Serial data buffer register)

มีแอดเดรสอยู่ที่ 99H ในพื้นที่ของรีจิสเตอร์ฟังก์ชั่นพิเศษหรือ SFR (Special Function Register) มีขนาด 8 บิต แบ่งเป็น 2 ส่วน คือ รีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูล (transmit buffer register) และรับข้อมูล (receive buffer register) เมื่อมีการเขียนข้อมูลมายังรีจิสเตอร์ SBUF ข้อมูลนั้นจะถูกส่งต่อไปยังบัฟเฟอร์สำหรับส่งข้อมูล เพื่อส่งออกจากไมโครคอนโทรลเลอร์ผ่านทางขา TxD หรือขา P3.1 ในกรณีที่มีการอ่านข้อมูลจากรีจิสเตอร์ SBUF ข้อมูลจะถูกส่งผ่านไปยังรีจิสเตอร์บัฟเฟอร์สำหรับข้อมูลเพื่อส่งต่อไปยังไมโครคอนโทรลเลอร์ต่อไป สำหรับการรับข้อมูลอนุกรมจากภายนอกนั้นจะผ่านมาทางขา RxD หรือ P3.0 ของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

- รีจิสเตอร์ควบคุมการทำงานของพอร์ตอนุกรมหรือ SCON (Serial Port Control Register)

เป็นรีจิสเตอร์ขนาด 8 บิต มีแอดเดรสอยู่ที่ 98H ในพื้นที่ของรีจิสเตอร์ SFR สามารถเข้าถึงได้ในระดับบิต มีรายละเอียดการทำงานดังนี้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

รูปที่ 2.11 รีจิสเตอร์ควบคุมการทำงานของพอร์ตอนุกรม

SM0-SM1 (Serial port mode bit 0-1): ใช้ในการเลือกโหมดการทำงานของพอร์ตอนุกรมภายในไมโครคอนโทรลเลอร์ MCS-51 ดังมีรายละเอียดต่อไปนี้

SM2: ใช้ในการกำหนดการสื่อสารในแบบมัลติโปรเซสเซอร์ (multiprocessor) ในการทำงานของโหมด 2 และ 3 ของพอร์ตอนุกรมในไมโครคอนโทรลเลอร์ MCS-51 ถ้าบิตนี้เป็น "1" บิต RI จะไม่แอกทีฟ ถ้าบิตที่ 9 ที่รับเข้ามาเป็น "0" (ข้อมูลบิตที่ 9 เก็บไว้ที่บิต RB8) ในการทำงานโหมด 1 ถ้าบิตนี้เซ็ต บิต RI จะไม่แอกทีฟ ถ้ายังไม่ได้รับบิตหยุด ส่วนในโหมด 0 บิตนี้ไม่มีการใช้งาน

REN (Enable serial reception): ใช้ในการเอ็นเอเบิลการรับข้อมูลของพอร์ตอนุกรม ทำการเซ็ตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์ ถ้าต้องการให้มีการรับข้อมูลต้องเซ็ตบิตนี้ให้เป็น "1"

TB8: ใช้สำหรับเก็บข้อมูลบิตที่ 9 ที่ต้องการทำการส่งออกไปในการทำงานโหมด 2 และ 3 ของพอร์ตอนุกรมในไมโครคอนโทรลเลอร์ MCS-51 เซ็ตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

RB8: ใช้สำหรับรับข้อมูลบิตที่ 9 ที่เข้ามาในการทำงานโหมด 2 และ 3 ของพอร์ตอนุกรมในไมโครคอนโทรลเลอร์ MCS-51 แต่ถ้าหากว่าพอร์ตอนุกรมทำงานอยู่ในโหมด 1 และบิต SM2 เป็น "0" ข้อมูลที่บิต RB8 คือ ข้อมูลของบิตหยุด (Stop Bit) สำหรับในการทำงานโหมด 0 บิตนี้จะไม่ใช้งานบิต RB8 นี้สามารถเซ็ตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

TI (Transmit Interrupt flag): ใช้ในการแสดงการเกิดอินเตอร์รัปต์เมื่อมีการส่งข้อมูลออกจากพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51 สามารถเซ็ตได้ด้วยกระบวนการทางฮาร์ดแวร์ เมื่อทำการส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลบิตที่ 8 ไปเรียบร้อยแล้วในการทำงานของโหมด 0 ส่วนในการทำงานโหมดอื่น บิตนี้จะเซ็ตเมื่อมีการเริ่มต้นส่งบิตหยุดออกไป การเคลียร์บิตนี้จะต้องใช้กระบวนการทางซอฟต์แวร์เท่านั้น

RI (Receive Interrupt Flag): ใช้แสดงการเกิดอินเตอร์รัปต์เมื่อมีการรับข้อมูลเข้าสู่พอร์ตอนุกรม สามารถเซ็ตได้ด้วยกระบวนการทางฮาร์ดแวร์ เมื่อทำการรับข้อมูลบิตที่ 8 เรียบร้อยแล้วในการทำงานโหมด 0 ส่วนในการทำงานโหมดอื่น บิตนี้จะเซ็ตเมื่อมีการรับบิตหยุดของข้อมูลอนุกรมไปได้ครึ่งทางแล้ว ยกเว้นในกรณีที่บิต SM2 มีการเซ็ต บิตนี้จะเซ็ตได้ก็ต่อเมื่อการรับบิตหยุดหรือบิตที่ 9 เกิดขึ้นอย่างสมบูรณ์แล้ว การเคลียร์บิตนี้ต้องใช้กระบวนการทางซอฟต์แวร์เท่านั้น

โหมดการทำงานของพอร์ตอนุกรมใน MCS-51

พอร์ตอนุกรมในไมโครคอนโทรลเลอร์ MCS-51 สามารถเลือกการทำงานได้ถึง 4 โหมด คือ

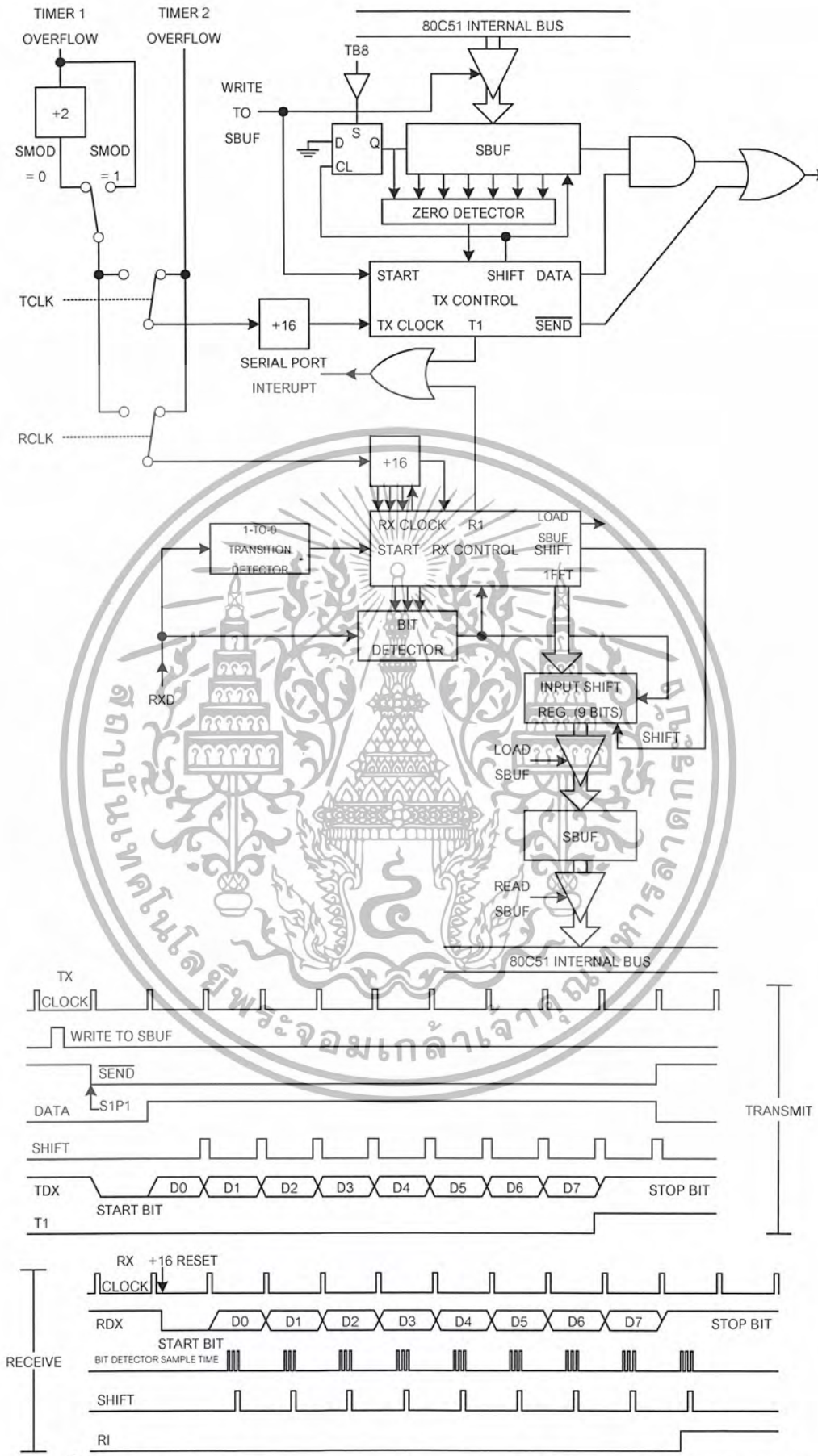
1. โหมด 0 เป็นการกำหนดให้พอร์ตอนุกรมทำงานในลักษณะชิฟต์รีจิสเตอร์
2. โหมด 1 เป็นการกำหนดให้เป็น UART ขนาด 8 บิต สามารถเลือกอัตราบอดได้
3. โหมด 2 เป็นการกำหนดให้เป็น UART ขนาด 9 บิต โดยมีอัตราบอดคงที่
4. โหมด 3 เป็นการกำหนดให้เป็น UART ขนาด 9 บิต สามารถเลือกอัตราบอดได้

การเลือกโหมดทำได้โดยการกำหนดข้อมูลให้แก่บิต SM0 และ SM1 ในรีจิสเตอร์ SCON

การทำงานในโหมด 1 ของวงจรพอร์ตอนุกรม

โครงงานนี้ได้เลือกใช้การติดต่อสื่อสารในโหมด 1 ซึ่งมีไดอะแกรมแสดงดังในรูปที่ 2.12 ในโหมดนี้ใช้ในการรับส่งข้อมูลรวม 10 บิต โดยส่งข้อมูลออกทางขา P3.1 หรือ TxD และรับข้อมูลเข้าทางขา P3.0 หรือ RxD ข้อมูลทั้ง 10 บิตประกอบด้วย บิตเริ่มต้น (มีค่าเป็น “0”) 1 บิต บิตข้อมูล 8 บิต โดยรับหรือส่งข้อมูลในบิต LSB ก่อน และบิตหยุดหรือบิตปิดท้าย (มีค่าเป็น “1”) ในการรับข้อมูล บิตหยุดจะถูกเก็บไว้ในบิต RB8 ในรีจิสเตอร์ SCON อัตราบอดในโหมดนี้ได้รับการกำหนดโดยอัตราการเกิดโอเวอร์โฟลว์ของไทเมอร์ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.12 ผังการทำงานในโหมด 1 ของพอร์ตอนุกรมภายในไมโครคอนโทรลเลอร์ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อัตราบอดของพอร์ตอนุกรมในไมโครคอนโทรลเลอร์ MCS-51

ในโหมดนี้สามารถเลือกแหล่งกำเนิดอัตราบอดได้ 2 แหล่ง คือ จากอัตราการโอเวอร์โฟลว์ของไทเมอร์ 1 และ 2 สำหรับอัตราบอดเมื่อใช้การโอเวอร์โฟลว์ของไทเมอร์ 1 จะต้องใช้ค่าของบิต SMOD ในรีจิสเตอร์ PCON มาพิจารณาประกอบด้วย สามารถคำนวณค่าอัตราบอดได้จาก

$$\text{อัตราบอด} = *(2^{\text{ค่าของบิต SMOD}}/32) * \text{อัตราการโอเวอร์โฟลว์ของไทเมอร์ 1}$$

ถ้าหากในไทเมอร์ 1 ไม่ได้กำหนดค่าการอินเตอร์รัปต์ไว้ สามารถคำนวณค่าอัตราบอดได้จาก

$$\text{อัตราบอด} = *(2^{\text{ค่าในรีจิสเตอร์ SMOD}}/32) * \{\text{ความถี่สัญญาณนาฬิกา} / [12 * (256-(TH1))]$$

ในตารางที่ 2.5 แสดงการกำหนดอัตราบอดโดยใช้ไทเมอร์ 1

ในกรณีที่ใช้ไทเมอร์ 2 ในการกำหนดอัตราบอด โดยกำหนดให้ไทเมอร์ 2 ทำงานในโหมดกำเนิดอัตราบอด (buad rate generator) สามารถคำนวณหาค่าอัตราบอดได้จาก

$$\text{อัตราบอด} = \text{อัตราการโอเวอร์โฟลว์ของไทเมอร์ 2} / 16 \text{ หน่วยเป็นบิตต่อวินาที}$$

ถ้าหากกำหนดให้ไทเมอร์ 2 ทำงานในโหมดปกติ สามารถคำนวณหาค่าอัตราบอดได้จาก

$$\text{อัตราบอด} = \text{ความถี่ของสัญญาณนาฬิกา} / (32 * (65536-(RCAP2H, RCAP2L)))$$

โดยที่ (RCAP2H, RCAP2L) เป็นค่าของรีจิสเตอร์ RCAP2H และ RCAP2L มีขนาด 16 บิตไม่คิดเครื่องหมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

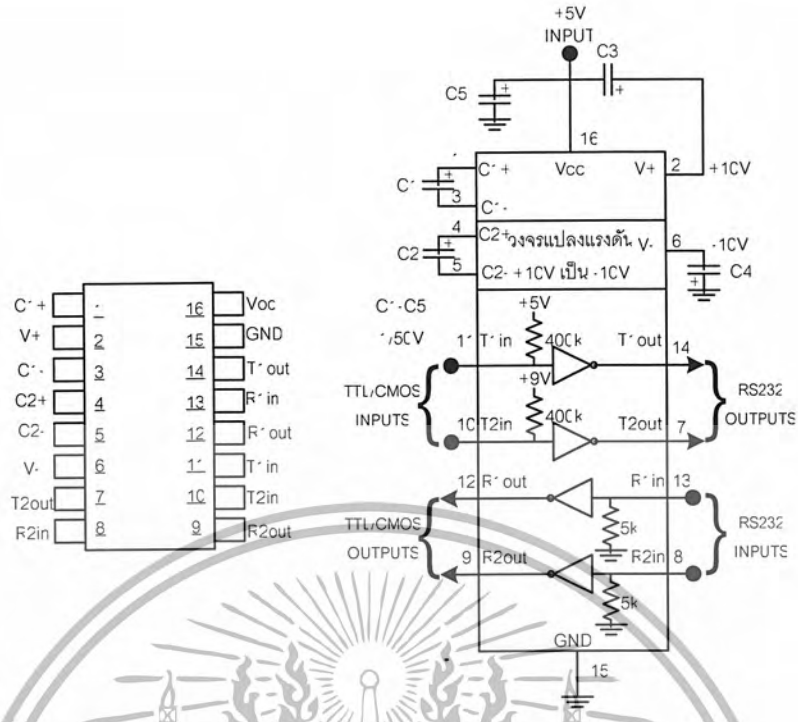
อัตราบอด (บิตต่อวินาที : bps)	ความถี่สัญญาณนาฬิกา	SMOD	ไทเมอร์ 1		
			C/T	Mode	ค่ารีโหลด
โหมด 0 : สูงสุด 1MHz	12 MHz	X	X	X	X
โหมด 2 : สูงสุด 375K	12 MHz	1	X	X	X
โหมด 1,3 : 62.5K	12 MHz	1	0	2	FFH
19.2 K (19,200)	11.0592MHz	1	0	2	FDH
9.6K (9600)	11.0592MHz	0	0	2	FDH
4.8K (4800)	11.0592MHz	0	0	2	FAH
2.4K (2400)	11.0592MHz	0	0	2	F4H
1.2K (1200)	11.0592MHz	0	0	2	E8H
137.5	11.0592MHz	0	0	2	1DH
110	6 MHz	0	0	2	72H
110	12 MHz	0	0	1	FEEBH

ตารางที่ 2.5 การเลือกอัตราพอร์ตของวงจรพอร์ตอนุกรมภายในไมโครคอนโทรลเลอร์ MCS-51

การเชื่อมต่อกับพอร์ตอนุกรมของคอมพิวเตอร์

การใช้งานวงจรพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51 มักนิยมใช้ในการติดต่อเพื่อแลกเปลี่ยนข้อมูลกับคอมพิวเตอร์ผ่านทางพอร์ตอนุกรมในมาตรฐาน RS-232 เป็นส่วนใหญ่ แต่เนื่องจากระดับสัญญาณของพอร์ตอนุกรม RS-232 มีระดับตั้งแต่ +3 ถึง +12V และ -3 ถึง -12V ในขณะที่ระดับสัญญาณของไมโครคอนโทรลเลอร์ MCS-51 อยู่ในระดับทีทีแอล ดังนั้นจึงไม่สามารถเชื่อมต่อกับพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51 เข้ากับพอร์ตอนุกรมของคอมพิวเตอร์ได้โดยตรง จึงต้องอาศัยการเชื่อมต่อผ่านไอซีพิเศษที่ทำหน้าที่ในการแปลงระดับสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

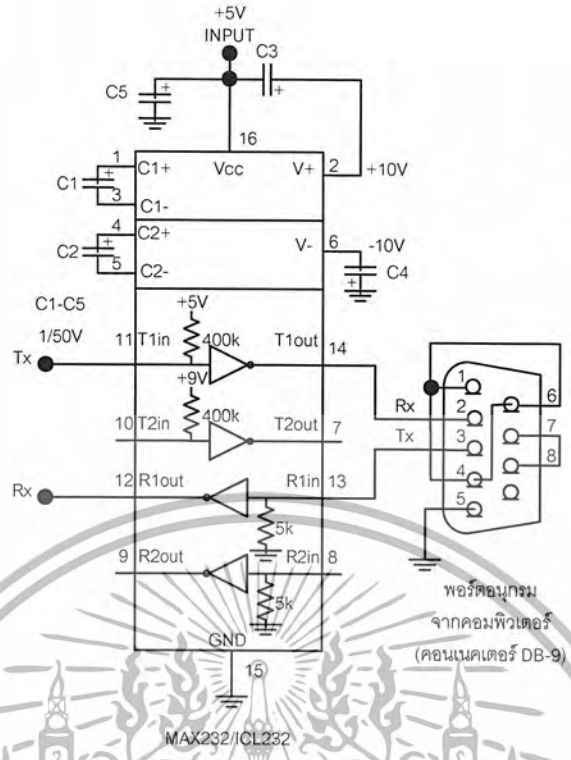


(ก) การจัดขาของ MAX232 หรือ ICL232 (ข) โครงสร้างภายในของ MAX232 หรือ ICL232

รูปที่ 2.13 รายละเอียดเบื้องต้นของไอซีแปลงสัญญาณเพื่อเชื่อมต่อกับพอร์ตอนุกรมของคอมพิวเตอร์

ไอซีที่ทำหน้าที่แปลงระดับสัญญาณนี้ จะต้องทำการแปลงข้อมูลส่งของไมโครคอนโทรลเลอร์ MCS-51 จากระดับทีทีแอลไปเป็นระดับของ RS-232 และทำการแปลงข้อมูลรับจากคอมพิวเตอร์จากระดับของ RS-232 เป็นระดับทีทีแอลเพื่อให้สามารถถ่ายทอดไปยังไมโครคอนโทรลเลอร์ MCS-51 ได้ อย่างสมบูรณ์ ไอซีดังกล่าวมีด้วยกันหลายหมายเลขจากหลายผู้ผลิต อาทิ MAX232 จาก MAXIM หรือ ICL232 จาก HARRIS เป็นต้น ในรูปที่ 2.13 แสดงการจัดขาของไอซี ICL232 ซึ่งใช้ในการแปลงสัญญาณ RS-232 ส่วนวงจรของการต่อกับไมโครคอนโทรลเลอร์ MCS-51 แสดงในรูปที่ 2.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

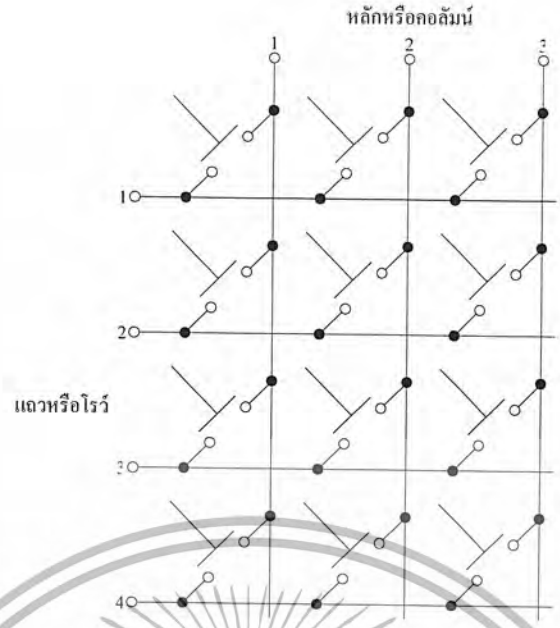


รูปที่ 2.14 วงจรเชื่อมต่อ MAX232 หรือ ICL232
เข้ากับพอร์ตอนุกรมของคอมพิวเตอร์และไมโครคอนโทรลเลอร์ MCS-51

2.3 การเชื่อมต่อกับคีย์แพด (Keypad)

การอ่านค่าหรือการรับค่าการกดสวิตช์มีด้วยกัน 2 ลักษณะใหญ่ๆ คือ ต่อแบบเข้ากับไฟเลี้ยงหรือกราวด์โดยตรง และต่อแบบวงจรเมตริกซ์ (Matrix Switch) โดยในโครงงานนี้จะใช้การต่อสวิตช์แบบเมตริกซ์ ดังแสดงในรูปที่ 2.15 ซึ่งสวิตช์จะถูกต่อกันในแนวแกนตั้งและแกนนอน จะเรียกแนวตั้งว่าหลักหรือคอลัมน์ (Column) ในขณะที่แนวนอนจะเรียกว่าแถวหรือโรว์ (Row) ดังนั้น ค่าของสวิตช์จะต้องประกอบด้วยตำแหน่งในแนวหลักและแถว กระบวนการซึ่งจะทำให้ได้มาซึ่งค่าของสวิตช์นั้นมีขั้นตอนที่ซับซ้อนพอสมควร แต่วงจรของสวิตช์แบบเมตริกซ์นี้มีข้อดีคือสามารถรองรับการเพิ่มของสวิตช์ได้อย่างสะดวก นั่นคือเพียงเพิ่มเติมจำนวนสวิตช์และแก้ไขซอฟต์แวร์อีกเล็กน้อยเท่านั้น ทำให้วงจรสวิตช์แบบเมตริกซ์เป็นที่นิยมใช้กันมาก ในระบบควบอัตโนมัติหรือกึ่งอัตโนมัติที่มีจำนวนสวิตช์มากกว่า 8 ตัว ในการใช้งานทั่วไปจะเรียกสวิตช์แบบเมตริกซ์นี้ว่า คีย์แพด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 วงจรของสวิตซ์แบบเมตริกซ์หรือคีย์แพด

การเชื่อมต่อคีย์แพดเข้ากับไมโครคอนโทรลเลอร์ MCS-51

จากโครงงาน เราจะทำการเชื่อต่อคีย์แพดเข้ากับไมโครคอนโทรลเลอร์ทางพอร์ต P1 โดยใช้สายทั้งหมด 7 เส้น เป็นสายของหลัก 3 เส้น และสายของแถว 4 เส้น โดยที่ขาพอร์ต P1.0-P1.3 จะต้องต่อตัวต้านทานพูลอัพไว้เพื่อกำหนดสภาวะเริ่มต้นที่ไม่มีการกดคีย์แพด ทุกครั้งที่มีการส่งข้อมูลไปยังสายหลักของคีย์แพด ไมโครคอนโทรลเลอร์จะทำการส่งข้อมูล "0" ไปยังขาพอร์ต P1.4-P1.6 พร้อมกันนั้นไมโครคอนโทรลเลอร์จะทำการอ่านค่าที่เข้ามาทางขาพอร์ต P1.0-P1.3 ด้วย หากไม่มีการกดคีย์แพด ค่าของ P1.0-P1.3 ก็จะเป็น "1" ทั้งหมด แต่ถ้ามีการกดคีย์แพด ค่าของ P1.0-P1.3 ก็จะไม่เป็น "1" ทั้งหมดอีกต่อไป เพื่อเป็นการแจ้งให้ทราบว่ามีการกดคีย์แพดเกิดขึ้นแล้ว จากนั้นไมโครคอนโทรลเลอร์ก็จะทำการค้นหาตำแหน่งต่อไป โดยการค้นหาตำแหน่งนั้น สิ่งที่ได้มาอย่างแรกคือค่าตำแหน่งของคีย์นั้น จากนั้นนำค่าตำแหน่งที่ได้ไปเปิดเทียบตารางข้อมูล เพื่อจะได้นำค่าที่ต้องการไปแสดงผล

2.4 VC++.NET

VC++.NET สามารถใช้ในการพัฒนาโปรแกรมเพื่อสร้างแอพพลิเคชั่นได้ทุกรูปแบบ ไม่ว่าจะเป็นโปรแกรมที่ปฏิบัติการบน Windows เรื่อยไปจนถึงโปรแกรมที่ทำงานโดยอาศัยแนวคิดของ .NET แต่แนวทางที่เหมาะสมคือ การพัฒนาโปรแกรมในแบบดั้งเดิม หรือเรียกว่า Native Code หรือ Unmanaged Code ซึ่งมีข้อดีคือ โปรแกรมที่ได้จะมีขนาดเล็ก ทำงานได้อิสระและรวดเร็วมาก

เราสามารถนำคุณสมบัตินี้มาใช้ในการติดต่อสื่อสารระหว่างอุปกรณ์กับเครื่องคอมพิวเตอร์ โดยพัฒนาโปรแกรม VC++.NET เพื่อรับข้อมูลอนุกรมที่มาจากพอร์ตอนุกรม แล้วใช้ในการควบคุมการจับภาพของกล้องโทรทัศน์วงจรปิด และยังใช้งานในส่วนของการนำข้อมูลไปเก็บไว้ยังฐานข้อมูลอีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1 การจัดเก็บข้อมูล

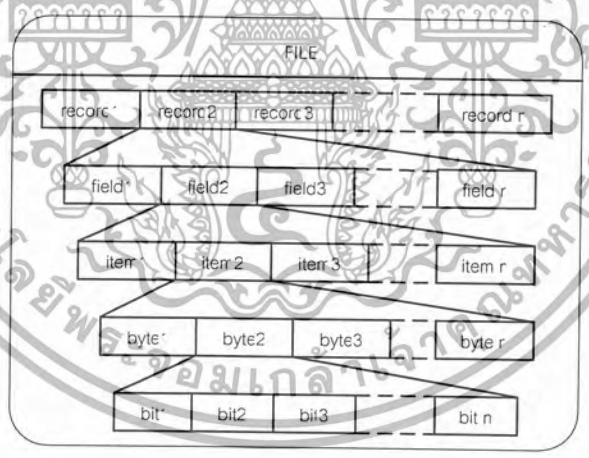
เป็นการบันทึกข้อมูลรูปภาพและข้อมูลหมายเลขโทรศัพท์โดยการสร้างแอพพลิเคชั่นแบบ SDI ที่ใช้ CFromview ในโปรแกรม VC++.NET นั่นเอง ซึ่งจะเป็นการสร้างโปรแกรมที่สามารถบันทึกข้อมูลในลักษณะคล้ายกับฟอร์มฐานข้อมูล แต่จะเก็บไว้ในฮาร์ดดิสก์ นอกจากนี้เราได้เพิ่มฟังก์ชันที่จำเป็นต่อการใช้งาน เช่น การเลื่อนเรคคอร์ดไปข้างหน้า ถอยหลัง แรกสุด และสุดท้าย พร้อมกับการเพิ่ม การลบและการอัพเดตเรคคอร์ดอีกด้วย โดยใช้คลาส CObList (ข้อมูล) ร่วมกับคลาส CFromview

ระบบฐานข้อมูล

ฐานข้อมูลคือ กลุ่มข้อมูลที่เป็นข้อเท็จจริงที่ถูกนำมาเก็บรวบรวมไว้ในที่เดียวกันอย่างเป็นระบบเพื่อนำไปใช้ในวัตถุประสงค์อย่างใดอย่างหนึ่งโดยกลุ่มผู้ใช้ตั้งแต่หนึ่งกลุ่มขึ้นไป ข้อมูลเหล่านี้อาจะเป็นข้อเท็จจริงที่เกี่ยวกับบุคคล สิ่งของ สถานที่ หรือเหตุการณ์ใดๆ ซึ่งเป็นได้ทั้งตัวเลข ข้อความ รูปภาพ หรืออื่นๆ

ลักษณะของฐานข้อมูลประกอบด้วย

- ข้อมูลทั้งหมดจะต้องถูกเก็บรวบรวมไว้ด้วยกัน
- จะต้องมีการจัดการข้อมูลนั้นอย่างเป็นระบบ
- ต้องสามารถนำข้อมูลนั้นไปใช้ได้ตามต้องการ



รูปที่ 2.16 แสดงโครงสร้างข้อมูลที่นำมาประมวลผลด้วยคอมพิวเตอร์

จากรูปที่ 2.16 สามารถอธิบายได้ดังนี้

- **บิต (Bit)** เป็นหน่วยข้อมูลที่เล็กที่สุด ที่แทนด้วยเลขฐานสอง (0 หรือ 1)

- **ไบต์ (Byte)** คือ กลุ่มของบิตที่แทนตัวอักษร ตัวเลข หรือสัญลักษณ์พิเศษหนึ่งตัว (character) เช่น รหัส ASCII 1 ไบต์ ซึ่งเก็บบิต 01000001 จะหมายถึงตัวอักษร A

- **ไอเท็ม (Item)** คือ ข้อมูลที่เกิดจากตัวอักษร ตัวเลข หรือสัญลักษณ์พิเศษมาเรียงต่อกันและมีความหมาย เช่น Smith แทนชื่อคน, Bangkok แทนชื่อจังหวัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- คอลัมน์ (Column) คือ ข้อมูลที่ประกอบด้วยไอเท็มตั้งแต่ 1 ไอเท็มขึ้นไป เช่น ไอเท็มวัน เดือน และปี รวมกันเป็นคอลัมน์วันเกิด หรือไอเท็มชื่อและนามสกุลรวมกันเป็นคอลัมน์ชื่อ-นามสกุล เป็นต้น คอลัมน์ที่ประกอบด้วยไอเท็มตั้งแต่ 2 ไอเท็มจะเป็นคอลัมน์แบบ Group Item ถ้าประกอบด้วยหนึ่งไอเท็มจะเป็นคอลัมน์แบบ Elementary item ตัวอย่างเช่น คอลัมน์วันเกิดเป็น Group Item เนื่องจากสามารถแยกข้อมูลลงไปเป็นวัน เดือน และปีเกิดได้ ในขณะที่ วัน เดือน และปีเกิดเป็น Elementary Item เนื่องจากแยกย่อยลงไปอีกไม่ได้ ถ้าแยกย่อยลงไปจะไม่ได้ข้อมูลที่สื่อความหมาย

- แถว (Row) คือ กลุ่มของคอลัมน์ที่มีความสัมพันธ์กัน เช่น แถวพนักงานประกอบด้วยคอลัมน์ รหัสพนักงาน ชื่อ นามสกุล แผนก ตำแหน่ง สถานภาพสมรส วันเข้าทำงาน ที่อยู่และอื่นๆ เป็นต้น หนึ่งแถวจะเก็บข้อมูลพนักงาน 1 คน โดยคอลัมน์เดียวกันในทุกๆแถวจะต้องเก็บข้อมูลชนิดเดียวกัน เช่น คอลัมน์ชื่อและนามสกุล จะเก็บข้อมูลแบบตัวอักษรเท่านั้น

-ไฟล์ (File) หรือ แฟ้มข้อมูล คือกลุ่มแถวที่ข้อมูลเป็นเรื่องเดียวกัน เช่น แฟ้มข้อมูลพนักงาน แฟ้มข้อมูลลูกค้า แฟ้มข้อมูลการสั่งซื้อสินค้า แฟ้มข้อมูลสินค้า เป็นต้น

ชื่อ-นามสกุล	ที่อยู่	โทรศัพท์
กุลสิริ ศิริพงษ์	130/11 ถนนพระราม 1 จังใหม่ ประทุมวัน	248-147
จันทนา ปรงรัตน์	53 ถนนลาดหญ้า สมเด็จ คลองสาน	455-444
อัญชลี ชัยรัตน์	36 ซอยสะพานใหม่ พหลโยธิน บางเขน	525-900
ศรราม ไทยแท้	1399 ถนนพิบูลสงคราม นนทบุรี	526-900
ปัทมา เพียรกิจ	183 ถนนศรีนครินทร์ อ.เมือง สมุทรปราการ	753-000

ตารางที่ 2.6 ตัวอย่างแฟ้มข้อมูลชื่อ นามสกุล ที่อยู่และหมายเลขโทรศัพท์

2.5 ความรู้เบื้องต้นเกี่ยวกับจดหมายอิเล็กทรอนิกส์

จดหมายอิเล็กทรอนิกส์หรือที่เรียกทั่วไปว่าอีเมล์ (electronics mail or e-mail) เป็นวิธีการส่งข้อความผ่านทางคอมพิวเตอร์ไปยังคอมพิวเตอร์เครื่องอื่นๆ ข้อความเหล่านี้มักจะประกอบด้วยส่วนต่างๆ ของข้อความตัวอักษรที่ผู้ใช้ต้องการส่งไปยังคอมพิวเตอร์เครื่องอื่น แม้ในขณะนั้นผู้รับยังไม่ได้ล็อกออน ก็ยังสามารถอ่านข้อความนั้นได้ในภายหลัง วิธีการรูปแบบนี้คล้ายคลึงกับการรับส่งจดหมายนั่นเอง

เดิมที ข้อความในจดหมายอิเล็กทรอนิกส์ถูกจำกัดอยู่ในรูปแบบของข้อความตัวอักษรธรรมดาเท่านั้น แต่ในปัจจุบันได้มีระบบรองรับข้อความที่มีรูปแบบซับซ้อนยิ่งขึ้นกว่าเดิม เช่น เอกสารที่มีทั้งรูปภาพและข้อความประกอบกัน

เมื่อจดหมายอิเล็กทรอนิกส์ถูกส่งมาถึงระบบคอมพิวเตอร์ มันจะถูกเก็บไว้ในกล่องจดหมายอิเล็กทรอนิกส์ (electronics mailbox) เพื่อให้ผู้รับอ่านได้ในภายหลัง กล่องจดหมายอิเล็กทรอนิกส์คือไฟล์พิเศษบนคอมพิวเตอร์ที่ผู้ใช้สามารถเข้าถึงได้โดยผ่านคำสั่งหลายคำสั่ง ผู้ใช้แต่ละคนจะมีกล่องจดหมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อิเล็กทรอนิกส์เป็นของตัวเอง

จดหมายอิเล็กทรอนิกส์สามารถถูกส่งระหว่างผู้ใช้ได้โดยตรง แม้ว่าเครื่องคอมพิวเตอร์ของผู้ใช้ทั้งคู่จะอยู่ในระบบคอมพิวเตอร์คนละระบบก็ตาม โดยมีข้อแม้ว่าระบบทั้งคู่จะต้องต่อเข้ากับเครือข่ายใหญ่ ในปัจจุบันหน่วยงานส่วนใหญ่เกือบทั่วโลกมีจดหมายอิเล็กทรอนิกส์เป็นของตัวเอง และจำนวนของผู้ที่มีจดหมายอิเล็กทรอนิกส์ยังเพิ่มขึ้นทุกวัน

2.6 ความรู้เบื้องต้นของโปรโตคอล SMTP (Simple Mail Transfer Protocol)

2.6.1 วิธีการทำงานของโปรโตคอล SMTP

วิธีการนี้เป็นวิธีที่นิยมใช้กันบนระบบยูนิกซ์ (unix) ซึ่งเป็นโปรโตคอลที่อาศัยวิธีการส่งจดหมายเป็นทอดๆระหว่างโฮสต์ต่อๆกันจนกว่าจะไปถึงโฮสปลายทาง ในวิธีการนี้ ถ้าไม่เป็นเครื่องคอมพิวเตอร์รับเอาไว้ตลอดเวลาก็จะไม่สามารถรับจดหมายได้ และในปัจจุบันเครื่องคอมพิวเตอร์ส่วนบุคคลทั้งหลายก็ไม่ได้ใช้ระบบปฏิบัติการยูนิกซ์ และระบบปฏิบัติการที่ใช้ก็ไม่รองรับไฟล์ในระบบยูนิกซ์นั่นก็หมายความว่า หากใช้เครื่องคอมพิวเตอร์ส่วนบุคคล ถึงจะเปิดเครื่องไว้เครื่องนั้นก็ไม่สามารถใช้ไฟล์นั้นได้อยู่ดี ในปัจจุบันระบบนี้จึงเป็นระบบเก่าที่ไม่ได้ใช้ประโยชน์เท่าใดนัก

2.6.2 การรับส่งจดหมายอิเล็กทรอนิกส์ระหว่างเครื่องโดยใช้โปรโตคอล SMTP

ระบบอินเตอร์เน็ตจะใช้โปรโตคอล TCP/IP เป็นฐานการติดต่อระหว่างกัน โดยที่โปรโตคอล TCP/IP มีการกำหนดรูปแบบการรับส่งข้อมูลระหว่างกันในรูปแบบการรับส่งจดหมายอิเล็กทรอนิกส์ คือ SMTP (Simple Mail Transfer Protocol) โปรโตคอล SMTP ได้รับการออกแบบมาให้มีการรับส่งจดหมายกันอย่างอัตโนมัติ กล่าวคือ ในเครื่องหลักที่กำลังทำงานจะมีโปรแกรมรับและส่งจดหมายทำงานเป็นแบ็กกราวนด์โปรเซส (background process) คือทำงานตลอดเวลา เพื่อทำหน้าที่รับและส่งจดหมาย เช่น โปรแกรม smail หรือ sendmail ทำหน้าที่คอยส่งจดหมาย ถ้ามีจดหมายอิเล็กทรอนิกส์ที่จะส่งก็จะติดต่อกับปลายทาง ถ้ายังส่งไม่ได้ก็จะเก็บจดหมายนั้นไว้ก่อน รอเวลาอีกชั่วขณะหนึ่งก็จะส่งใหม่อีก ทำซ้ำจนกว่าจะส่งได้สำเร็จ หากส่งไม่ได้ภายในเวลาที่กำหนดก็จะส่งกลับให้เจ้าของ พร้อมทั้งบอกเหตุผลของการส่งจดหมายไม่สำเร็จ

โปรโตคอล SMTP นี้ เป็นโปรโตคอลหลักสำหรับการรับส่งจดหมายอิเล็กทรอนิกส์มาตรฐานบนเครือข่ายอินเตอร์เน็ต ดังนั้นหากในระบบเครือข่ายอื่น เช่นเครือข่ายของไมโครซอฟต์มีการส่งด้วย msmail โปรแกรม msmail มีโปรโตคอลการส่งของตัวเอง หากต้องการผ่านเข้ามาทางอินเตอร์เน็ตจะต้องแปลงให้อยู่ในรูปโปรโตคอล SMTP ก่อน เส้นทางการเปลี่ยนนี้เรียกว่า SMTP เกตเวย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 ความรู้เบื้องต้นของ MIME (Multipurpose Internet Mail Extensions)

2.7.1 วิธีการทำงานของ MIME

เนื่องจากจดหมายอิเล็กทรอนิกส์ในสมัยที่เพิ่งเริ่มมีใช้ในระบบอินเตอร์เน็ต จะเป็นแค่เครื่องมือที่ใช้ในการส่งข้อความตัวอักษรโดยที่ผู้ใช้ไม่สามารถแนบเอกสารหรือรูปภาพส่งเพิ่มเติมไปได้ จนกระทั่งได้มีการพัฒนาโปรโตคอลตัวใหม่ที่มีชื่อว่า MIME ซึ่งเป็นมาตรฐานในการเข้ารหัสแฟ้มข้อมูลหลายชนิดไปรวมกับจดหมายอิเล็กทรอนิกส์ผ่านอินเตอร์เน็ต ซึ่งในปัจจุบันนี้ไม่มีไฟล์ประเภทใดที่ MIME ไม่รู้จัก ผู้ใช้สามารถส่งไฟล์แทรกไปพร้อมกับจดหมายอิเล็กทรอนิกส์ได้ทุกประเภท โดยมีวิธีการคือ แปลงไฟล์รูปภาพ, เสียง, วีดีโอ ซึ่งอยู่ในรูปแบบไบนารี (Binary) ให้อยู่ในรูปแบบตัวอักษร MIME เป็นตัวมาตรฐานที่กำหนดขึ้น เพื่อรองรับจุดประสงค์ที่หลากหลายจากการใช้งานรับและส่งจดหมายอิเล็กทรอนิกส์ผ่านอินเตอร์เน็ต ทั้งนี้เพื่อขยายประโยชน์ใช้สอยของจดหมายอิเล็กทรอนิกส์ได้มากขึ้น แฟ้มข้อมูลมาตรฐาน MIME สามารถใช้ร่วมกับการเก็บไฟล์ข้อมูลเพื่อส่งผ่านไปทางโปรโตคอล SMTP และ UUCP (Unix-to-Unix Copy Protocol) รวมถึง BitNet, X.400, SNADS, PROFS และยังมีความสามารถในการแลกเปลี่ยนข้อมูลบนระบบปฏิบัติการที่ต่างกันรวมทั้งชนิดของซอฟแวร์ที่ใช้ต่างกันได้อย่างน่าอัศจรรย์

ตัวอย่าง E-mail ที่ใช้ MIME

```
X-Gmail-Received: 296a7defbff71f6600e7763b07a93ec758986b6d
Delivered-To: jsiripongse@gmail.com
Received: by 10.38.10.7 with SMTP id 7cs3907rmj;
        Thu, 10 Mar 2005 05:59:08 -0800 (PST)
Received: by 10.11.100.43 with SMTP id x43mr77868cwb;
        Thu, 10 Mar 2005 05:59:08 -0800 (PST)
Return-Path: <summonmarth@hotmail.com>
Received: from nuengpretty ([161.246.18.32])
        by mx.gmail.com with ESMTP id
v71si1643324cwb.2005.03.10.05.58.57;
        Thu, 10 Mar 2005 05:59:08 -0800 (PST)
Received-SPF: softfail (gmail.com: domain of transitioning
summonmarth@hotmail.com does not designate 161.246.18.32 as permitted
sender)
Received: from [127.0.0.1] by nuengpretty
  (ArGoSoft Mail Server Freeware, Version 1.8 (1.8.5.5)); Thu, 10 Mar
2005 20:58:05
MIME-Version: 1.0
Date: Thu, 10 Mar 2005 20:58:04 +0700
X-Mailer: Chilkat Software Inc (http://www.chilkatsoft.com)
X-Priority: 3 (Normal)
To: "Siripongse Jantree" <jsiripongse@gmail.com>
From: summonmarth@hotmail.com
return-path: summonmarth@hotmail.com
subject: Visitor Information
Content-Type: multipart/mixed;
        boundary="-----=_NextPart_e71_c533_68664b1b.b17655a5_.MIX"
Message-ID: <CHILKAT-MID-7ad940c1-2474-4ef5-b018-
6aa016129475@nuengpretty>

This is a multi-part message in MIME format.

-------=_NextPart_e71_c533_68664b1b.b17655a5_.MIX
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Content-Type: text/plain
Content-Transfer-Encoding: quoted-printable

069776775
------=_NextPart_e71_c533_68664b1b.b17655a5_.MIX
Content-Type: image/jpeg;
        name="Nook.jpg"
Content-Transfer-Encoding: base64
Content-Disposition: attachment;
        filename="Nook.jpg"

/9j/4AAQSkZJRgABAQEAYABgAAD/2wBDAAIBAQEBAQIBAQECAgICAgQDAgICAgUEBAMEB
gUG
BgYFBgYGBwkIBgcJBwYGCAsICQoKCgoKBggLDAsKDAkKCgr/2wBDAQICAgICAgUDAwUKB
wYH
CgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgr/w
AAR
CAHgAoADASIAAhEBAxEB/8QAHwAAAQUBAQEBAQEAAAAAAAAAAAECAwQFBgcICQoL/8QAt
RAA
...

------=_NextPart_e71_c533_68664b1b.b17655a5_.MIX--
```

- **Content-Type** ระบุถึงข้อมูลที่อยู่ในส่วนของ Body ซึ่ง User Agent จะใช้ข้อมูลนี้ในการดำเนินการในการจัดเก็บข้อมูลนั้น
- **Content-Transfer-Encoding** ระบุชนิดของการแปลงรูปแบบที่ใช้ในการแสดงส่วนของ Body ซึ่งจะแสดงผลไปอย่างถูกต้อง และ การเปลี่ยนรูปแบบ mail เมื่อถูกส่งผ่าน mail gateway ไม่เกิดผลกระทบ
- **Content-Disposition** แสดงรายละเอียดการแสดงผลในส่วนของ Body

2.7.2 การเข้ารหัส MIME

การรับส่งจดหมายอิเล็กทรอนิกส์ผ่านเครือข่ายนั้น คอมพิวเตอร์ที่เชื่อมต่อกันอยู่ภายในเครือข่ายมักจะมีมากมายหลากหลายชนิด ดังนั้น ข้อมูลที่จะถูกส่งผ่านจึงต้องเป็นข้อมูลที่อยู่ในรูปแบบกลางๆซึ่งคอมพิวเตอร์จะรับรู้และเข้าใจได้เหมือนกัน เพื่อไม่ให้ข้อมูลที่รับหรือส่งเหล่านั้นเกิดผิดเพี้ยนไปจากความเป็นจริงและเพื่อให้สามารถส่งข้อมูลทั้งที่เป็นข้อความและไม่เป็นข้อความ (เช่น ข้อมูลที่เป็นรูปภาพและเสียง) รวมกันไปในจดหมายอิเล็กทรอนิกส์ฉบับเดียวกันได้ จึงได้นำเทคนิคการเข้ารหัสที่เรียกว่า MIME มาใช้เพื่อเข้ารหัสและถอดรหัสในการรับส่งจดหมายอิเล็กทรอนิกส์โดยทั่วไป

เทคนิคของ MIME นี้เป็นเทคนิคที่แปลงรหัสแอสกี้ (ASCII) ทั่วไปซึ่งมี 8 บิตให้เป็นค่า 7 บิต (ให้บิตที่ 0 มีค่าเป็น 0 เสมอ) โดยที่เทคนิคของ MIME นี้จะสามารถใช้รับส่งข้อมูลได้ทุกๆรูปแบบ ไม่ว่าจะเป็นข้อมูลของจดหมายอิเล็กทรอนิกส์หรือไฟล์ประเภทต่างๆที่แนบไปกับจดหมายอิเล็กทรอนิกส์ ซึ่งการรับส่งจดหมายอิเล็กทรอนิกส์ผ่านอินเตอร์เน็ตในยุคแรกๆ ในกรณีที่ต้องการรับส่งข้อมูลที่มีรูปแบบไฟล์แตกต่างไปจากค่าแอสกี้ โดยทั่วไปผู้ส่งจะต้องแปลงรหัสข้อมูลก่อนส่งด้วยคำสั่ง UUENCODE เพื่อแปลงข้อมูลให้อยู่ในรูปแบบของ MIME ในด้านผู้รับก็จะต้องถอดรหัสข้อมูลกลับมาอยู่ในรูปแบบเดิมโดยใช้คำสั่ง UUDECOCE ซึ่งทั้ง 2 คำสั่งนี้เริ่มพัฒนาขึ้นมาพร้อมกับระบบปฏิบัติการยูนิกซ์ ภายหลังจึงมีการใช้กันอย่างแพร่หลายในระบบปฏิบัติการอื่นๆ ปัจจุบัน โปรแกรมที่ทำหน้าที่รับส่งจดหมายอิเล็กทรอนิกส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะทำหน้าที่เข้ารหัสและถอดรหัสข้อมูลให้โดยอัตโนมัติ โดยไม่จำเป็นต้องอาศัยคำสั่ง UUENCODE และ UUDECODE อีกต่อไป

ลักษณะข้อมูลของ MIME ประกอบด้วย 2 ส่วน คือ ส่วนหัว หรือเรียกว่า Content Transfer Encode ซึ่งจะเก็บรายละเอียดของไฟล์ที่เข้ารหัสไว้ เช่น ประเภทของไฟล์เป็นต้น ส่วนที่ 2 เป็นส่วนของข้อมูลที่เข้ารหัสแล้ว การเข้ารหัสและการถอดรหัสของ MIME นี้จะถูกระบุ ไว้ในส่วนหัว เพื่อให้ผู้รับและผู้ส่งเข้าใจตรงกันว่า อีเมล์นี้เข้ารหัสและถอดรหัสด้วยวิธีใด ซึ่งมีอยู่ด้วยกัน 6 วิธี คือ

- **วิธีเข้ารหัสแบบ Quoted-Printable**

เทคนิคการเข้ารหัสวิธีนี้จะแปลงข้อมูลให้อยู่ในลักษณะที่อ่านได้เสมอ ซึ่งหากข้อมูลเป็นแอสกี้ 7 บิตอยู่แล้วก็จะไม่มีการแปลงข้อมูล แต่ถ้าบิตที่ตำแหน่งศูนย์มีค่าเป็น "1" ข้อมูลจะถูกแปลงให้มาอยู่ในรูปของเลขฐาน 16 และนำหน้าด้วยเครื่องหมายเท่ากับ (=) ตัวอย่างเช่น ข้อมูลที่เข้ารหัสแล้วมีค่าเป็น =A1 หมายถึงข้อมูลที่ค่ารหัสแอสกี้เป็น 161 (ในภาษาไทยคือ ค่า 'ก') หรือค่า Hex เป็น A1 เป็นต้น

- วิธีเข้ารหัสแบบ **Base64**

เป็นเทคนิคการเข้ารหัสโดยจะแปลงข้อมูลจำนวน 24 บิต (ข้อมูล 8 บิตจำนวน 3 ไบต์) ออกเป็นข้อมูล 6 บิตจำนวน 4 ชุด โดยหลังจากที่เข้ารหัสแล้ว ข้อมูลจะถูกแปลงให้อยู่ในรูปของตัวอักษร 64 ตัว มีค่าตามตาราง Base64 Alphabet แต่ข้อมูลดังกล่าวจะไม่เปลี่ยนแปลงค่าของ Carriage Return และ Line Feed และปิดท้ายข้อมูลด้วยเครื่องหมาย = ซึ่งเรียกว่า PAD Binary เป็นข้อมูลที่ต่อเนื่องกันเป็นค่าไบนารี ไม่แบ่งออกเป็นบรรทัด ซึ่งข้อมูลประเภทนี้จะส่งโดยไม่มีการเข้ารหัสข้อมูล

- วิธีเข้ารหัสแบบ **Seven-Bit**

เป็นข้อมูลที่มีค่า ASCII 7 บิต ซึ่งข้อมูลประเภทนี้จะส่งโดยไม่มีการเข้ารหัสข้อมูล

- วิธีเข้ารหัสแบบ **Eight-Bit**

เป็นข้อมูลที่มีค่า ASCII 8 บิต ซึ่งข้อมูลประเภทนี้จะส่งโดยไม่มีการเข้ารหัสข้อมูล

- **วิธีเข้ารหัสแบบ X-Token**

เป็นเทคนิคการเข้ารหัสที่ต้องมีการติดต่อและตกลงกันระหว่างด้านผู้ส่งและด้านผู้รับของ SMTP Server ก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 MMS (Multimedia Messaging Services)

MMS (Multimedia Messaging Service) เป็นบริการรับ-ส่งข้อความผ่านทางโทรศัพท์เคลื่อนที่ที่มีโครงสร้างพื้นฐานคล้ายกับการส่ง SMS (Short Messaging Service) และ EMS (Enhanced Messaging Service) เพียงแต่เพิ่มประสิทธิภาพของการส่งให้สอดคล้องกับเครือข่ายอินเทอร์เน็ต รวมทั้งรูปแบบของข้อความที่ต้องการส่งที่ไม่ได้มีแค่ตัวอักษรเท่านั้น หากยังครอบคลุมไปถึงรูปภาพและเสียงด้วย ทำให้ข้อความที่ได้รับผ่านมือถือมีสีสันและดูสนุกมากขึ้นนั่นเอง

MMS เป็นมาตรฐานการรับส่งข้อมูลที่ได้รับการออกแบบขึ้นสำหรับการใช้งานภายในเครือข่ายโทรศัพท์เคลื่อนที่ยุคที่ 3 (3G) โดยเป็นผลงานของหน่วยงาน 3GPP (3rd Generation Partnership Program)

อย่างไรก็ตาม ไม่ได้หมายความว่า MMS จะเป็นเทคโนโลยีสำหรับอนาคตที่ไม่สามารถนำมาใช้กับการสื่อสารผ่านเครือข่ายโทรศัพท์เคลื่อนที่ในปัจจุบัน อันที่จริงแล้วขอบเขตการให้บริการของ MMS นั้นกว้างมาก โดยสามารถเริ่มให้บริการได้กับเครือข่ายโทรศัพท์เคลื่อนที่ ตั้งแต่ยุค 2.5G อันได้แก่เครือข่าย GPRS, CDMA2000 หรือแม้กระทั่ง i-Mode ซึ่งต่างก็มีความสามารถในการให้บริการสื่อสารข้อมูลผ่านเครือข่ายในรูปแบบแพ็กเกตทั้งสิ้น (หมายความว่าผู้ใช้บริการสามารถสื่อสารข้อมูลกับเครือข่ายได้ทุกเมื่อโดยไม่ต้องขอเชื่อมต่อวงจรไว้ก่อน) คุณภาพและอัตราเร็วในการสื่อสารข้อมูลโดยใช้เทคโนโลยี MMS บนเครือข่าย 3G นั้นจะเหนือกว่าการใช้งานบนเครือข่าย 2.5G เนื่องจากอัตราเร็วในการสื่อสารข้อมูลที่เพิ่มสูงขึ้นของเครือข่าย 3G นั่นเอง

2.8.1 รูปแบบการให้บริการ MMS

บริการที่ MMS มีให้กับผู้ใช้บริการนั้นนับว่าเหนือกว่าเทคโนโลยี SMS และ EMS อย่างไม่อาจเทียบกันได้ เนื่องจาก SMS นั้นมีข้อจำกัดในการส่งผ่านข้อมูลในเรื่องของอัตราเร็วและขนาดของข้อมูลซึ่งถูกจำกัดโดยเครือข่ายโทรศัพท์เคลื่อนที่เอง สำหรับ EMS นั้นแม้จะมีลูกเล่นมากกว่า SMS แต่แท้จริงแล้ว ก็เป็นเพียงเทคนิคการกระจายส่งข้อมูลผ่านบนข้อความ SMS หลายๆ ชุด แล้วนำมาเรียงต่อกัน ทำให้ยังคงติดขัดอยู่กับข้อจำกัดของช่องสื่อสารแบบ SMS เช่นเดียวกัน ประกอบทั้งการคิดค่าบริการบนเทคโนโลยี EMS นั้นมีข้อจำกัดต่อทั้งเครือข่ายผู้ให้บริการและผู้ใช้บริการโทรศัพท์เคลื่อนที่เอง เนื่องจากในการใช้บริการ EMS แต่ละครั้งนั้น แท้จริงแล้วเป็นการส่งข้อความ SMS หลายๆ ชุด ตัวอย่างเช่น ในการดาวน์โหลดภาพหน้าจอ มายังเครื่องลูกข่ายหนึ่งครั้ง อาจต้องใช้ข้อความ SMS ถึง 7 ชุดเรียงต่อกัน ปัญหาจึงเกิดขึ้นว่า ผู้ให้บริการจะคิดค่าบริการในการดาวน์โหลดครั้งนี้เท่าไรจึงจะเหมาะสม เพราะหากคิดค่าส่งเท่ากับการส่งข้อความ SMS 7 ชุด ก็น่าจะเป็นราคาที่สูงเกินไปจนทำให้ไม่มีผู้ใดยอมใช้บริการ EMS อีก

สำหรับเทคโนโลยี MMS นั้น มีการใช้ช่องทางสื่อสารข้อมูลได้ทั้งบนเครือข่าย GPRS, วงจรสื่อสารแบบที่ต้องเชื่อมต่อล่วงหน้า (Circuit Switch) บนเครือข่าย GSM หรือแม้กระทั่งกับช่องทางสื่อสารแบบ SMS ซึ่งการเลือกใช้ช่องสื่อสารชนิดใดนั้น ก็ขึ้นอยู่กับเงื่อนไขที่ได้รับการกำหนดขึ้นโดยบริษัทผู้ให้บริการเอง ในที่นี้สามารถแบ่งประเภทของบริการบนเทคโนโลยี MMS ออกได้เป็น 4 ประเภทใหญ่ๆ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การรับส่งข้อมูลที่เป็นตัวอักษรเพียงอย่างเดียว ซึ่งเป็นได้ทั้งในรูปแบบ ASCII, Unicode หรือรูปแบบมาตรฐานสากลอื่นๆ
- การรับส่งข้อมูลที่เป็นรูปภาพสีและภาพนิ่งต่างๆทั้งที่อยู่ในรูปแบบมาตรฐาน JPEG, GIF หรือรูปแบบมาตรฐานสากลอื่นๆ
- การรับส่งข้อมูลเสียงต่างๆไม่ว่าจะอยู่ในรูปแบบการจัดเก็บแบบ MP3, MIDI, WAV, AMR หรือรูปแบบมาตรฐานสากล อื่นๆ
- การรับส่งสัญญาณวิดีโอหรือรายการภาพยนตร์ต่างๆ ที่อยู่ในรูปแบบการจัดเก็บแบบ MPEG4, H.263 หรือ Quick Time

ขอบเขตการรองรับข้อมูลในรูปแบบต่างๆ ที่หลากหลายของเทคโนโลยี MMS ประกอบกับความพร้อมของเครื่องโทรศัพท์เคลื่อนที่รุ่นใหม่ๆ ที่ได้รับการออกแบบมาเพื่อรองรับเทคโนโลยีดังกล่าว ไม่ว่าจะเป็นหน่วยความจำที่เพิ่มสูงขึ้น จอแสดงผลแบบสี มีระบบเสียงคุณภาพสูง ใช้หน่วยประมวลผลที่มีสมรรถนะสูงขึ้น ทั้งหมดนี้ล้วนเป็นปัจจัยสำคัญทางด้านเทคโนโลยีที่จะสร้างกระแสการสื่อสารข้อมูลผ่านโทรศัพท์เคลื่อนที่ได้อย่างที่ไม่เคยปรากฏมาก่อน ตัวอย่างแนวทางสำหรับการให้บริการที่น่าสนใจในทางปฏิบัติ มีดังนี้

- การเล่นเกมส์ หรือการเปิดให้บริการด้านความบันเทิงแบบ interactive ซึ่งผู้ใช้บริการ MMS จะมีการรับส่งข้อมูลกับเครือข่ายอยู่เสมอๆ

- การเล่นเกมส์ หรือการเปิดให้บริการด้านความบันเทิงแบบ interactive ซึ่งผู้ใช้บริการ MMS จะมีการรับส่งข้อมูลกับเครือข่ายอยู่เสมอๆ

- การให้บริการข้อมูลที่เกี่ยวข้องกับตำแหน่งที่อยู่ของผู้ใช้บริการ รวมไปถึงบริการจับคู่, หาเพื่อน ฯลฯ

- การเปิดให้ดาวน์โหลดเพลงแบบ MP3 โดยมีการคิดเงินทั้งแบบเก็บค่าสมาชิกรายเดือน หรือเก็บเงินตามจำนวนครั้งที่มีการดาวน์โหลด

- การรายงานข่าวสารแบบเจาะติดตามสถานการณ์ โดยเป็นการผลัก (Push) ข้อมูลข่าวสารต่างๆ ผ่านทางเทคโนโลยี MMS ไปให้กับผู้ใช้บริการ ไม่ว่าจะเป็นข่าวสารทั่วไป, ข่าวการเงิน, ข่าวกีฬา ฯลฯ

- การส่งรูปภาพพร้อมเสียงพูดระหว่างบุคคล เช่น การส่งการ์ดอวยพรที่มีทั้งภาพและเสียงเพลงหรือเสียงพูด ของผู้ใช้บริการรายหนึ่งไปให้กับเพื่อนหรือคนรัก โดยเครือข่ายมีรายการภาพให้เลือกหรืออาจเป็นการส่งโดยตรงระหว่างผู้ใช้บริการรายหนึ่งไปให้อีกรายหนึ่ง

- การทำธุรกรรมแบบ m-commerce โดยเพิ่มความน่าสนใจให้กับธุรกิจด้วยการเพิ่มเสียงและภาพวิดีโอ เช่น การขายหรือสั่งจองตั๋วภาพยนตร์ทางโทรศัพท์เคลื่อนที่ ซึ่งผู้ใช้บริการจะได้เห็นภาพและเสียงของหนังตัวอย่างต่างๆก่อนตัดสินใจกดปุ่มสั่งจองหรือซื้อตั๋ว

- การรับฝาก Voicemail แบบก้าวหน้า ซึ่งผู้ได้รับฝาก ไม่จำเป็นต้องใช้โทรศัพท์โทรเข้าไปฟังข้อความเหมือนในอดีต แต่เสียงพูดที่ถูกฝากไว้นั้นจะถูกแปลงเป็นข้อมูล WAV หรือรูปแบบอื่นๆ เพื่อส่งในรูปของข้อมูล MMS ไปยังเครื่องลูกข่าย ของผู้ถูกฝากโดยตรง

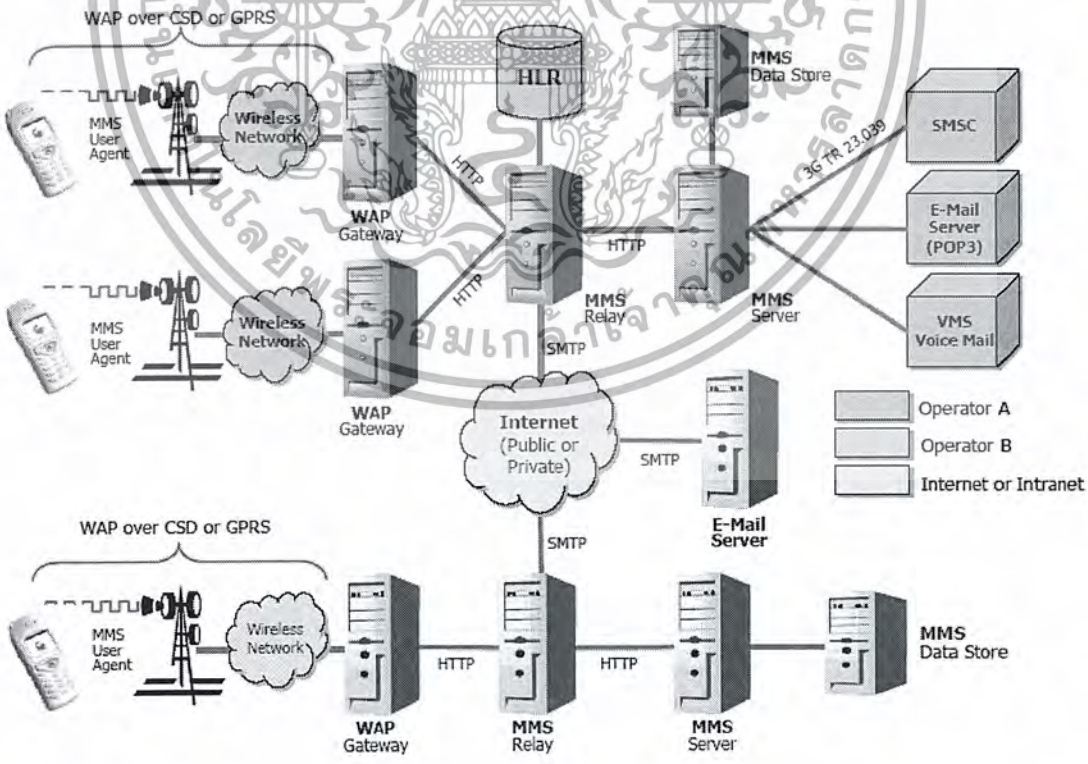
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8.2 โครงสร้างและการทำงานของเทคโนโลยี MMS

รูปที่ 2.17 แสดงส่วนประกอบสำคัญของการให้บริการ MMS ในเครือข่ายโทรศัพท์เคลื่อนที่ ซึ่งประกอบด้วยเครื่องลูกข่ายโทรศัพท์เคลื่อนที่ที่รองรับเทคโนโลยี MMS และอุปกรณ์ศูนย์ให้บริการข้อมูลแบบ MMS ซึ่งมีชื่อเรียกเป็นทางการว่า MMSC (Multimedia Messaging Service Center) การลงทุนติดตั้งอุปกรณ์ MMSC นั้น อาจกระทำโดยบริษัทผู้ให้บริการโทรศัพท์เคลื่อนที่เอง หรืออาจเป็นบริษัทอื่นๆที่ให้บริการแอปพลิเคชั่นแบบ MMS ก็ได้ อธิบายบทบาทหน้าที่ของส่วนประกอบที่สำคัญต่างๆ ได้ดังนี้

- **อุปกรณ์เครื่องลูกข่าย MMS** ครอบคลุมทั้งเครื่องลูกข่ายโทรศัพท์เคลื่อนที่ GSM, GPRS, CDMA และ 3G ที่มีรูปร่างหน้าตาเป็นโทรศัพท์พกพา และซอฟต์แวร์ที่ทำหน้าที่ติดต่อสื่อสารกับอุปกรณ์ MMSC ที่เขียนขึ้นมาเพื่อใช้ทำงานบนอุปกรณ์คอมพิวเตอร์แบบ Pocket PC

- **เครือข่ายโทรศัพท์เคลื่อนที่** สามารถเป็นได้ทั้งเครือข่ายแบบ GSM, GPRS, CDMA และ 3G หรืออาจเป็นเครือข่ายเฉพาะอื่นๆ เช่น i-mode ซึ่งเครือข่ายเหล่านี้จะต้องสามารถรองรับการสื่อสารข้อมูลที่มิใช่เสียงพูดระหว่างเครื่องลูกข่าย MMS กับอุปกรณ์ MMSC ได้อย่างน้อยหนึ่งประเภท ไม่ว่าจะเป็นการเชื่อมต่อตายตัวจนกว่าการสื่อสารจะจบลง การรับส่งข้อมูลแบบแพ็กเกต หรือเป็นเพียงช่องทางการสื่อสารแบบ SMS



รูปที่ 2.17 ส่วนประกอบของการให้บริการแบบ MMS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **เกตเวย์ WAP** ทำหน้าที่เป็นตัวกลางในการเชื่อมต่อระหว่างอุปกรณ์ MMSC กับเครือข่ายโทรศัพท์เคลื่อนที่ ซึ่งผู้ให้บริการอาจเลือกติดตั้งเข้าไปในเครือข่ายหรือไม่ก็ได้ หากติดตั้งอุปกรณ์เกตเวย์ WAP เข้าไปในเครือข่ายก็จะทำให้สามารถกำหนดลูกเล่นในการให้บริการ การคิดค่าบริการในรูปแบบต่างๆ และการรักษาความปลอดภัยของข้อมูลระหว่างเครื่องลูกข่ายกับอุปกรณ์ MMSC หรืออุปกรณ์คอมพิวเตอร์ภายนอกเครือข่ายที่ให้บริการแอปพลิเคชั่นอื่นๆ เช่น การหักเงินจากบัตรเครดิตเมื่อดาวน์โหลดแอปพลิเคชั่นบางอย่างได้อย่างมีความมั่นใจ การติดตั้งอุปกรณ์เกตเวย์ WAP เป็นตัวกลางในการเชื่อมต่อระหว่างอุปกรณ์ MMSC และเครือข่ายโทรศัพท์เคลื่อนที่ สามารถกระทำได้อย่างอิสระ โดยไม่จำเป็นต้องกังวลว่าเครือข่ายโทรศัพท์เคลื่อนที่เป็นมาตรฐานแบบใด เทคโนโลยี WAP

- **อุปกรณ์ MMSC** ทำหน้าที่เป็นตัวกลางในการรับและส่งข้อมูล MMS รวมถึงเก็บรวบรวมข้อมูลต่างๆทั้งที่เป็นฐานข้อมูลของผู้ใช้บริการโทรศัพท์เคลื่อนที่ที่เปิดใช้บริการ MMS และข้อมูล MMS ที่ยังไม่ถูกส่งไปถึงปลายทาง ไม่ว่าจะด้วยเหตุประการใดก็ตาม ภายในอุปกรณ์ MMSC โดยทั่วไปนั้นประกอบด้วยส่วนประกอบย่อยๆ ที่สำคัญ ดังนี้

  - **MMS Relay** เป็นหัวใจสำคัญของการสื่อสารข้อมูลแบบ MMS โดยมีหน้าที่หลักในการรับข้อมูล MMS ต่างๆทั้งจากเครื่องลูกข่าย MMS หรือจากแอปพลิเคชั่นภายนอก พร้อมกับหาทางส่งข้อมูลเหล่านั้นไปยังปลายทางซึ่งอาจเป็นทั้งเครื่องลูกข่าย MMS ด้วยกันหรืออาจเป็นแอปพลิเคชั่นภายนอก เช่น เว็บไซต์ อีเมล์ ฯลฯ หรืออาจเป็นการรับส่งข้อมูลระหว่างอุปกรณ์ MMS ด้วยกันเอง ในกรณีที่ผู้ใช้บริการโทรศัพท์เคลื่อนที่ต่างเครือข่ายกันส่งข้อมูล MMS ถึงกัน

  - **MMS Server และฐานข้อมูล MMS** รับผิดชอบเก็บพักข้อมูล MMS ต่างๆ ไม่ว่าจะเป็นรูปภาพ เสียง ข้อความ หรือวิดีโอที่มีการส่งหากันระหว่างผู้ใช้บริการ MMS ด้วยกัน หรือระหว่างแอปพลิเคชั่นภายนอกกับผู้ใช้บริการ MMS ทั้งนี้เนื่องจากเทคโนโลยี MMS ได้รับการออกแบบให้อยู่บนพื้นฐานของการสื่อสารแบบเก็บและส่งต่อ (Store and Forward) ในลักษณะเดียวกับการส่งข้อความแบบ SMS หรือ EMS ซึ่งจะช่วยให้เกิดความมั่นใจว่าข้อมูล MMS ต่างๆจะได้รับการเก็บพักไว้อย่างดีจนกว่าจะถูกส่งต่อไปให้กับเครื่องลูกข่าย MMS หรือแอปพลิเคชั่นปลายทาง ตัวอย่างการใช้งานเช่น เมื่อผู้ใช้บริการปิดเครื่องลูกข่าย MMS ของตน ซึ่งอุปกรณ์ MMS Server จะเก็บพักข้อมูล MMS ที่มีบุคคลอื่นส่งให้ไว้จนกว่าผู้ใช้บริการรายนั้นเปิดเครื่องอีกครั้ง

  - **ฐานข้อมูลผู้ใช้บริการ MMS** ใช้เก็บข้อมูลรายการผู้ใช้บริการโทรศัพท์เคลื่อนที่ที่เปิดใช้บริการ MMS พร้อมกับข้อมูลอื่นๆ ที่จำเป็นเกี่ยวกับการใช้บริการ MMS เมื่อมีผู้ใช้บริการโทรศัพท์เคลื่อนที่รายใดรายหนึ่ง ขอเปิดใช้บริการ MMS ก็จะถูกบันทึกข้อมูลที่สำคัญ เช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลขหมายโทรศัพท์และรายละเอียดการใช้งานอื่นๆ เช่น หมายเลขอีเมล์ ที่ต้องการให้อุปกรณ์ MMSC ส่งต่อข้อมูลไปให้แทน ใช้สำหรับผู้ใช้บริการที่ไม่มีเครื่องลูกข่ายแบบ MMS หรือเป็นผู้ใช้บริการที่มีเครื่องลูกข่าย MMS ในกรณีที่เครื่องลูกข่ายเกิดปัญหาขัดข้องจนใช้งานไม่ได้ และต้องการรับ หรือส่งข้อมูล MMS จากอีเมล์หรือเว็บไซด์แทน

การเปิดให้บริการ MMS นั้นสามารถทำได้ง่ายเนื่องจากความเป็นมาตรฐานของอุปกรณ์ MMSC ที่สามารถเชื่อมต่อกับเครือข่ายคอมพิวเตอร์ภายนอกได้ โดยใช้มาตรฐานการสื่อสารพื้นฐานที่มีการนิยมใช้งานทั่วไปในปัจจุบัน ไม่ว่าจะเป็น MIME ใช้สำหรับกำหนดรูปแบบของข้อมูล MMS ทั่วไป, SMIL (Synchronized Multimedia Integration Language) กำหนดวิธีการในการแสดงรูปแบบข้อมูล MMS ต่อผู้ใช้บริการ, WAP (Wireless Application Protocol) ใช้ในการติดต่อสื่อสารระหว่างเครื่องลูกข่าย MMS กับอุปกรณ์ MMSC หรือแม้กระทั่ง SMTP สำหรับใช้ในการเชื่อมต่อเพื่อการแลกเปลี่ยนข้อมูล MMS ระหว่างเครือข่ายโทรศัพท์เคลื่อนที่ด้วยกันหรือระหว่างเครือข่ายโทรศัพท์เคลื่อนที่กับผู้ให้บริการภายนอก

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

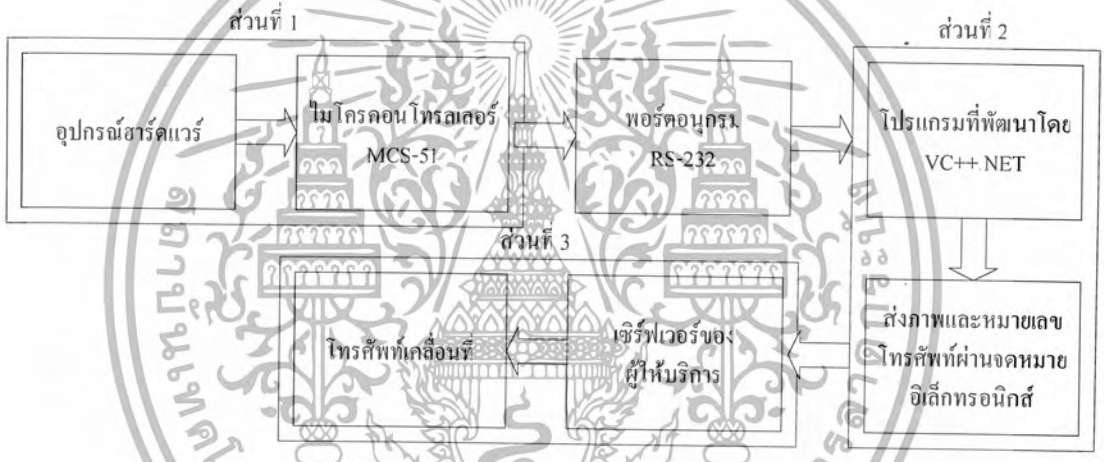
บทที่ 3
การคำนวณและการสร้าง

3.1 บล็อกไดอะแกรมการทำงานของระบบต้อนรับผู้มาเยือน

โครงสร้างโดยรวมของโครงงานนี้จะแบ่งออกเป็น 3 ส่วนคือ

1. ส่วนที่ควบคุมโดยไมโครคอนโทรลเลอร์ตระกูล MCS-51
2. ส่วนที่ควบคุมโดยโปรแกรมที่ถูกเขียนโดยใช้ภาษา Visual C++.NET
3. ส่วนส่งข้อมูลไปยังโทรศัพท์เคลื่อนที่

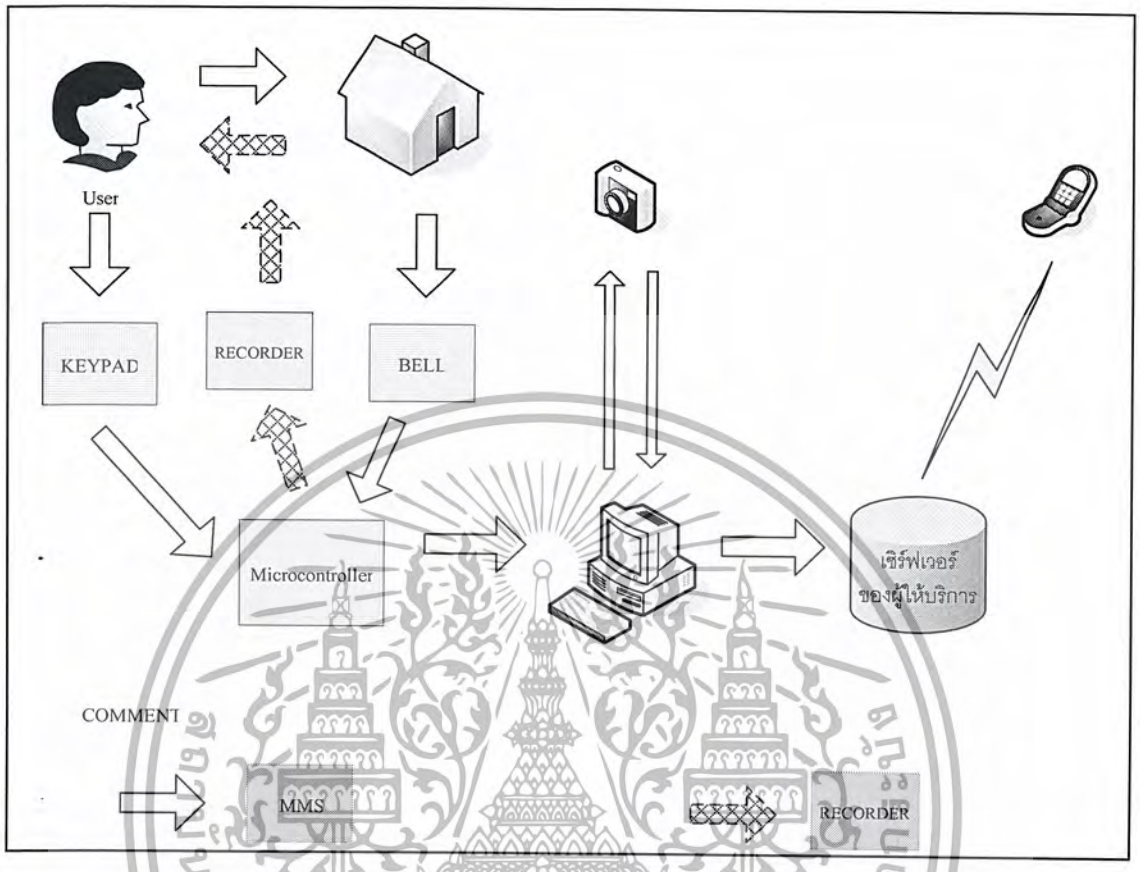
โดยส่วนที่ 1 จะติดต่อสื่อสารกับส่วนที่ 2 ผ่านทางพอร์ตอนุกรม และส่วนที่ 2 ติดต่อสื่อสารกับส่วนที่ 3 ผ่านทางสายดาต้าลิงค์ (data link) บล็อกไดอะแกรม (Block Diagram) รวมของโครงงานสามารถแสดงได้ดังรูปที่ 3.1



รูปที่ 3.1 บล็อกไดอะแกรมรวมของโครงงานระบบต้อนรับผู้มาเยือน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การทำงานของระบบต้อนรับผู้มาเยือน



รูปที่ 3.2 การทำงานของระบบต้อนรับผู้มาเยือน

จากรูปที่ 3.1 และรูปที่ 3.2 สามารถอธิบายการทำงานของส่วนต่างๆได้ดังนี้

3.2.1 การทำงานในส่วนที่ 1 อุปกรณ์ฮาร์ดแวร์และไมโครคอนโทรลเลอร์ MCS-51

การทำงานจะเริ่มต้นเมื่อมีการกดกริ่ง หลังจากนั้นจะมีพัลส์ไปกระตุ้นให้ไมโครคอนโทรลเลอร์ทำงาน ซึ่งกระบวนการทำงานของไมโครคอนโทรลเลอร์คือเรียกสัญญาณเสียงจากวงจรบันทึกเสียงซึ่งถูกโปรแกรมไว้แล้วให้ทำงานโต้ตอบกับผู้ใช้ (ผู้มาเยือน) หลังจากนั้นไมโครคอนโทรลเลอร์จะรับข้อมูลหมายเลขโทรศัพท์จากคีย์แพด แล้วส่งสัญญาณผ่านพอร์ตอนุกรมไปยังโปรแกรมVC++.NET เพื่อให้โปรแกรมVC++.NET เริ่มทำงานและส่งสัญญาณไปอีกครั้งเพื่อจบการทำงานของโปรแกรม VC++.NET ซึ่งอุปกรณ์ฮาร์ดแวร์ในส่วนที่ 1 นี้จะประกอบไปด้วยสัญญาณจากกริ่ง, คีย์แพด, วงจรบันทึกเสียง และไมโครคอนโทรลเลอร์ MCS-51ในส่วนของวงจรบันทึกเสียงนั้นแสดงดังรูปที่ 3.3

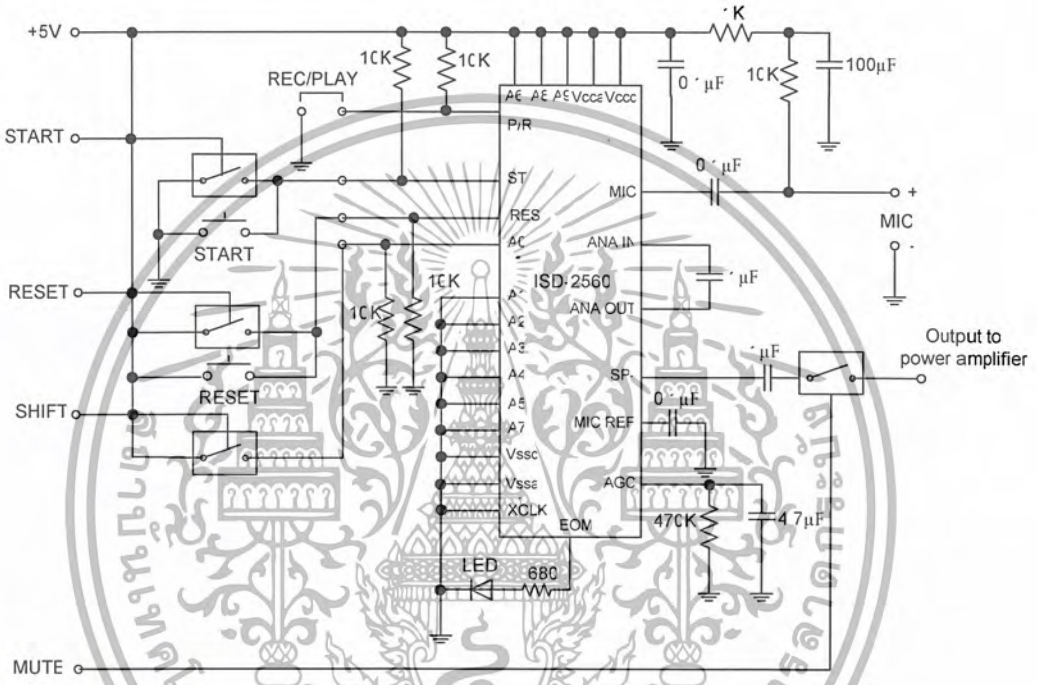
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรบันทึกเสียงและวงจรขยายกำลัง

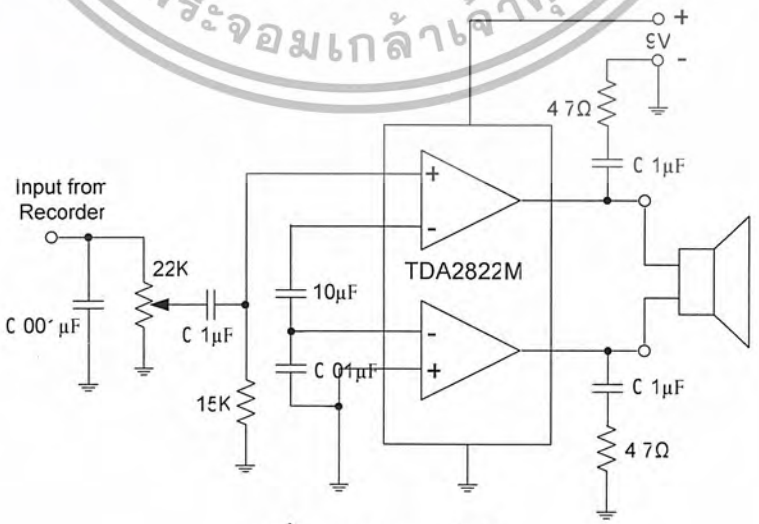
ในส่วนของวงจรบันทึกเสียงนั้นจะประกอบไปด้วยไอซี 2 ตัว คือ

1. ไอซี ISD-2560 ซึ่งใช้ในการบันทึกเสียง ดังคุณสมบัติที่ได้กล่าวไว้แล้วในบทที่ 2

2. ไอซี CD4066 ซึ่งภายในไอซีตัวนี้จะมีสวิตช์อยู่ 4 ตัว โดยในที่นี้เราจะใช้เป็นสวิตช์สำหรับการ START, SHIFT, RESET และ MUTE โดยการควบคุมการทำงานผ่านไมโครคอนโทรลเลอร์ MCS-51 และสวิตช์แต่ละตัวจะประกอบด้วยขา 3 ขา คือ ขาคอนโทรล, ขาอินพุตและขาเอาต์พุต ซึ่งการทำงานของสวิตช์แต่ละตัว คือ ถ้าป้อนไฟเข้าที่ขาคอนโทรลก็จะทำให้มีสัญญาณออกทางเอาต์พุต แต่ถ้าไม่ได้ป้อนไฟให้ขาคอนโทรลก็จะไม่มีสัญญาณออกทางเอาต์พุต



รูปที่ 3.3 วงจรบันทึกเสียง

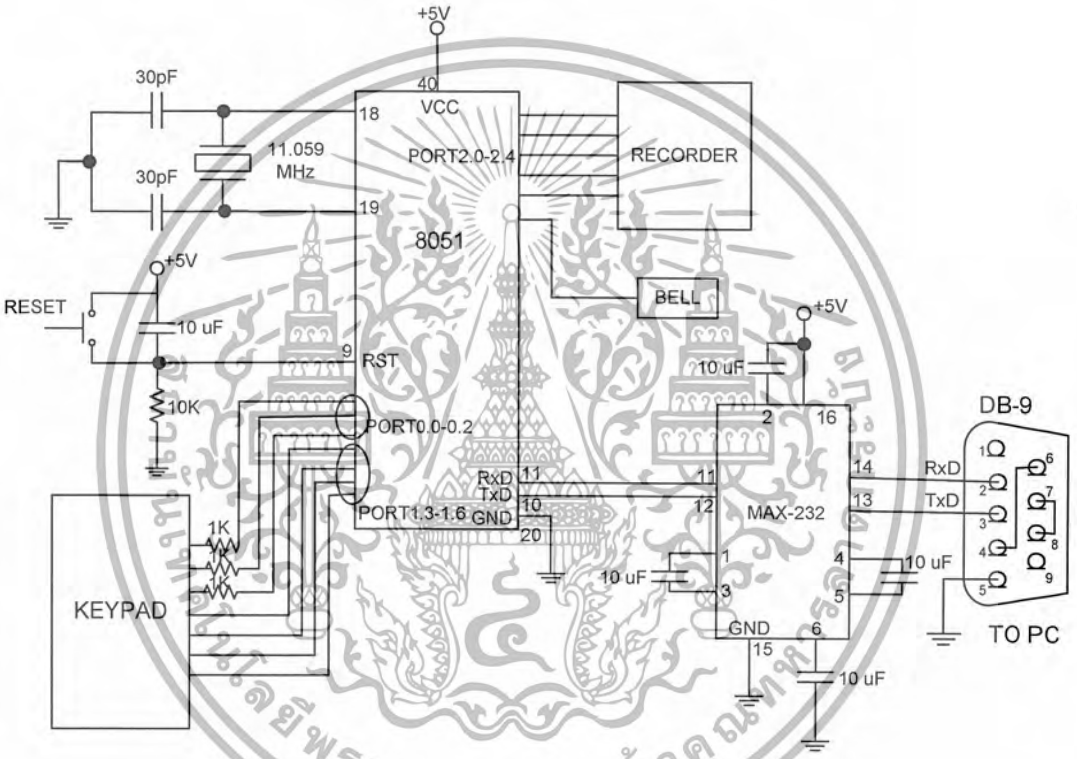


รูปที่ 3.4 วงจรขยายกำลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรเชื่อมต่อ MAX-232 เข้ากับพอร์ตอนุกรมของคอมพิวเตอร์และไมโครคอนโทรลเลอร์ MCS-51

ในการใช้งานไมโครคอนโทรลเลอร์ MCS-51 ติดต่อเพื่อแลกเปลี่ยนข้อมูลกับคอมพิวเตอร์ผ่านทางพอร์ตอนุกรมในมาตรฐาน RS-232 จะต้องอาศัยการเชื่อมต่อผ่านไอซีพิเศษที่ทำหน้าที่ในการแปลงระดับสัญญาณ สำหรับโครงงานนี้ได้เลือกใช้ไอซีหมายเลข MAX232 ซึ่งใช้ในการแปลงข้อมูลส่งของไมโครคอนโทรลเลอร์ MCS-51 จากระดับทีทีแอลไปเป็นระดับของ RS-232 และทำการแปลงข้อมูลรับจากคอมพิวเตอร์จากระดับของ RS-232 ไปเป็นระดับของทีทีแอลเพื่อให้สามารถทำการถ่ายทอดต่อไปยังไมโครคอนโทรลเลอร์ได้อย่างสมบูรณ์ โดยในรูปที่ 3.5 แสดงวงจรในการต่อกับไมโครคอนโทรลเลอร์ MCS-51



รูปที่ 3.5 วงจรรวมของการเชื่อมต่อของอุปกรณ์ต่างๆกับไมโครคอนโทรลเลอร์
และไมโครคอนโทรลเลอร์กับพอร์ตอนุกรม

3.2.2 การทำงานในส่วนที่ 2 โปรแกรมที่พัฒนาโดยใช้ VC++.NET และฐานข้อมูล

การทำงานของโปรแกรมที่พัฒนาโดยใช้ VC++.NET จะรับข้อมูลอนุกรม (หมายเลขโทรศัพท์) เพื่อเริ่มการทำงานของโปรแกรมจากพอร์ตอนุกรมที่ผ่านมาทาง RS-232 แล้วควบคุมให้กล้องโทรทัศน์วงจรปิดจับภาพในขณะนั้น และใส่ชื่อของรูปภาพที่บันทึกได้เป็นวันและเวลาที่กล้องจับภาพแล้วนำข้อมูลทั้งหมด (หมายเลขโทรศัพท์, ชื่อภาพ, ภาพ) ไปเก็บไว้ที่ฐานข้อมูล โดยโปรแกรมจะจบการทำงานในส่วนของการบันทึกข้อมูลเมื่อได้รับสัญญาณให้หยุดการทำงาน (หมายเลข 1) จากคีย์แพด หลังจากนั้นโปรแกรมจะทำงานในส่วนของการส่งข้อมูลภาพ และหมายเลขโทรศัพท์ในรูปแบบของการส่งจดหมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อิเล็กทรอนิกส์ โดยจะส่งจดหมายอิเล็กทรอนิกส์ไปยัง AIS Server เพื่อให้ทำการส่งข้อมูล MMS ไปยังโทรศัพท์เคลื่อนที่ต่อไป

นอกจากนั้น โปรแกรมยังมีการเรียกดูข้อมูลจากฐานข้อมูลเพื่อสะดวกในการตรวจสอบและการจัดระบบของข้อมูลและง่ายต่อการส่งและรับข้อมูลไว้ด้วย

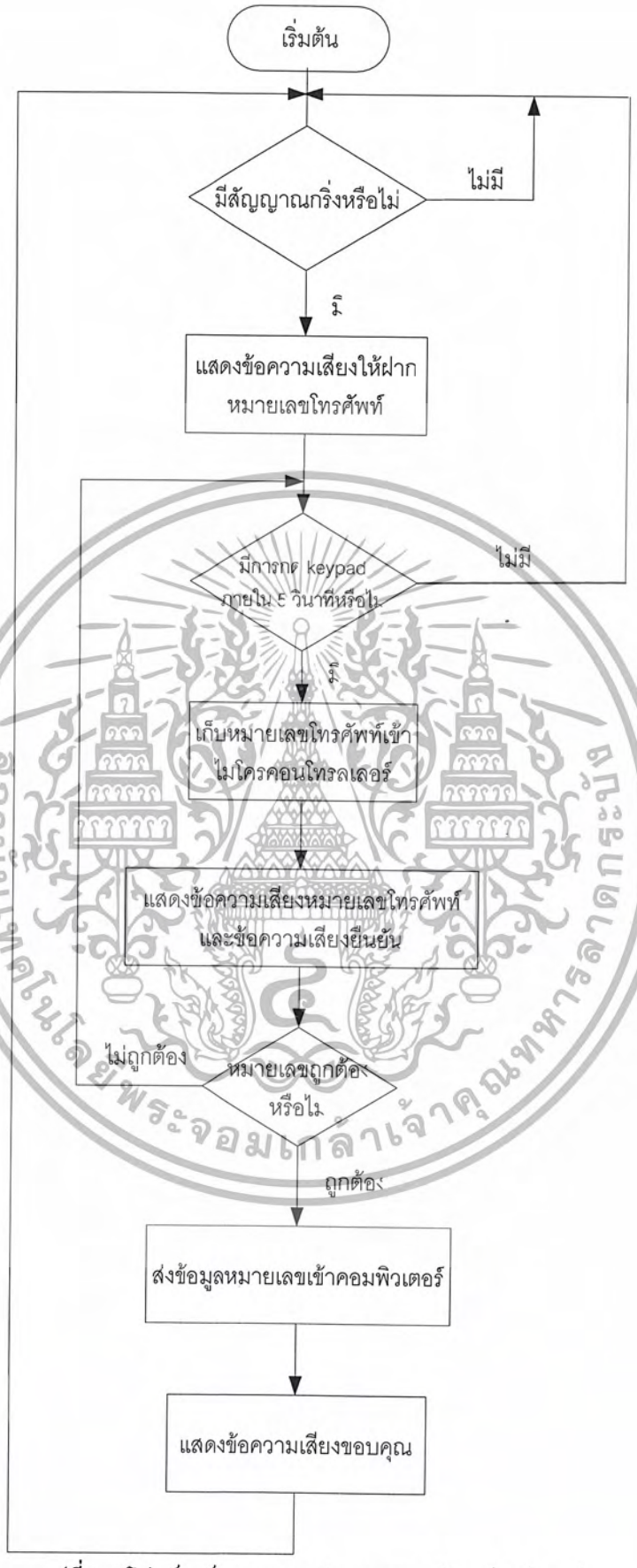
3.2.3 การทำงานในส่วนที่ 3 เซิร์ฟเวอร์ของผู้ให้บริการ

การทำงานในส่วนนี้ จะเป็นการรับข้อมูลจากจดหมายอิเล็กทรอนิกส์ที่ส่งไปโดยโปรแกรมที่ถูกพัฒนาโดยใช้ VC++.NET แล้วส่งข้อความมัลติมีเดียไปยังโทรศัพท์เคลื่อนที่ผ่านบริการของผู้ให้บริการเครือข่ายโทรศัพท์เคลื่อนที่

3.3 โฟลว์ชาร์ตการทำงานของไมโครคอนโทรลเลอร์

โฟลว์ชาร์ตการทำงานของไมโครคอนโทรลเลอร์สามารถแสดงได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 โฟลว์ชาร์ตแสดงการทำงานของอุปกรณ์ฮาร์ดแวร์
ซึ่งควบคุมโดยไมโครคอนโทรลเลอร์ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 โฟลว์ชาร์ตการทำงานของ VC++.NET

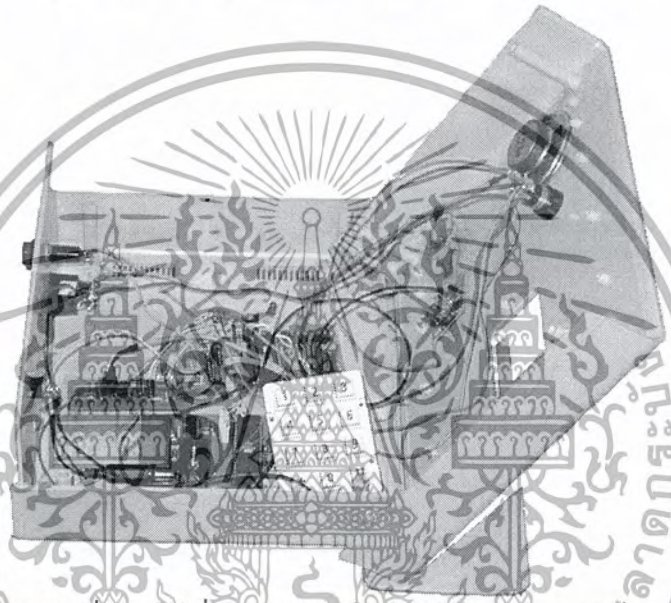
รูปที่ 3.7 โฟลว์ชาร์ตแสดงการทำงานในส่วนของการถ่ายภาพ
และการเก็บหมายเลขโทรศัพท์ลงในฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่4
ผลการทดลอง

4.1 ผลการทดลองส่วนของอุปกรณ์ฮาร์ดแวร์และไมโครคอนโทรลเลอร์

- วงจรบันทึกเสียงสามารถใช้งานร่วมกับคีย์แพดในการโต้ตอบกับผู้ใช้งาน และใช้ในการรับฝากหมายเลขโทรศัพท์ได้
- สามารถควบคุมการทำงานระหว่างวงจรบันทึกเสียง, คีย์แพด และวงจรกริ่ง ให้ทำงานสัมพันธ์กันได้
- สามารถส่งข้อมูลผ่านพอร์ตอนุกรมโดย RS-232 ได้



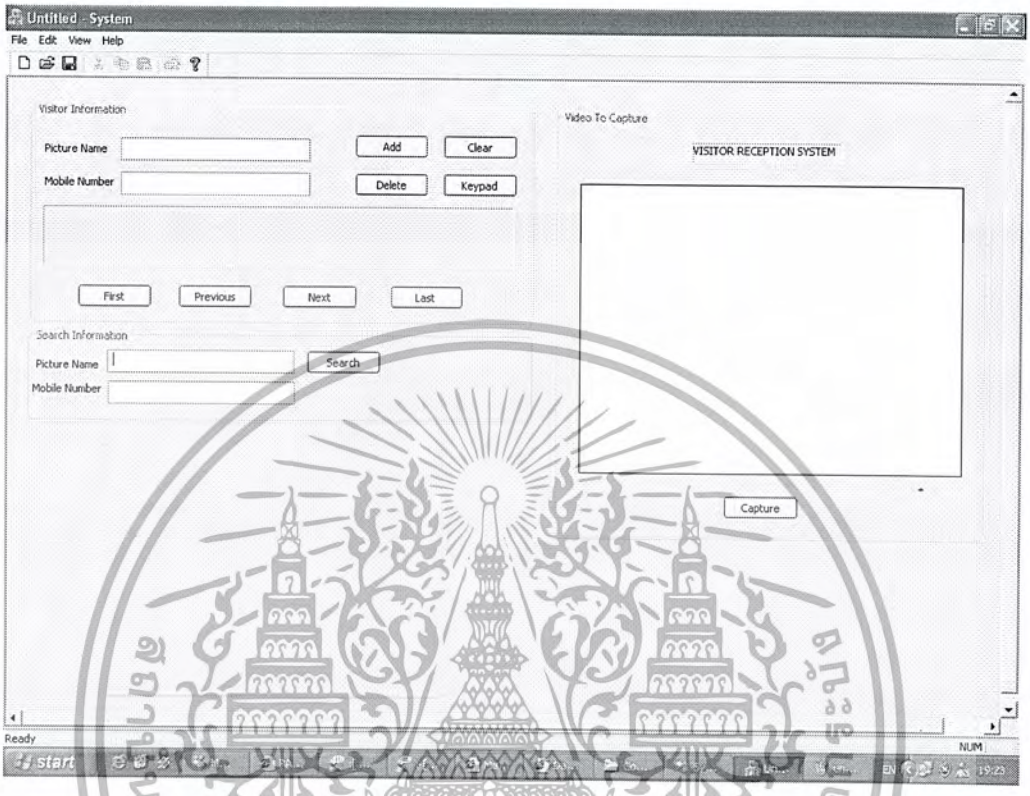
รูปที่ 4.1 การเชื่อมต่ออุปกรณ์กับไมโครคอนโทรลเลอร์

รูปที่ 4.2 อุปกรณ์พร้อมติดตั้งเพื่อใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ผลการทดลองในส่วนของ VC++.NET

- สามารถรับข้อมูลจากพอร์ตอนุกรมที่ผ่านมาทาง RS-232 ได้
- สามารถสร้างฐานข้อมูลโดยใช้คลาส CObList ร่วมกับคลาส CFromview ได้



รูปที่ 4.3 รูปแบบหน้าจอแสดงผลในการรับและแสดงข้อมูลของผู้มาเยือน

4.2.1 ผลการทดลองในส่วนของการแสดงผลที่ได้จากพอร์ตอนุกรม

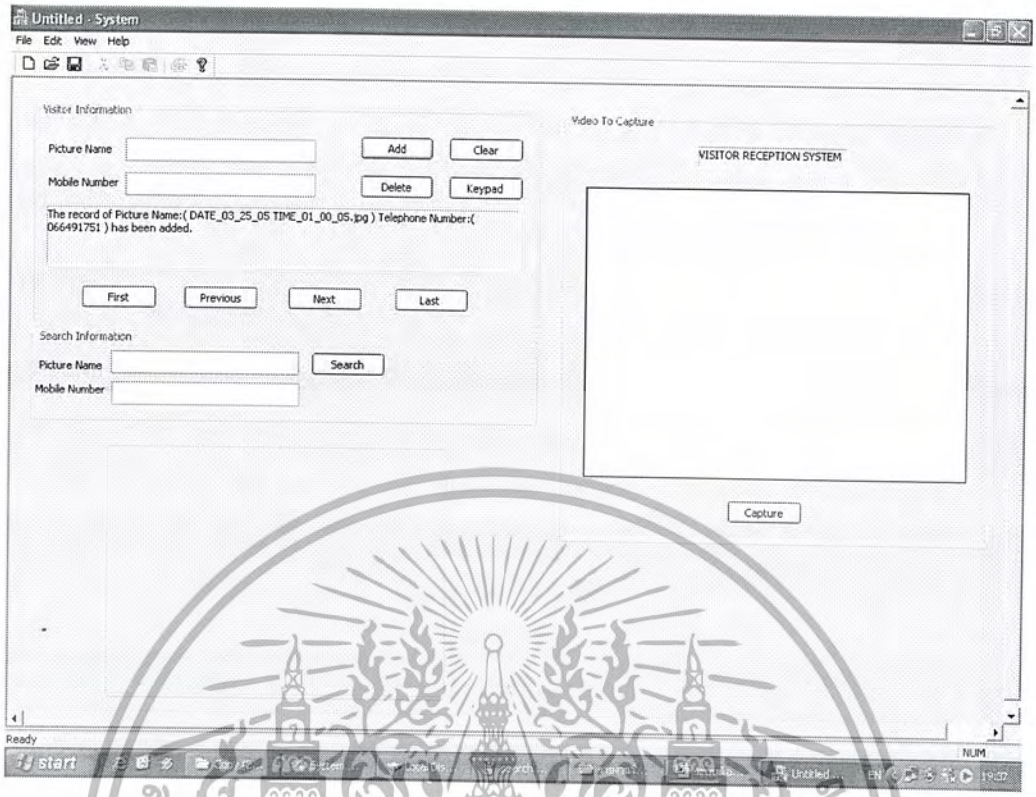
เป็นการแสดงการรับค่าหมายเลขโทรศัพท์จากการกดคีย์แพดของผู้ใช้งานเพื่อบันทึกข้อมูล

4.2.2 ผลการทดลองในส่วนระบบฐานข้อมูล

เป็นตัวอย่างการรับและการแสดงผลในกรณีที่สามารถรับค่าในส่วนของการถ่ายภาพได้ ซึ่งในที่นี้เป็นการกำหนด Picture Name ขึ้นมาเอง พร้อมกับการรับหมายเลขโทรศัพท์จากการกดคีย์แพด ซึ่งการทำงานของระบบฐานข้อมูลจะเป็นดังต่อไปนี้

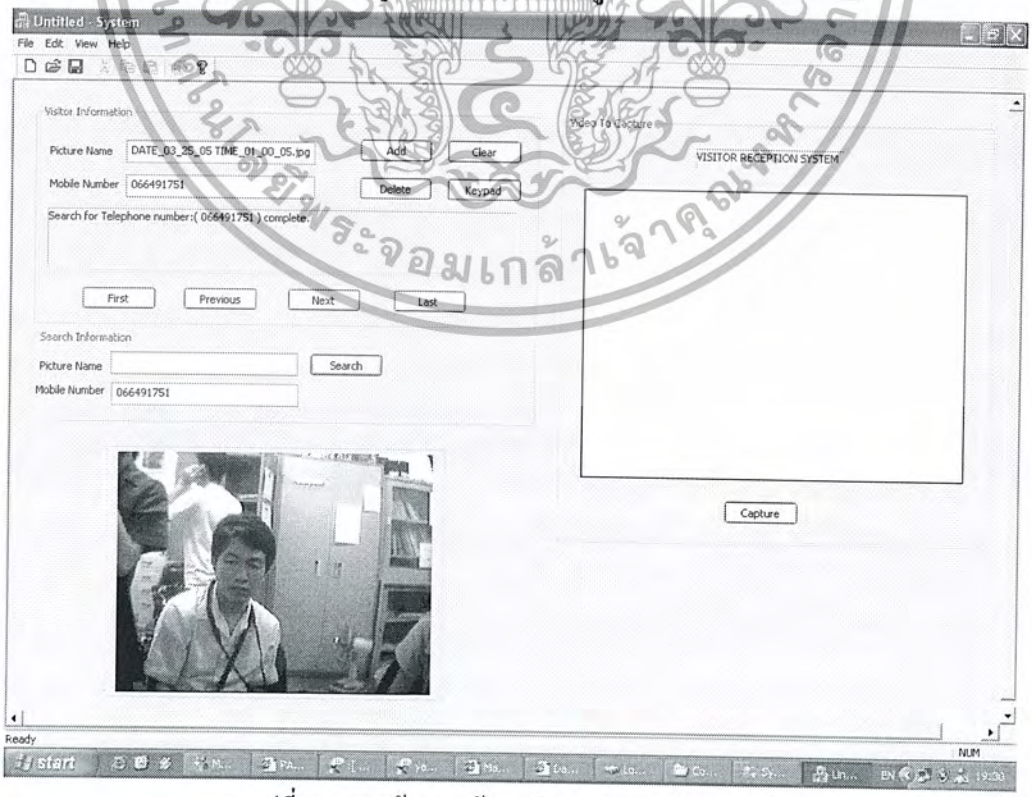
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2.1 การทำงานในส่วนของการรับและบันทึกข้อมูล



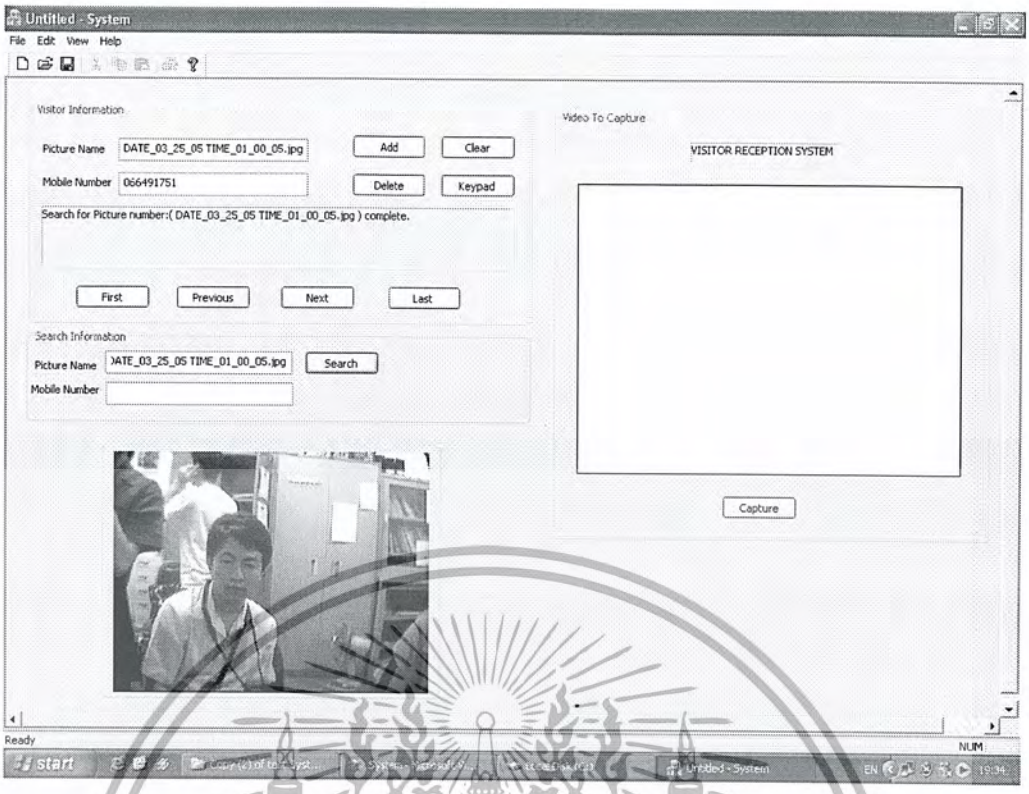
รูปที่ 4.4 การรับข้อมูลของระบบฐานข้อมูลและบันทึกข้อมูล

4.2.2.2 การทำงานในส่วนของการเรียกดูรายละเอียดของข้อมูล



รูปที่ 4.5 การค้นหาดูข้อมูลโดยหมายเลขโทรศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

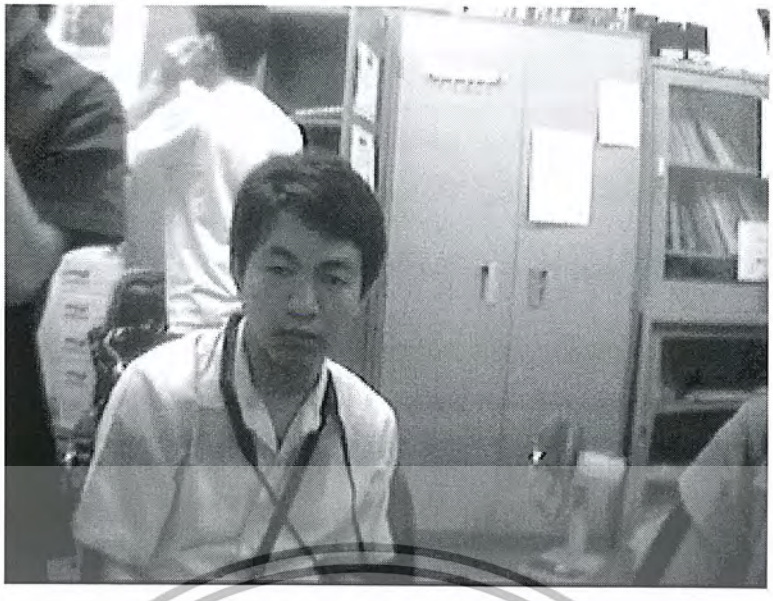


รูปที่ 4.6 การค้นหาดูข้อมูลโดยชื่อภาพ

4.2.3 ผลการทดลองในส่วนของการจับภาพ

รูปที่ 4.7 การถ่ายภาพของกล้องโทรทัศน์วงจรปิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 รูปภาพที่ได้จากการจับภาพของการถ่ายภาพจากกล้องโทรทัศน์วงจรปิด
โดยการกดปุ่ม Capture ภาพที่ได้จะถูกเก็บไว้ที่ Local Disk (C:)

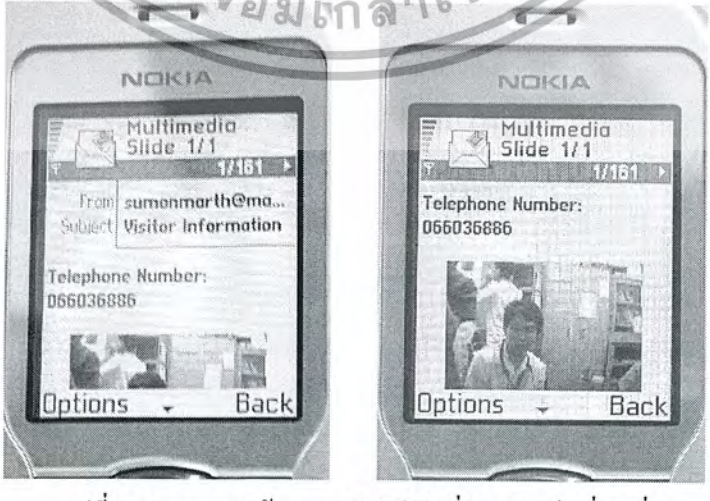
4.3 การทำงานในส่วนของการส่งจดหมายอิเล็กทรอนิกส์

ในการทดลอง ได้ทำการทดลองส่งจดหมายอิเล็กทรอนิกส์ ไปยังเซิร์ฟเวอร์ของผู้ให้บริการรายต่างๆ

4.3.1 ผลการทดลองส่วนของผู้ให้บริการและการรับข้อมูลของโทรศัพท์เคลื่อนที่

การทดลองจะทดลองกับเซิร์ฟเวอร์ของผู้ให้บริการ 2 รายคือ บริษัท แอดวานซ์อินโฟเซอร์วิส จำกัด และ บริษัท โทเทิลแอคเซสคอมมูนิเคชั่น จำกัด

4.3.1.1 ผลการทดลองเมื่อส่งไปยังเซิร์ฟเวอร์ของบริษัท แอดวานซ์อินโฟเซอร์วิส จำกัด

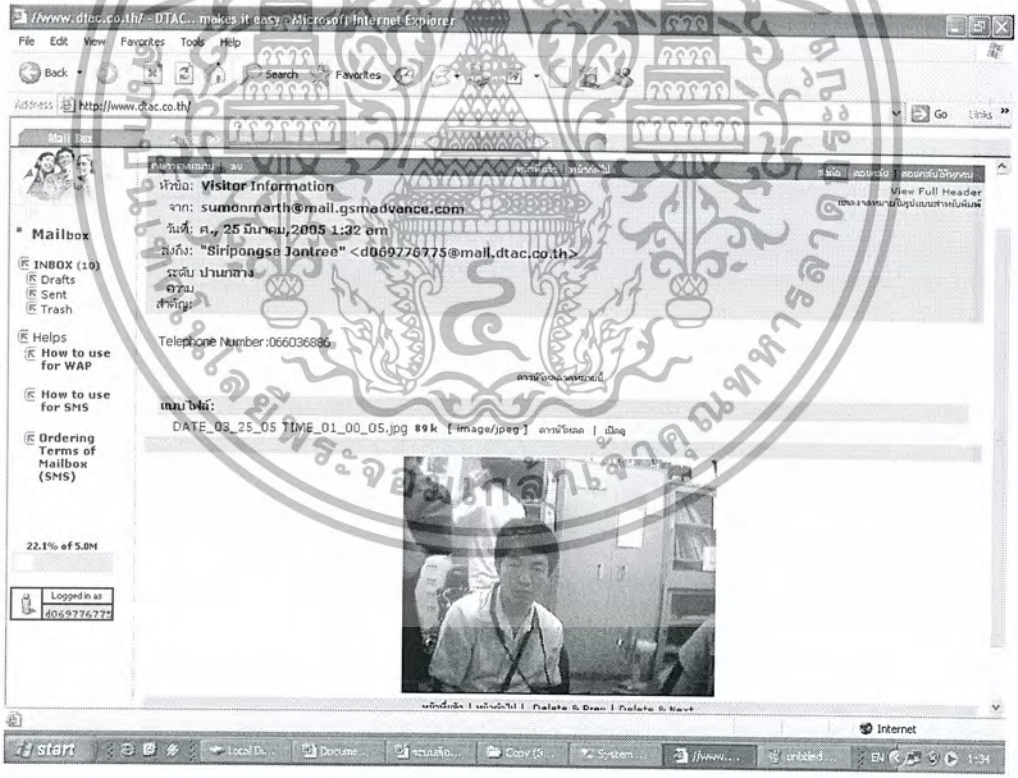


รูปที่ 4.9 การแสดงข้อความมัลติมีเดียที่โทรศัพท์เคลื่อนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.1.2 ผลการทดลองเมื่อส่งไปยังเซิร์ฟเวอร์ของบริษัท โทเทิลแอคเซสคอมมูนิเคชั่น จำกัด

รูปที่ 4.10 การแสดงข้อความสั้นแจ้งเตือนที่โทรศัพท์เคลื่อนที่



รูปที่ 4.11 ผลการรับจดหมายอิเล็กทรอนิกส์ของ http://dtac.co.th

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทวิจารณ์และสรุปผล

5.1 สรุปผลการทดลอง

5.1.1 การทำงานในส่วนของอุปกรณ์และไมโครคอนโทรลเลอร์

- วงจรบันทึกเสียงสามารถใช้งานร่วมกับคีย์แพดในการโต้ตอบกับผู้ใช้ รวมทั้งสามารถรับฝากหมายเลขโทรศัพท์ได้
- ไมโครคอนโทรลเลอร์สามารถควบคุมการทำงานระหว่างวงจรบันทึกเสียง, คีย์แพด และวงจรกริ่ง ให้ทำงานสัมพันธ์กันได้
- สามารถส่งข้อมูลผ่านพอร์ตอนุกรมโดย RS-232 ได้

5.1.2 การทำงานในส่วนของ VC++.NET

- สามารถรับข้อมูลจากพอร์ตอนุกรมที่ผ่านมาทาง RS-232 ได้
- สามารถควบคุมการจับภาพของกล้องโทรทัศน์วงจรปิดและบันทึกภาพได้
- สามารถสร้างฐานข้อมูลโดยใช้คลาส CObList (ข้อมูล) ร่วมกับคลาส CFromview ได้
- สามารถส่งจดหมายอิเล็กทรอนิกส์ไปยังเซิร์ฟเวอร์ของผู้ให้บริการ เพื่อส่งข้อความมัลติมีเดียไปยังโทรศัพท์เคลื่อนที่ได้

5.1.3 การทำงานในส่วนของผู้ให้บริการและการรับข้อมูลของโทรศัพท์เคลื่อนที่

- เครื่องรับโทรศัพท์เคลื่อนที่สามารถรับข้อความมัลติมีเดียซึ่งเป็นรูปภาพ พร้อมด้วยหมายเลขโทรศัพท์จากเซิร์ฟเวอร์ของผู้ให้บริการได้

5.2 ปัญหาที่พบในการทดลอง

- การแสดงผลที่เครื่องรับโทรศัพท์เคลื่อนที่ขึ้นอยู่กับขีดความสามารถของตัวเครื่องรับ รวมถึงลักษณะการให้บริการของผู้ให้บริการโทรศัพท์เคลื่อนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

กฤษฎา ใจเย็น, ชัยวัฒน์ ลิ้มพรจิตวิไล และอรรถพล บุญยะโภคา. **เรียนรู้และปฏิบัติการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกผ่านพอร์ตอนุกรม** กรุงเทพฯ: บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด, 2521.

ชัยวัฒน์ ลิ้มพรจิตวิไล และวรพจน์ กรแก้ววัฒนกุล. **เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์ MCS-51 Flash Microcontroller** กรุงเทพฯ: บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด, 2542.

ยุทธนา ลีลาศวัฒนกุล. **คู่มือการเขียนโปรแกรมและการใช้งาน Visual C++ .NET ฉบับสมบูรณ์** นนทบุรี: อินโฟเพรส, 2546.

ยุทธนา ลีลาศวัฒนกุล. **คู่มือการเขียนโปรแกรมวินโดวส์ขั้นสูงด้วย Visual C++ .NET Episode One** กรุงเทพฯ: บริษัท ดวงกมลสมัย จำกัด, 2546

ยุทธนา ลีลาศวัฒนกุล. **เริ่มต้นการเขียนโปรแกรมด้วยภาษา C++** กรุงเทพฯ: บริษัท ดวงกมลสมัย จำกัด, พิมพ์ครั้งที่ 2, 2547.

สมยศ จุณณะปิยะ. **การประยุกต์ใช้งานไมโครคอนโทรลเลอร์** กรุงเทพฯ: คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2543.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซอร์สโค้ดของโปรแกรมที่ใช้ควบคุมไมโครคอนโทรลเลอร์

```
                ORG    0000H
                STA    EQU P2.0
                RES    EQU P2.1
                SHIFT  EQU P2.2
                EOM    EQU P2.3
                MUTE   EQU P2.4
                MOV    TMOD,#20H
                MOV    TH1,#0FDH
                MOV    SCON,#50H
                MOV    PCON,#00H
                SETB   TR1
                SETB   P0.3
                CLR    STA
                CLR    RES
                CLR    SHIFT
                CLR    MUTE
                LCALL  RESET_SOUND

BELL:           JB     P0.3,$
SND:            MOV    A,#01H
                LCALL  TSND
                LCALL  DELAY_SOUND
                MOV    A,#02H
                LCALL  TSND
                MOV    R2,#09H
                MOV    R0,#3FH
                MOV    20H,#00H

SCAN_KEY:       MOV    A,20H
                JB     ACC.0,SETF1          ;LED
                CLR    P2.5
                SJMP   status2
SETF1:          SETB   P2.5
status2:        JB     ACC.1,SETF2
                CLR    P2.6
                SJMP   status3
SETF2:          SETB   P2.6
status3:        JB     ACC.2,SETF3
                CLR    P2.7
                SJMP   status4
SETF3:          SETB   P2.7
status4:        JB     ACC.3,SETF4
                CLR    P3.4
                SJMP   status5
SETF4:          SETB   P3.4
status5:        LCALL  KEY1                 ;SCAN KEY
                MOV    A,30H
                CJNE   A,#10H,ZERO          ;DELAY KEY
                LJMP   BELL
ZERO:           CJNE   A,#0BH,YES           ;0
                MOV    A,#00H
                LJMP   REC
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
YES:        CJNE  A,#0CH,NOSS       ;#
            LJMP  SCAN_KEY

NOSS:       CJNE  A,#0AH,REC        ;*
            LJMP  SCAN_KEY

REC:        MOV   @R0,A             ;RECORD VALUE
            INC   R0
            INC   20H
            DJNZ  R2,SCAN_KEY

            MOV   A,#02H
            LCALL TSND              ;YOUR NO.

            MOV   R2,#09H
            MOV   R0,#3FH

PLAYSOUND:  MOV   A,@R0
            ADD   A,#03H
            LCALL TSND
            LCALL DELAY_SOUND
            INC   R0
            DJNZ  R2,PLAYSOUND
            MOV   A,#13H
            LCALL TSND

CONFIRM:    LCALL KEY1
            MOV   A,30H
            CJNE  A,#10H,YESS
            LJMP  BELL
YESS:       CJNE  A,#01H,NOSSS
            MOV   R2,#09H
            MOV   R0,#3FH

TX:         MOV   A,@R0
ASC:        ADD   A,#30H
SEND:       MOV   SBUF,A
            JNB   TI,$
            CLR   TI
            INC   R0
            DJNZ  R2,TX
            MOV   A,#14H
            LCALL TSND
            LJMP  BELL
NOSSS:      CJNE  A,#0AH,CONFIRM
RETSND:     LJMP  SND

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;; SUB   SCAN   KEY;;;;;;;;;;;;;;;;;;;;;;;;;;;

KEY1:       MOV   50H,#05H          ;DELAY KEYPAD
KEY2:       MOV   R6,#0FFH
KEY3:       MOV   R7,#0FFH
KEY:        MOV   R3,#10111111B
            MOV   R4,#04H
            SETB  P0.0
            SETB  P0.1
            SETB  P0.2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SCAN:       MOV   A,R3
            MOV   31H,R3
            MOV   P1,A
INKEY:      MOV   A,P0
            ANL   A,#07H
            CJNE  A,#07H,KEYDOWN
            MOV   A,R3
            RR    A
            MOV   R3,A
            DJNZ  R4,SCAN

            DJNZ  R7,KEY
            DJNZ  R6,KEY3
            DJNZ  50H,KEY2
            MOV   30H,#10H
            RET

KEYDOWN:
            MOV   60H,A
            MOV   R1,#28H
            LCALL DELAY
            MOV   A,P0
            ANL   A,#07H
            CJNE  A,60H,KEY
BACK:
            MOV   A,P0
            ANL   A,#07H
            CJNE  A,#07H,BACK
RE:
            MOV   A,P0
            ANL   A,#07H
            MOV   61H,A
            MOV   R1,#28H
            LCALL DELAY

            MOV   A,P0
            ANL   A,#07H
            CJNE  A,61H,KEY
RELESE:     LJMP  CHKVAL

      ;;;;;;;;;;;;;;;;;;;;;;;; check value ;;;;;;;;;;;;;;;;;;;;;;;;

CHKVAL:     MOV   A,R3
ROW1:       ANL   A,#01111000B
            CJNE  A,#00111000B,ROW2
            MOV   30H,#01H
            AJMP  COL
ROW2:       CJNE  A,#01011000B,ROW3
            MOV   30H,#04H
            AJMP  COL
ROW3:       CJNE  A,#01101000B,ROW4
            MOV   30H,#07H
            AJMP  COL
ROW4:       CJNE  A,#01110000B,RTURN
            MOV   30H,#0AH
            AJMP  COL
RTURN:      RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;; COL ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

COL:        MOV     A,60H
            ANL     A,#00000111B
COL1:       CJNE    A,#00000110B,COL2
            MOV     A,30H
            RET
COL2:       CJNE    A,#00000101B,COL3
            MOV     A,30H
            INC     A
            MOV     30H,A
            RET
COL3:       CJNE    A,#00000011B,CTURN
            MOV     A,30H
            INC     A
            INC     A
            MOV     30H,A
CTURN:      RET

;;;;;;;;;;;;;;;;;;;;;;;;;;;;Subroutine_TSND;;;;;;;;;;;;;;;;;;;;;;;;;;;;

TSND:               CJNE        A,#01H,SOUND2
                    ACALL       SOUND_1
                    AJMP        SOUND_END

SOUND2:             CJNE        A,#02H,SOUND3
                    ACALL       SOUND_2
                    AJMP        SOUND_END

SOUND3:             CJNE        A,#03H,SOUND4
                    ACALL       SOUND_3
                    AJMP        SOUND_END

SOUND4:             CJNE        A,#04H,SOUND5
                    ACALL       SOUND_4
                    AJMP        SOUND_END

SOUND5:             CJNE        A,#05H,SOUND6
                    ACALL       SOUND_5
                    AJMP        SOUND_END

SOUND6:             CJNE        A,#06H,SOUND7
                    ACALL       SOUND_6
                    AJMP        SOUND_END

SOUND7:             CJNE        A,#07H,SOUND8
                    ACALL       SOUND_7
                    AJMP        SOUND_END

SOUND8:             CJNE        A,#08H,SOUND9
                    ACALL       SOUND_8
                    AJMP        SOUND_END

SOUND9:             CJNE        A,#09H,SOUND10
                    ACALL       SOUND_9
                    AJMP        SOUND_END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SOUND10:        CJNE        A,#0AH,SOUND11
                ACALL       SOUND_10
                AJMP        SOUND_END

SOUND11:        CJNE        A,#0BH,SOUND12
                ACALL       SOUND_11
                AJMP        SOUND_END

SOUND12:        CJNE        A,#0CH,SOUND13
                ACALL       SOUND_12
                AJMP        SOUND_END

SOUND13:        CJNE        A,#13H,SOUND14
                ACALL       SOUND_13
                AJMP        SOUND_END

SOUND14:        CJNE        A,#14H,SOUND_END
                ACALL       SOUND_14
                AJMP        SOUND_END

SOUND_END:      RET

;;;;;;;;;;;;;;;;;;;;;;;;;;;;Subroutine_REC;;;;;;;;;;;;;;;;;;;;;;;;

          CLR   STA
          CLR   RES
          CLR   SHIFT
          CLR   MUTE
          MOV   P2,#0FFH

SOUND_1:  SETB      MUTE  ;HELLO
          LCALL     DELAY_SOUND
          LCALL     RESET_SOUND
          LCALL     DELAY_SOUND
          LCALL     START_SOUND
          LCALL     WAIT_SOUND
          CLR             MUTE
          RET

SOUND_2:
          LCALL     DELAY_SOUND ;YOUR NO.
          LCALL     RESET_SOUND
          LCALL     DELAY_SOUND
          SETB      SHIFT
          LCALL     DELAY_SOUND
          LCALL     START_SOUND
          LCALL     DELAY_SOUND
          CLR             SHIFT
          SETB      MUTE
          LCALL     DELAY_SOUND
          LCALL     START_SOUND
          LCALL     WAIT_SOUND
          CLR             MUTE
          RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SOUND_3:
        LCALL   DELAY_SOUND ;0
        LCALL   RESET_SOUND
        LCALL   DELAY_SOUND
        SETB    SHIFT
        LCALL   DELAY_SOUND
        LCALL   START_SOUND
        LCALL   DELAY_SOUND
        LCALL   START_SOUND
        LCALL   DELAY_SOUND
        CLR             SHIFT
        SETB    MUTE
        LCALL   DELAY_SOUND
        LCALL   START_SOUND
        LCALL   WAIT_SOUND
        CLR             MUTE
        RET

SOUND_4:
        LCALL   DELAY_SOUND ;1
        LCALL   RESET_SOUND
        LCALL   DELAY_SOUND
        SETB    SHIFT
        LCALL   DELAY_SOUND
        LCALL   START_SOUND
        LCALL   DELAY_SOUND
        LCALL   START_SOUND
        LCALL   DELAY_SOUND
        LCALL   START_SOUND
        LCALL   DELAY_SOUND
        CLR             SHIFT
        SETB    MUTE
        LCALL   DELAY_SOUND
        LCALL   START_SOUND
        LCALL   WAIT_SOUND
        CLR             MUTE
        RET

SOUND_5:
        LCALL   DELAY_SOUND ;2
        LCALL   RESET_SOUND
        LCALL   DELAY_SOUND
        SETB    SHIFT
        LCALL   DELAY_SOUND
        LCALL   START_SOUND
        LCALL   DELAY_SOUND
        LCALL   START_SOUND
        LCALL   DELAY_SOUND
        LCALL   START_SOUND
        LCALL   DELAY_SOUND
        LCALL   START_SOUND
        LCALL   DELAY_SOUND
        CLR             SHIFT
        SETB    MUTE
        LCALL   DELAY_SOUND
        LCALL   START_SOUND
        LCALL   WAIT_SOUND
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            CLR             MUTE
            RET

SOUND_6:
            LCALL       DELAY_SOUND ;3
            LCALL       RESET_SOUND
            LCALL       DELAY_SOUND
            SETB        SHIFT
            LCALL       DELAY_SOUND
            LCALL       START_SOUND
            LCALL       DELAY_SOUND
            LCALL       START_SOUND
            LCALL       DELAY_SOUND
            LCALL       START_SOUND
            LCALL       DELAY_SOUND
            LCALL       START_SOUND
            LCALL       DELAY_SOUND
            LCALL       START_SOUND
            LCALL       DELAY_SOUND
            CLR             SHIFT
            SETB        MUTE
            LCALL       DELAY_SOUND
            LCALL       START_SOUND
            LCALL       WAIT_SOUND
            CLR             MUTE
            RET

SOUND_7:
            LCALL       DELAY_SOUND ;4
            LCALL       RESET_SOUND
            LCALL       DELAY_SOUND
            SETB        SHIFT
            LCALL       DELAY_SOUND
            LCALL       START_SOUND
            LCALL       DELAY_SOUND
            LCALL       START_SOUND
            LCALL       DELAY_SOUND
            LCALL       START_SOUND
            LCALL       DELAY_SOUND
            LCALL       START_SOUND
            LCALL       DELAY_SOUND
            LCALL       START_SOUND
            LCALL       DELAY_SOUND
            LCALL       START_SOUND
            LCALL       DELAY_SOUND
            CLR             SHIFT
            SETB        MUTE
            LCALL       DELAY_SOUND
            LCALL       START_SOUND
            LCALL       WAIT_SOUND
            CLR             MUTE
            RET

SOUND_8:
            LCALL       DELAY_SOUND ;5
            LCALL       RESET_SOUND
            LCALL       DELAY_SOUND
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                SETB        SHIFT
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                CLR             SHIFT
                SETB        MUTE
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       WAIT_SOUND
                CLR             MUTE
                RET

SOUND_9:
                LCALL       DELAY_SOUND ;6
                LCALL       RESET_SOUND
                LCALL       DELAY_SOUND
                SETB        SHIFT
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                CLR             SHIFT
                SETB        MUTE
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       WAIT_SOUND
                CLR             MUTE
                RET

SOUND_10:
                LCALL       DELAY_SOUND ;7
                LCALL       RESET_SOUND
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                LCALL       DELAY_SOUND
                SETB        SHIFT
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                CLR             SHIFT
                SETB        MUTE
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       WAIT_SOUND
                CLR             MUTE
                RET

SOUND_11:

                LCALL       DELAY_SOUND ;8
                LCALL       RESET_SOUND
                LCALL       DELAY_SOUND
                SETB        SHIFT
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                CLR             SHIFT
                SETB        MUTE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       WAIT_SOUND
                CLR             MUTE
                RET

SOUND_12:
                LCALL       DELAY_SOUND ;9
                LCALL       RESET_SOUND
                LCALL       DELAY_SOUND
                SETB        SHIFT
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND

                CLR             SHIFT
                SETB        MUTE
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       WAIT_SOUND
                CLR             MUTE
                RET

SOUND_13:
                LCALL       DELAY_SOUND ;CONFIRM
                LCALL       RESET_SOUND
                LCALL       DELAY_SOUND
                SETB        SHIFT
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                CLR             SHIFT
                SETB        MUTE
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       WAIT_SOUND
                CLR             MUTE
                RET

SOUND_14:

                LCALL       DELAY_SOUND ;THANK YOU
                LCALL       RESET_SOUND
                LCALL       DELAY_SOUND
                SETB        SHIFT
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       DELAY_SOUND
                CLR             SHIFT
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                SETB        MUTE
                LCALL       DELAY_SOUND
                LCALL       START_SOUND
                LCALL       WAIT_SOUND
                CLR         MUTE
                RET

START_SOUND:    SETB        STA
                LCALL       DELAY_SOUND
                CLR         STA
                RET

RESET_SOUND:    SETB        RES
                LCALL       DELAY_SOUND
                CLR         RES
                RET

WAIT_SOUND:
                JB          EOM,WAIT_SOUND
                RET

DELAY_SOUND:    MOV         R5,#060H
DELAY01:        MOV         R7,#0FFH
                DJNZ        R7,$
                DJNZ        R5,DELAY01
                RET

DELAY:          MOV         R1,#04H
DELAY1:         MOV         R4,#00H
                DJNZ        R4,$
                DJNZ        R1,DELAY1
                RET
                END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซอร์สโค้ดของโปรแกรมที่พัฒนาโดยใช้ VC++.NET

```
// SystemView.h : interface of the CSystemView class
//


#pragma once
#include "videoocx.h"
#include "afxwin.h"


class CSystemView : public CFormView
{
protected: // create from serialization only
      CSystemView();
      DECLARE_DYNCREATE(CSystemView)

public:
   .  enum{ IDD = IDD_SYSTEM_FORM };

// Attributes
public:
      CSystemDoc* GetDocument() const;

// Operations
public:

// Overrides
      public:
virtual BOOL PreCreateWindow(CREATESTRUCT& cs);
protected:
      virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV
support
      virtual void OnInitialUpdate(); // called first time after construct

// Implementation
public:
      virtual ~CSystemView();
#ifdef _DEBUG
      virtual void AssertValid() const;
      virtual void Dump(CDumpContext& dc) const;
#endif

protected:

// Generated message map functions
protected:
      DECLARE_MESSAGE_MAP()
public:
      CString m_Picture;
      CString m_Telephone;
public:

      CString m_strshow;
      CString m_find;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        CString m_find2;
        afx_msg void OnBnClickedAddRecord();
        afx_msg void OnBnClickedDeleteRecord();
        afx_msg void OnBnClickedButton2();
        afx_msg void OnBnClickedClear();
        afx_msg void OnBnClickedFirstRecord();
        afx_msg void OnBnClickedPreRecord();
        afx_msg void OnBnClickedNextRecord();
        afx_msg void OnBnClickedLastRecord();
        afx_msg void OnBnClickedSearchRecord();
protected:
        virtual void OnUpdate(CView* /*pSender*/, LPARAM /*lHint*/, CObject*
/*pHint*/);
private:

        CObList* m_viList;
        POSITION m_pCurrent;
        BOOL m_bFlag;
        void ShowRecord(POSITION position);
        void ClearRecord(void);
public:
        BOOL flag1;
        void Add(void);
        CString m_strcomport;
        CString m_picturefound;
        CString m_telephonefound;
        void InitialSerial(CString com);
        CVideoOCX m_Video;
        CButton m_Capture;
protected:
        virtual void OnDraw(CDC* /*pDC*/);
public:
        void Draw(CString filename);
        afx_msg void OnBnClickedWork();
        void WORK2(void);
        void STOP2(void);
        void SAVE(void);
        long m_Image;
        bool m_Running;
        char sumbuffer[50];
        void WORK(void);
        afx_msg void OnCaptureVideosource(void);
        void OnCaptureVideoformat139(void);
        void SendEmail(void);
        CString m_TelephoneMail;
        int heigh;
        int width;
        int BPP;
};

#ifndef _DEBUG  // debug version in SystemView.cpp
inline CSystemDoc* CSystemView::GetDocument() const
   { return reinterpret_cast<CSystemDoc*>(m_pDocument); }
#endif
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// SystemView.cpp : implementation of the CSystemView class
//
#include <string.h>
#include "time.h"
#include "stdafx.h"
#include "System.h"
#include "Visitor.h"
#include "SystemDoc.h"
#include "SystemView.h"
#include "atlimage.h"
#include ".\systemview.h"
#include "PImage.h"

#include "C:/ChilkatLib2/Include/CkSettings.h"
#include "C:/ChilkatLib2/Include/CkEmail.h"
#include "C:/ChilkatLib2/Include/CkEmailBundle.h"
#include "C:/ChilkatLib2/Include/CkMailMan.h"
#include "C:/ChilkatLib2/Include/CkString.h"
#include "C:/ChilkatLib2/Include/CkByteData.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#endif
HANDLE hComp;
DCB dcb;
char sum[15];
CString buff;
CString buff2;
CString strtel;
char chBuff;
int i;
int valReturn;
unsigned long nReadPort;
unsigned long nWritePort;

CPImage image;
// CSystemView

IMPLEMENT_DYNCREATE(CSystemView, CFormView)

BEGIN_MESSAGE_MAP(CSystemView, CFormView)
	ON_BN_CLICKED(IDC_ADD_RECORD, OnBnClickedAddRecord)
	ON_BN_CLICKED(IDC_DELETE_RECORD, OnBnClickedDeleteRecord)
	ON_BN_CLICKED(IDC_BUTTON2, OnBnClickedButton2)
	ON_BN_CLICKED(IDC_CLEAR, OnBnClickedClear)
	ON_BN_CLICKED(IDC_FIRST_RECORD, OnBnClickedFirstRecord)
	ON_BN_CLICKED(IDC_PRE_RECORD, OnBnClickedPreRecord)
	ON_BN_CLICKED(IDC_NEXT_RECORD, OnBnClickedNextRecord)
	ON_BN_CLICKED(IDC_LAST_RECORD, OnBnClickedLastRecord)
	ON_BN_CLICKED(IDC_SEARCH_RECORD, OnBnClickedSearchRecord)
	ON_BN_CLICKED(IDC_WORK, OnBnClickedWork)
END_MESSAGE_MAP()

// CSystemView construction/destruction

CSystemView::CSystemView()
	: CFormView(CSystemView::IDD)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        , m_Picture(_T(""))
        , m_Telephone(_T(""))
        , m_strshow(_T(""))
        , m_find(_T(""))
        , m_find2(_T(""))
        , m_strcomport(_T(""))
        , m_picturefound(_T(""))
        , m_telephonefound(_T(""))
        , m_Image(0)
        , m_Running(false)
        , flag1(false)
        {
        // TODO: add construction code here
        m_pCurrent=0;
        m_viList=0;
        m_bFlag=FALSE;
        InitialSerial("COM1");


}

CSystemView::~CSystemView()
{
}

void CSystemView::DoDataExchange(CDataExchange* pDX)
{
        CFormView::DoDataExchange(pDX);
        DDX_Text(pDX, IDC_PICTURE, m_Picture);
        DDV_MaxChars(pDX, m_Picture, 50);
        DDX_Text(pDX, IDC_TELEPHONE, m_Telephone);
        DDV_MaxChars(pDX, m_Telephone, 50);
        DDX_Text(pDX, IDC_SHOW, m_strshow);
        DDX_Text(pDX, IDC_EDIT1, m_find);
        DDX_Text(pDX, IDC_EDIT2, m_find2);
        DDX_Control(pDX, IDC_VIDEOOCXCTRL1, m_Video);
        DDX_Control(pDX, IDC_WORK2, m_Capture);
}

BOOL CSystemView::PreCreateWindow(CREATESTRUCT& cs)
{
        // TODO: Modify the Window class or styles here by modifying
        //  the CREATESTRUCT cs

        return CFormView::PreCreateWindow(cs);
}

void CSystemView::OnInitialUpdate()
{
        OnUpdate(this,0,0);
        CFormView::OnInitialUpdate();
        GetParentFrame()->RecalcLayout();
        ResizeParentToFit();
}


// CSystemView diagnostics
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#ifdef _DEBUG
void CSystemView::AssertValid() const
{
      CFormView::AssertValid();
}

void CSystemView::Dump(CDumpContext& dc) const
{
      CFormView::Dump(dc);
}

CSystemDoc* CSystemView::GetDocument() const // non-debug version is inline
{
      ASSERT(m_pDocument->IsKindOf(RUNTIME_CLASS(CSystemDoc)));
      return (CSystemDoc*)m_pDocument;
}
#endif //_DEBUG


// CSystemView message handlers

//}


void CSystemView::OnBnClickedAddRecord()
{// TODO: Add your control notification handler code here
      m_bFlag = TRUE;
      UpdateData(TRUE);
      if (m_Picture.IsEmpty())
      {     AfxMessageBox("Please enter the Picture Name!");
            return;
      }
      if (m_Telephone.IsEmpty())
      {     AfxMessageBox("Please enter the Telephone Number!");
            return;
      }
      if (!m_viList->IsEmpty()){
            m_pCurrent = m_viList->GetHeadPosition();
            CVisitor *pVisitor = (CVisitor *) m_viList->GetAt(m_pCurrent);
            m_picturefound=pVisitor->GetPicture();
            m_telephonefound=pVisitor->GetTelephone();

            if (m_picturefound==m_Picture)
            {     m_strshow="The record for Picture Name:( "+m_Picture+"
)is already exists.";
                  UpdateData(FALSE);
                  return;
                  }
            if (m_telephonefound==m_Telephone)
            {     m_strshow="The record for Telephone Number:(
"+m_Telephone+" ) is already exists.";
                  UpdateData(FALSE);
                  return;
            }
            else
            {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                for (long int l=2;l<=(m_viList->GetSize());l++)
                {
                      m_viList->GetNext(m_pCurrent);
                      if (m_pCurrent)
                      {     CVisitor *pVisitor = (CVisitor *) m_viList-
>GetAt(m_pCurrent);
                            m_picturefound=pVisitor->GetPicture();
                            m_telephonefound=pVisitor->GetTelephone();
                            if (m_picturefound==m_Picture)
                               {m_strshow="The record for Picture Name:(
"+m_Picture+" ) is already exists.";
                                     UpdateData(FALSE);
                                     return;
                               }
                            if (m_telephonefound==m_Telephone)
                            {     m_strshow="The record for Telephone
Number:( "+m_Telephone+" ) is already exists.";
                                  UpdateData(FALSE);
                                  return;
                            }
                      }

                }
          }
      }
      Add();
      UpdateData(TRUE);
      m_strshow="The record of Picture Name:( "+m_Picture+" ) Telephone
Number:( "+m_Telephone+" ) has been added.";
      ClearRecord();
      UpdateData(FALSE);
}

void CSystemView::OnBnClickedDeleteRecord()
{
      // TODO: Add your control notification handler code here
      m_bFlag = FALSE;
      CSystemDoc* pDoc = GetDocument();
      ASSERT_VALID(pDoc);
      POSITION pos;
      CObject * pE = 0;
      pos = m_pCurrent;
      if (AfxMessageBox("Are you sure you want to delete this
record?",MB_YESNO) != IDYES)
            return;
      if (m_pCurrent !=NULL)
      {     m_viList->GetNext(pos);
            if (pos == NULL)
            {     pos = m_viList->GetHeadPosition();
                  if (pos == m_pCurrent)
                        pos = NULL;
            }
            ShowRecord(pos);
            pE = m_viList->GetAt(m_pCurrent);
            m_viList->RemoveAt(m_pCurrent);
            delete pE;
            m_pCurrent = pos;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            pDoc->SetModifiedFlag();
            pDoc->UpdateAllViews(this);
            m_strshow="The record has been deleted.";
            UpdateData(FALSE);
      }
      else
      {     AfxMessageBox("No Record for deletion!!!",MB_OK |
MB_ICONINFORMATION);
            OnBnClickedAddRecord();
      }
}

void CSystemView::OnBnClickedButton2()//ปุ่มKEYPAD นะ
{
      CString ab;
      CString ba;
      UpdateData(TRUE);
      ab=m_find;
      ba=m_find2;
            BYTE bb='R';
            DWORD iBytesWritten=0;
            if(WriteFile(hComp,&bb,1,&iBytesWritten,NULL)==0)
            exit(2);
      for (int i=0;i<=8;i++)
      {
            ReadFile(hComp,&chBuff,1,&nReadPort,NULL);
            if (chBuff=='?')
                  break;
            sum[i]=chBuff;
      }

      buff=sum;
      m_Telephone=buff;
      m_strshow=(_T(""));
      m_Picture=(_T(""));
      m_find=ab;
      m_find2=ba;
      UpdateData(FALSE);



}/*   // TODO: Add your control notification handler code here
CString ab;
      CString ba;
      UpdateData(TRUE);
      ab=m_find;
      ba=m_find2;
if (dcb.fInX=TRUE)
      {for ( int i=0;i<=8;i++)
            {
                  ReadFile(hComp,&chBuff,1,&nReadPort,NULL);
                  sum[i]=chBuff;          }
                  buff=sum;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    m_Telephone=buff;
                    m_strshow=(_T(""));
                    m_Picture=(_T(""));
                    m_find=ab;
                    m_find2=ba;
                    UpdateData(FALSE);}

}*/

void CSystemView::OnBnClickedClear()
{
      // TODO: Add your control notification handler code here
      //SendEmail();
      m_Picture=(_T(""));
      m_Telephone=(_T(""));
      m_strshow=(_T(""));
      m_find=(_T(""));
      m_find2=(_T(""));
      UpdateData(FALSE);
}

void CSystemView::OnBnClickedFirstRecord()
{
      // TODO: Add your control notification handler code here
            m_bFlag = FALSE;
      if (!m_viList->IsEmpty())
      {     m_pCurrent = m_viList->GetHeadPosition();
            ShowRecord(m_pCurrent);
            m_strshow=(_T(""));
            UpdateData(FALSE);
      }
      else
      {AfxMessageBox("There is no record.",MB_OK|MB_ICONINFORMATION);
      m_strshow=(_T(""));
      UpdateData(FALSE);}
}

void CSystemView::OnBnClickedPreRecord()
{
      // TODO: Add your control notification handler code here
      m_bFlag = FALSE;
      POSITION pos;
      pos = m_pCurrent;
      if (pos != NULL)
      {     m_viList->GetPrev(pos);
            if(pos)
            {     m_pCurrent = pos;
                  ShowRecord(m_pCurrent);
                  m_strshow=(_T(""));
                  UpdateData(FALSE);
            }
            else
            {AfxMessageBox("First record
already.",MB_OK|MB_ICONINFORMATION);
            m_strshow=(_T(""));
            UpdateData(FALSE);}
      }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        else
        {AfxMessageBox("There is no record.",MB_OK|MB_ICONINFORMATION);
        m_strshow=(_T(""));
        UpdateData(FALSE);}

        UpdateData(FALSE);
}

void CSystemView::OnBnClickedNextRecord()
{
        // TODO: Add your control notification handler code here
        m_bFlag = FALSE;
        POSITION pos;
        pos = m_pCurrent;
        if (pos != NULL)
        {       m_viList->GetNext(pos);
                if (pos)
                {       m_pCurrent = pos;
                        ShowRecord(m_pCurrent);
                        m_strshow=(_T(""));
                        UpdateData(FALSE);
                }
                else
                {       AfxMessageBox("Last record
already.",MB_OK|MB_ICONINFORMATION);
                m_strshow=(_T(""));
                UpdateData(FALSE);}
        }
        else {AfxMessageBox("There is no record.",MB_OK|MB_ICONINFORMATION);
        m_strshow=(_T(""));
        UpdateData(FALSE);}
}
void CSystemView::OnBnClickedLastRecord()
{
        // TODO: Add your control notification handler code here
        m_bFlag = FALSE;
        if (!m_viList->IsEmpty())
        {       m_pCurrent = m_viList->GetTailPosition();
                ShowRecord(m_pCurrent);
                m_strshow=(_T(""));
                UpdateData(FALSE);
        }
        else
        {AfxMessageBox("There is no record.", MB_OK|MB_ICONINFORMATION);
        m_strshow=(_T(""));
        UpdateData(FALSE);}
}

void CSystemView::OnBnClickedSearchRecord()
{
        // TODO: Add your control notification handler code here
        UpdateData(TRUE);
if ((m_find.IsEmpty())&&(m_find2.IsEmpty()))
{       AfxMessageBox("Please enter data to  find.");
        return;
}
if ((!m_find.IsEmpty())&&(!m_find2.IsEmpty()))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{     AfxMessageBox("Please do not enter 2 data at the same time.");
      return;
}
      // TODO: Add your control notification handler code here
      m_bFlag=FALSE;
      if (!m_viList->IsEmpty())
      {     m_pCurrent = m_viList->GetHeadPosition();
            CVisitor *pVisitor = (CVisitor *) m_viList->GetAt(m_pCurrent);
            m_picturefound=pVisitor->GetPicture();
            m_telephonefound=pVisitor->GetTelephone();

            if ((m_picturefound==m_find)||(m_telephonefound==m_find2))
            {     m_Picture=m_picturefound;
                  m_Telephone=m_telephonefound;
                  //Draw(m_picturefound);
                  if (m_find==(_T("")))
                  m_strshow="Search for Telephone number:( "+m_find2+ " )
complete.";
                  if (m_find2==(_T("")))
                  m_strshow="Search for Picture number:( "+m_find+ " )
complete.";
                  UpdateData(FALSE);
            }
            else
                  {

                        for (long int l=2;l<=(m_viList->GetSize());l++)
                        {
                        m_viList->GetNext(m_pCurrent);
                        if (m_pCurrent)
                              {     CVisitor *pVisitor = (CVisitor *)
m_viList->GetAt(m_pCurrent);
                                    m_picturefound=pVisitor->GetPicture();
                                    m_telephonefound=pVisitor-
>GetTelephone();
                                    if
((m_picturefound==m_find)||(m_telephonefound==m_find2))
                                    {m_Picture=m_picturefound;
                                    m_Telephone=m_telephonefound;
                                    //Draw(m_picturefound);
                                    if (m_find==(_T("")))
                                    m_strshow="Search for Telephone
number:( "+m_find2+ " ) complete.";

                                    if (m_find2==(_T("")))
                                    m_strshow="Search for Picture
Number:( "+m_find+ " ) complete.";

                                    UpdateData(FALSE);
                                    return;
                                    }

                              }

                        }

                  AfxMessageBox("Record not found.",
MB_OK|MB_ICONINFORMATION);

                  }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
      else
            AfxMessageBox("There is no record.",MB_OK|MB_ICONINFORMATION);

      CClientDC dc(this);
      CString name;
      name.Format("c:\\" + m_picturefound);
      image.Load(name);

      HBITMAP hBit = (HBITMAP)  image;
            Sleep(100);
      if (image == NULL)
      {
            AfxMessageBox("Please open a picture
before.",MB_OK|MB_ICONINFORMATION);
            return;
      }

       if (image.IsIndexed())
      {AfxMessageBox("Indexed image can not be transformed. Please open RGB
picture.",MB_OK|MB_ICONINFORMATION);
            return;}
            BPP=image.GetBPP();
            width=image.GetWidth();
            heigh=image.GetHeight();
            //Mirror();
            image.StretchBlt(dc,110,370,width,heigh,SRCCOPY);
            //image.StretchBlt(dc,-20,-20,width*(2.5),heigh*(2.5),SRCCOPY);
            image.Detach();
            image.Destroy();
}



void CSystemView::OnUpdate(CView* /*pSender*/, LPARAM /*lHint*/, CObject*
/*pHint*/)
{
      // TODO: Add your specialized code here and/or call the base class
      m_viList=GetDocument()->GetVisitorList();
      m_pCurrent=m_viList->GetHeadPosition();
      ShowRecord(m_pCurrent);
}

void CSystemView::ShowRecord(POSITION position)
{if (position)
      {
      CVisitor *pVisitor=(CVisitor*) m_viList->GetAt(position);
      m_Picture=pVisitor->GetPicture();
      m_Telephone=pVisitor->GetTelephone();
      }
      else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        { ClearRecord();
        }
        UpdateData(FALSE);
}

void CSystemView::ClearRecord(void)
{       m_Picture=_T("");
        m_TelephoneMail = m_Telephone;
        m_Telephone=_T("");
}

void CSystemView::Add(void)
{CSystemDoc* pDoc = GetDocument();
        ASSERT_VALID(pDoc);
        UpdateData(TRUE);

        CVisitor * pVisitor = new CVisitor;
              pVisitor->SetPicture(m_Picture);
              pVisitor->SetTelephone(m_Telephone);
        if (m_bFlag)
        {       //m_bFlag=FALSE;
        if (!m_viList->IsEmpty())
              { m_pCurrent = m_viList->GetTailPosition();
                m_pCurrent = m_viList->InsertAfter(m_pCurrent, pVisitor);
              }
        else
              { m_pCurrent = m_viList->InsertBefore(m_pCurrent, pVisitor);
              }
        }
        else
        {       BOOL btailStatus = FALSE;
              POSITION pos;
              CObject * pE = 0;
              pos = m_pCurrent;
              if (m_pCurrent !=NULL)
              {       m_viList->GetNext(pos);
                    if (pos == NULL)
                    {       btailStatus = TRUE;
                          pos = m_viList->GetHeadPosition();
                          if (pos == m_pCurrent)
                                pos = NULL;
                    }
                    pE = m_viList->GetAt(m_pCurrent);
                    m_viList->RemoveAt(m_pCurrent);
                    delete pE;
                    m_pCurrent = pos;
                    if (pos == m_viList->GetHeadPosition() && btailStatus)
                    {       m_pCurrent = m_viList->GetTailPosition();
                          m_pCurrent = m_viList-
>InsertAfter(m_pCurrent,pVisitor);
                    }
                    else  m_pCurrent = m_viList-
>InsertBefore(m_pCurrent,pVisitor);
              }
        }
        pDoc->SetModifiedFlag();
        pDoc->UpdateAllViews(this);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}


void CSystemView::InitialSerial(CString com)
{     m_strcomport=com;
hComp = CreateFile(m_strcomport, GENERIC_READ | GENERIC_WRITE, 0,
                                          NULL, OPEN_EXISTING, 0,NULL    );


if (hComp == INVALID_HANDLE_VALUE) {
   AfxMessageBox("Can not use "+m_strcomport);

   return;
   }

valReturn = GetCommState(hComp, &dcb);

if (!valReturn) {
      AfxMessageBox("Can not use "+m_strcomport);

   return;
}

  dcb.BaudRate = CBR_9600;
  dcb.ByteSize = 8;
  dcb.Parity = NOPARITY;
  dcb.StopBits = ONESTOPBIT;

  valReturn = SetCommState(hComp, &dcb);

  if (!valReturn) {
     m_strshow="Com Port Setting Error";
     UpdateData(FALSE);
       return;
  }

}


void CSystemView::STOP2(void)
            {
                  if (m_Running)
      {     m_Running = FALSE;       // tell thread to end
      Sleep(100);                 // wait for thread to end

      // end capture mode
      if (!m_Video.Stop())
      {
            // error
            MessageBox(m_Video.GetLastErrorString(), "VideoOCX Error",
MB_OK);
      }}
      if (!m_Video.Close())
      {
            // error while closing video connection
            MessageBox(m_Video.GetLastErrorString(), "VideoOCX Error",
MB_OK);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      }
      // free memory
      m_Video.ReleaseImageHandle(m_Image);
      m_Image = 0;
      }


void CSystemView::SAVE(void)
{     char dbuffer [9];
    char tbuffer [9];
   _strdate( dbuffer );
   _strtime( tbuffer );

      strcpy(sumbuffer,"c:\\DATE_");
      strcat(sumbuffer,dbuffer);
      strcat(sumbuffer," TIME_");
      strcat(sumbuffer,tbuffer);
      strcat(sumbuffer,".jpg");

for (int i=2;i<=50;i++)
{     if ((sumbuffer[i]=='/') || (sumbuffer[i]==':'))
            sumbuffer[i]='_';
}

            if (m_Image != 0)
      {     m_Video.SaveJPEG(m_Image, 100,sumbuffer);
            UpdateData(TRUE);
            m_Picture = sumbuffer;
            UpdateData(FALSE);
            /*if (!m_Video.SaveJPEG(m_Image, 50, "neung"))
                  // Error while saving JPG

                  MessageBox(m_Video.GetLastErrorString(), "VideoOCX
Error", MB_OK);
            else
                  // ok...
                  MessageBox("Picture saved.", "Sysytem", MB_OK +
MB_ICONINFORMATION);*/
      }
      else
            // nothing to save
            MessageBox("Nothing to save. Press Capture first.", "System",
MB_OK);
}

void CSystemView::OnBnClickedWork()
{
      WORK2();
}




UINT CaptureThread(LPVOID pParam)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
	// Capture Thread
	CSystemView *dlg;	// reference to parent dialog

	// Get pointer to parent dialog
	dlg = (CSystemView *)pParam;

	// capture loop
	while (dlg->m_Running)
	{
		//if(!m_Pause)
		//{
			// try to capture, if Capture returns FALSE, no image was
available
			// at this time.
			if (dlg->m_Video.Capture(dlg->m_Image))
			{

				// ----------------------------------------------
---------------
				// This is the place for you to implement any
functionality on the
				// captured image:
				//
				// --> DO SOMETHING WITH THE IMAGE DATA <--
				//
				// ----------------------------------------------
---------------
				// EXAMPLE: Threshold each byte

					// // Get Handle to image data
					// BYTE *image;
					// image = (BYTE*)dlg-
>m_Video.GetDataPointer(dlg->m_Image);
					// // get data size
					// int size;
					// size = dlg->m_Video.GetImageDataSize(dlg-
>m_Image);
					// // Run through image
					// for (int j = 0; j < size; j++)
					// {
					//	// Threshold values that are
smaller than 100
					//		if (image[j] < 100)
					//			image[j] = 0;
					// }

				// END OF EXAMPLE
				// ----------------------------------------------
---------------

				// if capturing was successful and Preview/Overlay
are disabled, show:
				//if ((!dlg->m_Preview.GetCheck()) && (!dlg-
>m_Overlay.GetCheck()))
				//{
					dlg->m_Video.Show(dlg->m_Image);
			//	}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                }
        }
                //else
        //        Sleep(5);

                // leave time for other processes
                Sleep(5);
        //}

        // End Thread
        return(0);
}


void CSystemView::WORK2(void)
{
        // TODO: Add your control notification handler code here
        STOP2();

        if (m_Image != 0)
                return;

        if (!m_Video.Init())
        {
                // Error while trying to init control
                if (m_Capture.EnableWindow(FALSE))
                        flag1=TRUE;
                else flag1=FALSE;
                ClearRecord();
                m_strshow=(_T(""));
                UpdateData(FALSE);
                STOP2();

                MessageBox(m_Video.GetLastErrorString(), "VideoOCX Error",
MB_OK);
                return;                 // exit
        }

        // allocate memory according to grayscale check box
        //   Make sure you free the resources later using
VideoOCX::ReleaseImageHandle(long).

        m_Image = m_Video.GetColorImageHandle();

        // Set Preview and Overlay modes according to the current state of
the checkboxes

        if (!m_Video.SetPreview(TRUE))
        {
                // error while setting preview mode
                MessageBox(m_Video.GetLastErrorString(), "VideoOCX Error",
MB_OK);
        }

        if (m_Video.Start())                                    // start
capture mode
        {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            m_Running = TRUE;
            AfxBeginThread(CaptureThread, this);      // start capture thr}
    }
      else
      {
            MessageBox(m_Video.GetLastErrorString(), "VideoOCX Error",
MB_OK);
      }
      /*for ( int i=0;i<=8;i++)
      {
            ReadFile(hComp,&chBuff,1,&nReadPort,NULL);
            if (chBuff=='?')
                  break;
            sum[i]=chBuff;
      }*/


            OnBnClickedButton2();

            m_bFlag=FALSE;
            SAVE();
            OnBnClickedAddRecord();
      SendEmail1();
            WORK2();
}




afx_msg void CSystemView::OnCaptureVideosource(void)
{STOP2();
      if (!m_Video.Init())
      {
            // Error while trying to init control
            MessageBox(m_Video.GetLastErrorString(), "License Error",
MB_OK);
            return;                 // exit
      }
            if (m_Video.GetMode() == 0)
      {
            if (!m_Video.ShowSourceDlg())
            {
                  // Source dialog failed
                  //MessageBox(m_Video.GetLastErrorString(), "Capture
Error", MB_OK);
            }
            if (flag1==TRUE)
            {     flag1=FALSE;
                  OnBnClickedWork();
            }
       }
       else
       {
            // Only in VfW Mode
            //MessageBox("This feature is only vailable in VfW
Mode.","Capture Error", MB_OK + MB_ICONINFORMATION);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
	}
	// TODO: Add your command handler code here
}

void CSystemView::OnCaptureVideoformat139(void)
{// TODO: Add your command handler code here

if (!m_Video.Init())
	{
		// Error while trying to init control
		MessageBox(m_Video.GetLastErrorString(), "License Error",
MB_OK);
		return;			// exit
	}
	// check current mode
	if (m_Video.GetMode() == 0)
	{
		if (!m_Video.ShowFormatDlg())
		{
			// Format dialog failed
			MessageBox(m_Video.GetLastErrorString(), "VideoOCX
Error", MB_OK);
		}

		// Get adjusted image handle
		m_Video.ReleaseImageHandle(m_Image);

		/*if(m_Grayscale.GetCheck() > 0)
			m_Image = m_Video.GetGrayImageHandle();
		else
			m_Image = m_Video.GetColorImageHandle();*/
	}
	else
	{
		// Only in VfW Mode
		MessageBox("This feature is only available in VfW
Mode.","VideoOCX Sample Error", MB_OK + MB_ICONINFORMATION);

	}
	STOP2();
}

void CSystemView::SendEmail(void)
{
    CkMailMan mailman;

    // Any string passed to UnlockComponent begins the 30-day trial.
    bool unlocked = mailman.UnlockComponent("30-day trial");
    if (!unlocked)
      {
      printf("Failed to unlock component\n");
      return;
      }

    mailman.put_SmtpHost("mail.gsmadvance.com");

  //  Send an email with a file attachment.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    CkEmail email;
      strtel="Telephone Number:"+m_TelephoneMail;
      email.put_Body(strtel);

      email.AddTo("Preaw","019246624@mms.mobilelife.co.th");
      //email.AddTo("Dr.Pipat","pipat@telecom.kmitl.ac.th");
      //email.AddTo("Aor","aor_aun@yahoo.com");
      //email.AddTo("Dr.Pipat","pipatprommee@yahoo.com");
      //email.AddTo("Siripongse Jantree","jsiripongse@gmail.com");
      //email.AddTo("X","legendary_midfielder@hotmail.com");
      //email.AddTo("Neung","sumonmarth@hotmail.com");
      //email.AddTo("Preaw","017242970@mms.mobilelife.co.th");
      //email.AddTo("Siripongse Jantree","d069776775@mail.dtac.co.th");
       //email.AddTo("Test","webmaster@localhost");

    email.put_FromAddress("sumonmarth@mail.gsmadvance.com");

    email.put_Subject("Visitor Information");

//  Add a file attachment.
      if (!email.AddFileAttachment(sumbuffer,0))
      {
      email.SaveLastError("errors.xml");
      return;
      }

    // Send the email
    if (!mailman.SendEmail(&email))
      {
      mailman.SaveLastError("errors.xml");
      }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ChipCorder®
TECHNOLOGY BY ISD

ISD2560/75/90/120 Products

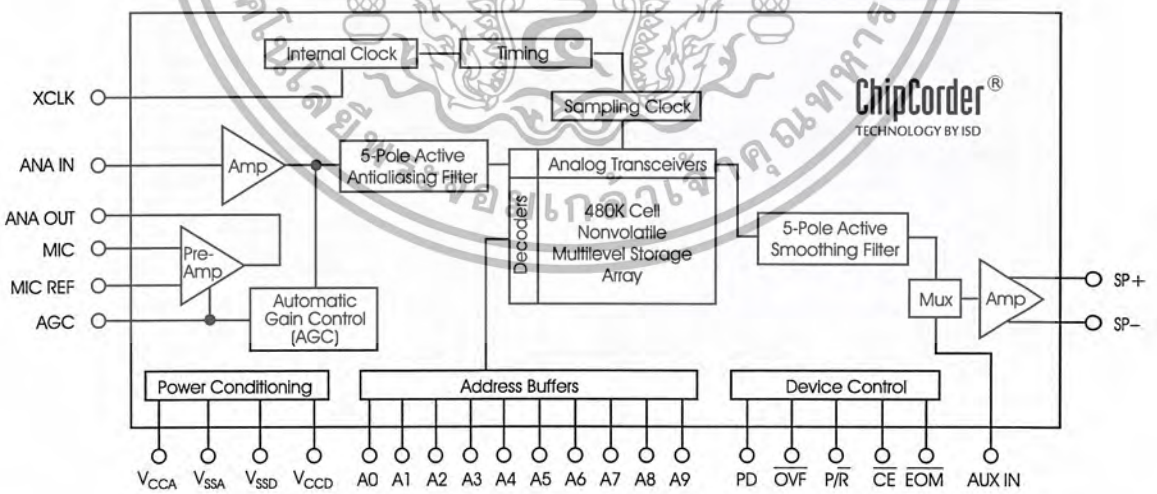
Single-Chip Voice Record/Playback Devices
60-, 75-, 90-, and 120-Second Durations

GENERAL DESCRIPTION

Information Storage Devices' ISD2500 ChipCorder® Series provides high-quality, single-chip record/playback solutions for 60- to 120-second messaging applications. The CMOS devices include an on-chip oscillator, microphone preamplifier, automatic gain control, antialiasing filter, smoothing filter, speaker amplifier, and high density multilevel storage array. In addition, the ISD2500 is microcontroller compatible, allowing complex messaging and addressing to be achieved.

Recordings are stored in on-chip nonvolatile memory cells, providing zero-power message storage. This unique, single-chip solution is made possible through ISD's patented multilevel storage technology. Voice and audio signals are stored directly into memory in their natural form, providing high-quality, solid-state voice reproduction.

Figure i: ISD2560/75/90/120 Device Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FEATURES

- Easy-to-use single-chip voice record/ playback solution
- High-quality, natural voice/audio reproduction
- Manual switch or microcontroller compatible playback can be edge- or level-activated
- Single-chip durations of 60, 75, 90, and 120 seconds
- Directly cascadable for longer durations
- Automatic Power-Down (Push-Button Mode)
  - Standby current 1 μA (typical)
- Zero-power message storage
  - Eliminates battery backup circuits
- Fully addressable to handle multiple messages
- 100-year message retention (typical)
- 100,000 record cycles (typical)
- On-chip clock source
- Programmer support for play-only applications
- Single +5 volt power supply
- Available in die form, DIP, SOIC, and TSOP packaging

Table i: ISD2560/75/90/120 Product Summary

Part Number	Duration (Seconds)	Input Sample Rate (KHz)	Typical Filter Pass Band (KHz)
ISD2560	60	8.0	3.4
ISD2575	75	6.4	2.7
ISD2590	90	5.3	2.3
ISD25120	120	4.0	1.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DETAILED DESCRIPTION

SPEECH/SOUND QUALITY

The ISD2500 series includes devices offered at 4.0, 5.3, 6.4, and 8.0 KHz sampling frequencies, allowing the user a choice of speech quality options. Increasing the duration within a product series decreases the sampling frequency and bandwidth, which affects sound quality. Please refer to the ISD2560/75/90/120 Product Summary table on page *ii* to compare filter pass band and product durations.

The speech samples are stored directly into on-chip nonvolatile memory without the digitization and compression associated with other solutions. Direct analog storage provides a very true, natural sounding reproduction of voice, music, tones, and sound effects not available with most solid-state digital solutions.

DURATION

To meet end system requirements, the ISD2500 series offers single-chip solutions at 60, 75, 90, and 120 seconds. Parts may also be cascaded together for longer durations.

EEPROM STORAGE

One of the benefits of ISD's ChipCorder technology is the use of on-chip nonvolatile memory, providing zero-power message storage. The message is retained for up to 100 years typically without power. In addition, the device can be re-recorded typically over 100,000 times.

MICROCONTROLLER INTERFACE

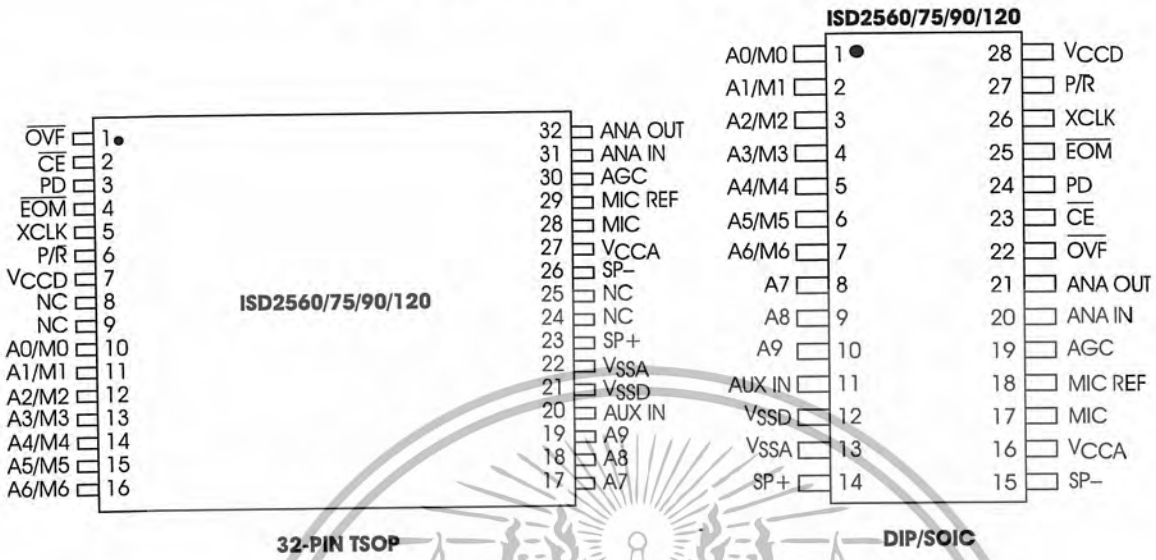
In addition to its simplicity and ease of use, the ISD2500 series includes all the interfaces necessary for microcontroller-driven applications. The address and control lines can be interfaced to a microcontroller and manipulated to perform a variety of tasks, including message assembly, message concatenation, predefined fixed message segmentation, and message management.

PROGRAMMING

The ISD2500 series is also ideal for playback-only applications, where single or multiple messages are referenced through buttons, switches, or a microcontroller. Once the desired message configuration is created, duplicates can easily be generated via an ISD programmer.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 1: ISD2560/75/90/120 Device Pinouts



PIN DESCRIPTIONS

VOLTAGE INPUTS (V_{CCA}, V_{CCD})

To minimize noise, the analog and digital circuits in the ISD2500 series devices use separate power busses. These voltage busses are brought out to separate pins and should be tied together as close to the supply as possible. In addition, these supplies should be decoupled as close to the package as possible.

GROUND INPUTS (V_{SSA}, V_{SSD})

The ISD2500 series of devices utilizes separate analog and digital ground busses. These pins should be connected separately through a low-impedance path to power supply ground.

POWER DOWN INPUT (PD)

When not recording or playing back, the PD pin should be pulled HIGH to place the part in a very low power mode (see I_{SB} specification). When overflow ($\overline{OVF}$) pulses LOW for an overflow condition, PD should be brought HIGH to reset the address pointer back to the beginning of the record/playback space. The PD pin has additional functionality in the M6 (Push-Button) Operational Mode described later in the Operational Mode section.

CHIP ENABLE INPUT ($\overline{CE}$)

The $\overline{CE}$ pin is taken LOW to enable all playback and record operations. The address inputs and playback/record input (P/$\overline{R}$) are latched by the falling edge of $\overline{CE}$. $\overline{CE}$ has additional functionality in the M6 (Push-Button) Operational Mode described later in the Operational Mode section.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PLAYBACK/RECORD INPUT (P/$\overline{R}$)

The P/$\overline{R}$ input is latched by the falling edge of the $\overline{CE}$ pin. A HIGH level selects a playback cycle while a LOW level selects a record cycle. For a record cycle, the address inputs provide the starting address and recording continues until PD or $\overline{CE}$ is pulled HIGH or an overflow is detected (i.e. the chip is full). When a record cycle is terminated by pulling PD or $\overline{CE}$ HIGH, an End-Of-Message ($\overline{EOM}$) marker is stored at the current address in memory. For a playback cycle, the address inputs provide the starting address and the device will play until an $\overline{EOM}$ marker is encountered. The device can continue past an $\overline{EOM}$ marker in an Operational Mode, or if $\overline{CE}$ is held LOW in address mode. (See page 5 for more Operational Modes).

END-OF-MESSAGE / RUN OUTPUT ($\overline{EOM}$)

A nonvolatile marker is automatically inserted at the end of each recorded message. It remains there until the message is recorded over. The $\overline{EOM}$ output pulses LOW for a period of T_{EOM} at the end of each message.

In addition, the ISD2500 series has an internal V_{CC} detect circuit to maintain message integrity should V_{CC} fall below 3.5 V. In this case, $\overline{EOM}$ goes LOW and the device is fixed in playback-only mode.

When the device is configured in Operational Mode M6 (Push-Button Mode), this pin provides an active-HIGH RUN signal, indicating the device is currently recording or playing. This signal can conveniently drive an LED for a visual indicator of a record or playback operation in process.

OVERFLOW OUTPUT ($\overline{OVF}$)

This signal pulses LOW at the end of memory space, indicating the device has been filled and the message has overflowed. The $\overline{OVF}$ output then follows the $\overline{CE}$ input until a PD pulse has reset the device. This pin can be used to cascade several ISD2500 devices together to increase record/playback durations.

MICROPHONE INPUT (MIC)

The microphone input transfers its signal to the on-chip preamplifier. An on-chip Automatic Gain Control (AGC) circuit controls the gain of this preamplifier from –15 to 24 dB. An external microphone should be AC coupled to this pin via a series capacitor. The capacitor value, together with the internal 10 KΩ resistance on this pin, determines the low-frequency cutoff for the ISD2500 series passband. See Application Information for additional information on low-frequency cutoff calculation.

MICROPHONE REFERENCE INPUT (MIC REF)

The MIC REF input is the inverting input to the microphone preamplifier. This provides a noise-canceling or common-mode rejection input to the device when connected to a differential microphone.

AUTOMATIC GAIN CONTROL INPUT (AGC)

The AGC dynamically adjusts the gain of the preamplifier to compensate for the wide range of microphone input levels. The AGC allows the full range of whispers to loud sounds to be recorded with minimal distortion. The "attack" time is determined by the time constant of a 5 KΩ internal resistance and an external capacitor (C2 on the schematic on page 18) connected from the AGC pin to V_{SSA} analog ground. The "release" time is determined by the time constant of an external resistor (R2) and an external capacitor (C2) connected in parallel between the AGC Pin and V_{SSA} analog ground. Nominal values of 470 KΩ and 4.7 μF give satisfactory results in most cases.

ANALOG OUTPUT (ANA OUT)

This pin provides the preamplifier output to the user. The voltage gain of the preamplifier is determined by the voltage level at the AGC pin.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ANALOG INPUT (ANA IN)

The analog input pin transfers its signal to the chip for recording. For microphone inputs, the ANA OUT pin should be connected via an external capacitor to the ANA IN pin. This capacitor value, together with the 3.0 KΩ input impedance of ANA IN, is selected to give additional cutoff at the low-frequency end of the voice passband. If the desired input is derived from a source other than a microphone, the signal can be fed, capacitively coupled, into the ANA IN pin directly.

EXTERNAL CLOCK INPUT (XCLK)

The external clock input for the ISD2500 devices has an internal pull-down device. These devices are configured at the factory with an internal sampling clock frequency centered to ±1 percent of specification. The frequency is then maintained to a variation of ±2.25 percent over the entire commercial temperature and operating voltage ranges. The internal clock has a ±5 percent tolerance over the industrial temperature and voltage range. A regulated power supply is recommended for industrial temperature range parts. If greater precision is required, the device can be clocked through the XCLK pin as follows:

Table 1: External Clock Sample Rates

Part Number	Sample Rate	Required Clock
ISD2560	8.0 KHz	1024 KHz
ISD2575	6.4 KHz	819.2 KHz
ISD2590	5.3 KHz	682.7 KHz
ISD25120	4.0 KHz	512 KHz

These recommended clock rates should not be varied because the antialiasing and smoothing filters are fixed, and aliasing problems can occur if the sample rate differs from the one recommended. The duty cycle on the input clock is not critical, as the clock is immediately divided by two. **If the XCLK is not used, this input must be connected to ground.**

SPEAKER OUTPUTS (SP+/SP–)

All devices in the ISD2500 series include an on-chip differential speaker driver, capable of driving 50 mW into 16 Ω from AUX IN (12.2 mW from memory).

The speaker outputs are held at V_{SSA} levels during record and power down. It is therefore not possible to parallel speaker outputs of multiple ISD2500 devices or the outputs of other speaker drivers.

NOTE *Connection of speaker outputs in parallel may cause damage to the device.*

A single output may be used alone (including a coupling capacitor between the SP pin and the speaker). These outputs may be used individually with the output signal taken from either pin. Using the differential outputs results in a 4 to 1 improvement in output power.

NOTE *Never ground or drive an unused speaker output.*

AUXILIARY INPUT (AUX IN)

The Auxiliary Input is multiplexed through to the output amplifier and speaker output pins when $\overline{CE}$ is HIGH, P/$\overline{R}$ is HIGH, and playback is currently not active or if the device is in playback overflow. When cascading multiple ISD2500 devices, the AUX IN pin is used to connect a playback signal from a following device to the previous output speaker drivers. For noise considerations, it is suggested that the auxiliary input not be driven when the storage array is active.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ADDRESS/MODE INPUTS (AX/MX)

The Address/Mode Inputs have two functions depending on the level of the two Most Significant Bits (MSB) of the address (A8 and A9).

If either or both of the two MSBs are LOW, the inputs are all interpreted as address bits and are used as the start address for the current record or playback cycle. The address pins are inputs only and do not output internal address information as the operation progresses. Address inputs are latched by the falling edge of $\overline{CE}$.

If both MSBs are HIGH, the Address/Mode Inputs are interpreted as Mode bits according to the Operational Mode table. There are six Operational Modes (M0..M6) available as indicated in the table. It is possible to use multiple Operational Modes simultaneously. Operational Modes are sampled on each falling edge of $\overline{CE}$, and thus Operational Modes and direct addressing are mutually exclusive.

OPERATIONAL MODES

The ISD2500 series is designed with several built-in Operational Modes that provide maximum functionality with minimum additional components. These are described in detail below. The Operational Modes use the address pins on the ISD2500 devices, but are mapped outside the valid address range. When the two Most Significant Bits (MSBs) are HIGH (A8 and A9), the remaining address signals are interpreted as mode bits and not as address bits. Therefore, Operational Modes and direct addressing are not compatible and cannot be used simultaneously.

There are two important considerations for using Operational Modes. First, all operations begin initially at address 0, which is the beginning of the ISD2500 address space. Later operations can begin at other address locations, depending on the Operational Mode(s) chosen. In addition, the address pointer is reset to 0 when the device is changed from record to playback, playback to record (except M6 mode), or when a Power-Down cycle is executed.

Second, Operational Modes are executed when $\overline{CE}$ goes LOW and the two MSBs are HIGH. This Operational Mode remains in effect until the next LOW-going $\overline{CE}$ signal, at which point the current address/mode levels are sampled and executed.

Table 2: Operational Modes Table

Mode Control	Function	Typical Use	Jointly Compatible[1]
M0	Message cueing	Fast-forward through messages	M4, M5, M6
M1	Delete $\overline{EOM}$ markers	Position $\overline{EOM}$ marker at the end of the last message	M3, M4, M5, M6
M2	Not applicable	Reserved	N/A
M3	Looping	Continuous playback from Address 0	M1, M5, M6
M4	Consecutive addressing	Record/play multiple consecutive messages	M0, M1, M5
M5	$\overline{CE}$ level-activated	Allows message pausing	M0, M1, M3, M4
M6	Push-button control	Simplified device interface	M0, M1, M3

1. Additional Operational Modes can be used simultaneously with the given mode.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OPERATIONAL MODES DESCRIPTION

The Operational Modes can be used in conjunction with a microcontroller, or they can be hard-wired to provide the desired system operation.

M0 — MESSAGE CUEING

Message Cueing allows the user to skip through messages, without knowing the actual physical addresses of each message. Each $\overline{CE}$ LOW pulse causes the internal address pointer to skip to the next message. This mode should be used for playback only, and is typically used with the M4 Operational Mode.

M1 — DELETE $\overline{EOM}$ MARKERS

The M1 Operational Mode allows sequentially recorded messages to be combined into a single message with only one $\overline{EOM}$ marker set at the end of the final message. When this Operational Mode is configured, messages recorded sequentially are played back as one continuous message.

M2 — UNUSED

When Operational Modes are selected, the M2 pin should be LOW.

M3 — MESSAGE LOOPING

The M3 Operational Mode allows for the automatic, continuously repeated playback of the message located at the beginning of the address space. A message can completely fill the ISD2500 device and will loop from beginning to end without $\overline{OVF}$ going LOW.

M4 — CONSECUTIVE ADDRESSING

During normal operations, the address pointer will reset when a message is played through to an $\overline{EOM}$ marker. The M4 Operational Mode inhibits the address pointer reset on $\overline{EOM}$, allowing messages to be played back consecutively.

M5 — $\overline{CE}$-LEVEL ACTIVATED

The default mode for ISD2500 devices is for $\overline{CE}$ to be edge-activated on playback and level-activated on record. The M5 Operational Mode causes the $\overline{CE}$ pin to be interpreted as level-activated as opposed to edge-activated during playback. This is specifically useful for terminating playback operations using the $\overline{CE}$ signal.

In this mode, $\overline{CE}$ LOW begins a playback cycle, at the beginning of the device memory. The playback cycle continues as long as $\overline{CE}$ is held LOW. When $\overline{CE}$ goes HIGH, playback will immediately end. A new $\overline{CE}$ LOW will restart the message from the beginning unless M4 is also HIGH.

M6 — PUSH-BUTTON MODE

The ISD2500 series of devices contain a Push-Button Operational Mode. The Push-Button mode is used primarily in very low-cost applications and is designed to minimize external circuitry and components, thereby reducing system cost. In order to configure the device in Push-Button Operational Mode, the two most significant address bits must be HIGH, and the M6 mode pin must also be HIGH. A device in this mode always powers down at the end of each playback or record cycle after $\overline{CE}$ goes HIGH.

When this Operational Mode is implemented, several of the pins on the device have alternate functionality:

Table 3: Alternate Functionality in Pins

Pin Name	Alternate Functionality in Push-Button Mode
$\overline{CE}$	Start/Pause Push-Button (LOW pulse-activated)
PD	Stop/Reset Push-Button (HIGH pulse activated)
$\overline{EOM}$	Active-HIGH Run Indicator

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$\overline{\text{CE}}$ PIN (START/PAUSE)

In Push-Button Operational Mode, $\overline{\text{CE}}$ acts as a LOW-going pulse-activated START/PAUSE signal. If no operation is currently in progress, a LOW-going pulse on this signal will initiate a playback or a record cycle according to the level on the P/$\overline{\text{R}}$ pin. A subsequent pulse on the $\overline{\text{CE}}$ pin, before an End-Of-Message is reached in playback or an overflow condition occurs, will cause the device to pause. The address counter is not reset, and another $\overline{\text{CE}}$ pulse will cause the device to continue the operation from the place where it was paused.

PD PIN (STOP/RESET)

In push-button Operational Mode, PD acts as a HIGH-going pulse-activated STOP/RESET signal. When a playback or record cycle is in progress and a HIGH-going pulse is observed on PD, the current cycle is terminated and the address pointer is reset to address 0, the beginning of the message space.

$\overline{\text{EOM}}$ PIN (RUN)

In Push-Button Operational Mode, $\overline{\text{EOM}}$ becomes an active-HIGH RUN signal which can be used to drive an LED or other external device. It is HIGH whenever a record or playback operation is in progress.

Recording in Push-Button Mode

1. The PD pin should be LOW, usually using a pull-down resistor.
2. The P/$\overline{\text{R}}$ pin is taken LOW.
3. The $\overline{\text{CE}}$ pin is pulsed LOW. Recording starts, $\overline{\text{EOM}}$ goes HIGH to indicate an operation in progress.
4. The $\overline{\text{CE}}$ pin is pulsed LOW. Recording pauses, $\overline{\text{EOM}}$ goes back LOW. The internal address pointers are not cleared, but an EOM marker is stored in memory to point to the message end. The P/$\overline{\text{R}}$ pin may be taken HIGH at this time. Any subsequent $\overline{\text{CE}}$ would start a playback at address 0.
5. The $\overline{\text{CE}}$ pin is pulsed LOW. Recording starts at the next address after the previous set EOM marker. $\overline{\text{EOM}}$ goes back HIGH.

NOTE *If the M1 Operational Mode pin is also HIGH, the just previously written EOM bit is erased, and recording starts at that address.)*

---

6. When the recording sequences are finished, the final $\overline{\text{CE}}$ pulse LOW will end the last record cycle, leaving a set $\overline{\text{EOM}}$ marker at the message end. Recording may also be terminated by a HIGH level on PD, which will leave a set EOM marker.

Playback in Push-Button Mode

1. The PD pin should be LOW.
2. The P/$\overline{\text{R}}$ pin is taken HIGH.
3. The $\overline{\text{CE}}$ pin is pulsed LOW. Playback starts, $\overline{\text{EOM}}$ goes HIGH to indicate an operation in progress.
4. If the $\overline{\text{CE}}$ pin is pulsed LOW or an EOM marker is encountered during an operation, the part will pause. The internal address pointers are not cleared, and $\overline{\text{EOM}}$ goes back LOW. The P/$\overline{\text{R}}$ pin may be changed at this time. A subsequent record operation would not reset the address pointers and the recording would begin where playback ended.
5. $\overline{\text{CE}}$ is again pulsed LOW. Playback starts where it left off, with $\overline{\text{EOM}}$ going HIGH to indicate an operation in progress.
6. Playback continues as in steps 4 and 5 until PD is pulsed HIGH or overflow occurs.
7. If in overflow, pulling $\overline{\text{CE}}$ LOW will reset the address pointer and start playback from the beginning. After a PD pulse, the part is reset to address 0.

NOTE *Push-button mode can be used in conjunction with modes M0, M1, and M3.*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GOOD AUDIO DESIGN PRACTICES

ISD products are very high-quality single-chip voice recording and playback systems. To ensure the highest quality voice reproduction, it is important that good audio design practices on layout and power supply decoupling be followed. See the ISD Application Notes in this book for details.

ISD1000A COMPATIBILITY

The ISD2500 series of devices is designed to provide upward compatibility with the ISD1000A family. When designing with the ISD2500 series, the following differences should be noted.

ADDRESSING

The ISD2560/75/90/120 devices have 480K storage cells designed to provide 60 seconds of storage at a sampling rate of 8.0 KHz. This is approximately four times the storage of the ISD1000A family. To enable the same addressing resolution, two additional address pins have been added. The address space of each device is divisible into 600 increments with valid addressing from 00 to 257 Hex. Some higher addresses are mapped into the Operational Modes. All other addresses are invalid.

OVERFLOW

The ISD1000A series combined two functions on the $\overline{EOM}$ pin: end-of-message indication and overflow. The ISD2500 separates these two functions. Pin 25 (PDIP package) remains as $\overline{EOM}$, but outputs only the EOM signal indication. Pin 22 (PDIP package) becomes $\overline{OVF}$ and pulses LOW only when the device reaches its end of memory, or is "full." This change allows easy message cueing and addressability across device boundaries. This also means that the M2 Operational Mode found in the ISD1000A family is not implemented in the ISD2500 series.

PUSH-BUTTON MODE

The ISD2500 series includes an additional Operational Mode called Push-Button mode. This provides an alternative interface to the record and playback functions of the part. The $\overline{CE}$ and PD pins become redefined as edge-activated "push-buttons." A pulse on $\overline{CE}$ initiates a cycle, and if triggered again, pauses the current cycle without resetting the address pointer (i.e., a Start or Pause function). PD stops any current cycle and resets the address pointer to the beginning of the message space (i.e., a Stop and Reset function). Additionally, the $\overline{EOM}$ pin functions as an active-HIGH run indicator, and can be used to drive an LED indicating a record or playback operation is in progress. Devices in the Push-Button mode cannot be cascaded.

LOOPING MODE

The ISD2500 series can loop with a message that completely fills the memory space.

NOTE *Additional descriptions of ISD2500 device functionality and application examples are provided in the ISD Application Notes in this book.*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TIMING DIAGRAMS

Figure 2: Record

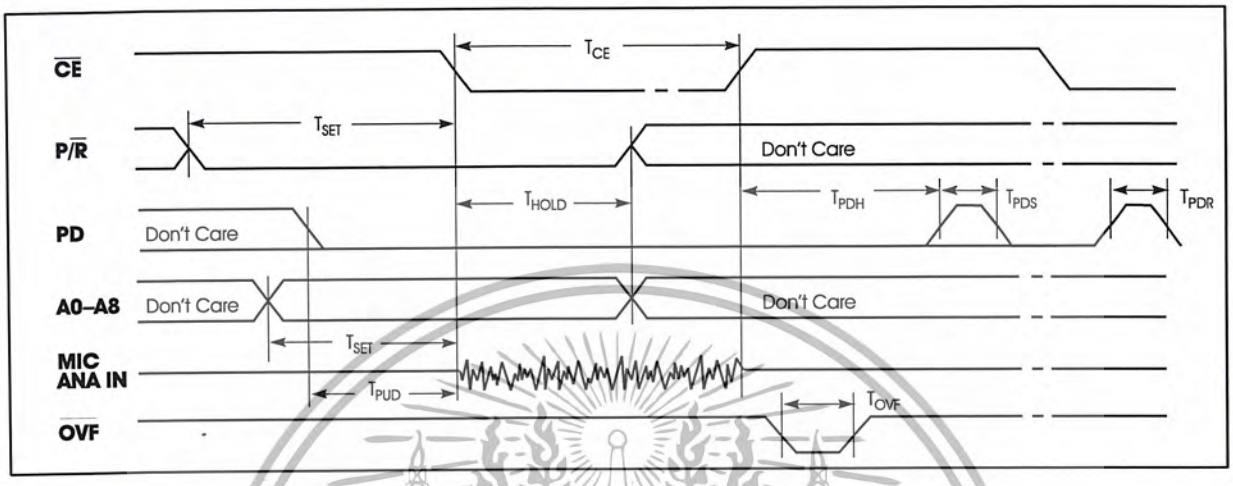
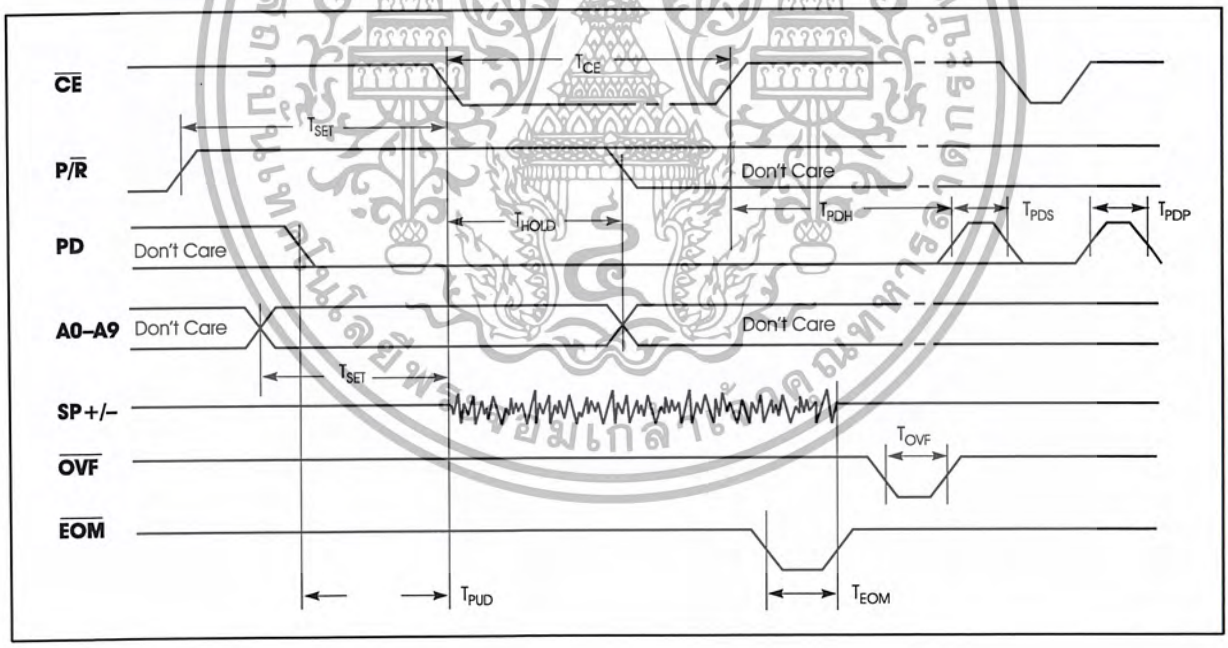


Figure 3: Playback



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 4: Absolute Maximum Ratings (Packaged Parts)[1]

Condition	Value
Junction temperature	150°C
Storage temperature range	–65°C to +150°C
Voltage applied to any pin	(V_{SS} – 0.3 V) to (V_{CC} + 0.3 V)
Voltage applied to any pin (Input current limited to ±20 mA)	(V_{SS} – 1.0 V) to (V_{CC} + 1.0 V)
Lead temperature (soldering – 10 seconds)	300°C
$V_{CC} - V_{SS}$	–0.3 V to +7.0 V

1. *Stresses above those listed may cause permanent damage to the device. Exposure to the absolute maximum ratings may affect device reliability. Functional operation is not implied at these conditions.*

Table 5: Operating Conditions (Packaged Parts)

Condition	Value
Commercial operating temperature range[1]	0°C to +70°C
Industrial operating temperature range[1]	
Supply voltage (V_{CC})[2]	+4.5 V to +5.5 V
Ground voltage (V_{SS})[3]	0 V

1. *Case temperature.*
2. *$V_{CC} = V_{CCA} = V_{CCD}$.*
3. *$V_{SS} = V_{SSA} = V_{SSD}$.*
4. *Consult factory.*

Table 6: DC Parameters (Packaged Parts)

Symbol	Parameters	Min[2]	Typ[1]	Max[2]	Units	Conditions
V_{IL}	Input Low Voltage			0.8	V	
V_{IH}	Input High Voltage	2.0			V	
V_{OL}	Output Low Voltage			0.4	V	I_{OL} = 4.0 mA
V_{OH}	Output High Voltage	V_{CC}–0.4			V	I_{OH} = –10 µA
V_{OH1}	$\overline{OVF}$ Output High Voltage	2.4			V	I_{OH} = –1.6 mA
V_{OH2}	$\overline{EOM}$ Output High Voltage	V_{CC} – 1.0	V_{CC} – 0.8		V	I_{OH} = –3.2 mA
I_{CC}	V_{CC} Current (Operating)		25	30	mA	$R_{EXT} = \infty$ [3]
I_{SB}	V_{CC} Current (Standby)		1	10	µA	[3]
I_{IL}	Input Leakage Current			±1	µA	
I_{ILPD}	Input Current HIGH w/Pull Down			130	µA	Force V_{CC} [4]
R_{EXT}	Output Load Impedance	16			Ω	Speaker Load
R_{MIC}	Preamp In Input Resistance	4	9	15	KΩ	MIC and MIC REF Pins
R_{AUX}	AUX INPUT Resistance	5	11	20	KΩ	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 6: DC Parameters (Packaged Parts)

Symbol	Parameters	Min(2)	Typ (1)	Max(2)	Units	Conditions
$R_{ANA\ IN}$	ANA IN Input Resistance	2.3	3	5	KΩ	
A_{PRE1}	Preamp Gain 1	21	24	26	dB	AGC = 0.0 V
A_{PRE2}	Preamp Gain 2		–15	5	dB	AGC = 2.5 V
A_{AUX}	AUX IN/SP+ Gain		0.98	1.0	V/V	
A_{ARP}	ANA IN to SP+/– Gain	21	23	26	dB	
R_{AGC}	AGC Output Resistance	2.5	5	9.5	KΩ	

1. *Typical values @ T_A = 25°C and 5.0 V.*
2. *All Min/Max limits are guaranteed by ISD via electrical testing or characterization. Not all specifications are 100 percent tested.*
3. *V_{CCA} and V_{CCD} connected together.*
4. *XCLK pin only.*

Table 7: AC Parameters (Packaged Parts)

Symbol	Characteristic		Min(2)	Typ (1)	Max(2)	Units	Conditions
F_S	Sampling Frequency	ISD2560		8.0		KHz	(7)
		ISD2575		6.4		KHz	(7)
		ISD2590		5.3		KHz	(7)
		ISD25120		4.0		KHz	(7)
F_{CF}	Filter Pass Band	ISD2560		3.4		KHz	3 dB Roll-Off Point (3) (8)
		ISD2575		2.7		KHz	3 dB Roll-Off Point (3) (8)
		ISD2590		2.3		KHz	3 dB Roll-Off Point (3) (8)
		ISD25120		1.7		KHz	3 dB Roll-Off Point (3) (8)
T_{REC}	Record Duration	ISD2560	58.1	60.0	62.0	sec	Commercial Operation(7)
		ISD2560	56.5	60.0	63.8	sec	Industrial Operation(7)
		ISD2575	72.6	75.0	77.5	sec	Commercial Operation(7)
		ISD2575	70.7	75.0	79.7	sec	Industrial Operation(7)
		ISD2590	87.1	90.0	93.0	sec	Commercial Operation(7)
		ISD25120	116.1	120.0	123.9	sec	Commercial Operation(7)
T_{PLAY}	Playback Duration	ISD2560	58.1	60.0	62.0	sec	Commercial Operation
		ISD2560	56.5	60.0	63.8	sec	Industrial Operation
		ISD2575	72.6	75.0	77.5	sec	Commercial Operation
		ISD2575	70.7	75.0	79.7	sec	Industrial Operation
		ISD2590	87.1	90.0	93.0	sec	Commercial Operation
		ISD25120	116.1	120.0	123.9	sec	Commercial Operation
T_{CE}	$\overline{CE}$ Pulse Width			100		nsec	
T_{SET}	Control/Address Setup Time			300		nsec	
T_{HOLD}	Control/Address Hold Time			0		nsec	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 7: AC Parameters (Packaged Parts)

Symbol	Characteristic		Min(2)	Typ (1)	Max(2)	Units	Conditions
T_{PUD}	Power-Up Delay	ISD2560	24.1	25.0	27.8	msec	Commercial Operation
		ISD2560	23.5		28.5	msec	Industrial Operation
		ISD2575	30.2	31.3	34.3	msec	Commercial Operation
		ISD2575	29.3	31.3	35.2	msec	Industrial Operation
		ISD2590	36.2	37.5	40.8	msec	Commercial Operation
		ISD25120	48.2	50.0	53.6	msec	Commercial Operation
T_{PDR}	PD Pulse Width Record	ISD2560		25		msec	
		ISD2575		31.25		msec	
		ISD2590		37.5		msec	
		ISD25120		50.0		msec	
T_{PDP}	PD Pulse Width Play	ISD2560		12.5		msec	
		ISD2575		15.625		msec	
		ISD2590		18.75		msec	
		ISD25120		25.0		msec	
T_{PDS}	PD Pulse Width Static			100		nsec	(6)
T_{PDH}	Power Down Hold			0		nsec	
T_{EOM}	$\overline{EOM}$ Pulse Width	ISD2560		12.5		msec	
		ISD2575		15.625		msec	
		ISD2590		18.75		msec	
		ISD25120		25.0		msec	
T_{OVF}	Overflow Pulse Width			6.5		µsec	
THD	Total Harmonic Distortion			1	2	%	@ 1 KHz
P_{OUT}	Speaker Output Power			12.2	50	mW	R_{EXT} = 16 Ω (4)
V_{OUT}	Voltage Across Speaker Pins				2.5	V p-p	R_{EXT} = 600 Ω
V_{IN1}	MIC Input Voltage				20	mV	Peak-to-Peak (5)
V_{IN2}	ANA IN Input Voltage				50	mV	Peak-to-Peak
V_{IN3}	Aux Input Voltage				1.25	V	Peak-to-Peak; R_{EXT} = 16 Ω

1. *Typical values @ T_A = 25°C and 5.0 V.*
2. *All Min/Max limits are guaranteed by ISD via electrical testing or characterization. Not all specifications are 100 percent tested.*
3. *Low-frequency cutoff depends upon the value of external capacitors (see Pin Descriptions).*
4. *From AUX IN; if ANA IN is driven at 50 mV p-p, the P_{OUT} = 12.2 mW, typical.*
5. *With 5.1 KΩ series resistor at ANA IN.*
6. *T_{PDS} is required during a static condition, typically overflow.*
7. *Sampling Frequency and playback Duration can vary as much as ±2.25 percent over the commercial temperature range and voltage range and ±5 percent over the industrial temperature and voltage range. For greater stability, an external clock can be utilized (see Pin Descriptions).*
8. *Filter specification applies to both the antialiasing filter and the smoothing filter. Therefore, from input to output, expect a 6 dB drop by nature of passing through both filters.*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TYPICAL PARAMETER VARIATION WITH VOLTAGE AND TEMPERATURE (PACKAGED PARTS)

Chart 1: Record Mode Operating Current (I_{CC})

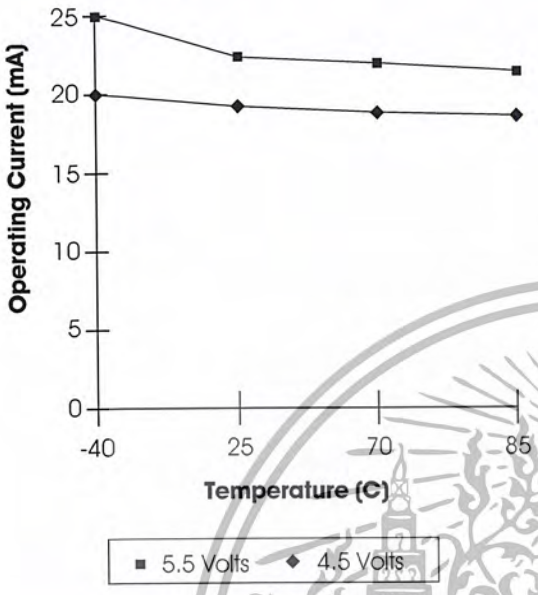


Chart 3: Standby Current (I_{SB})

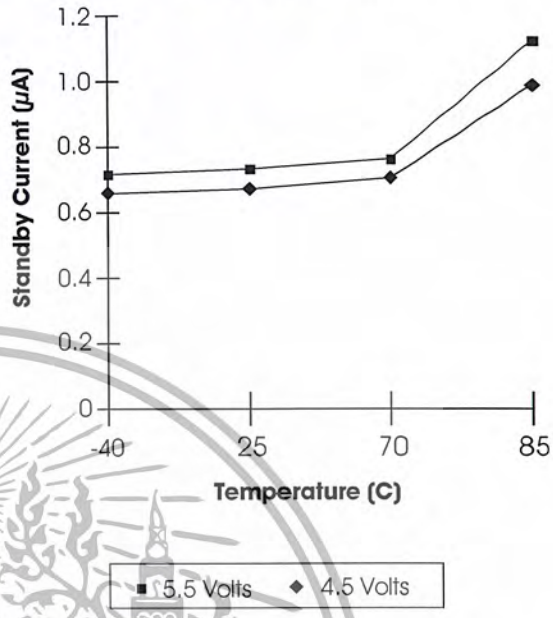


Chart 2: Total Harmonic Distortion

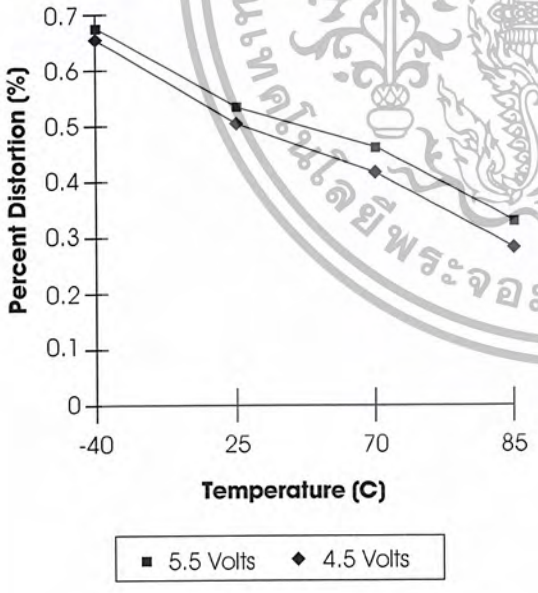
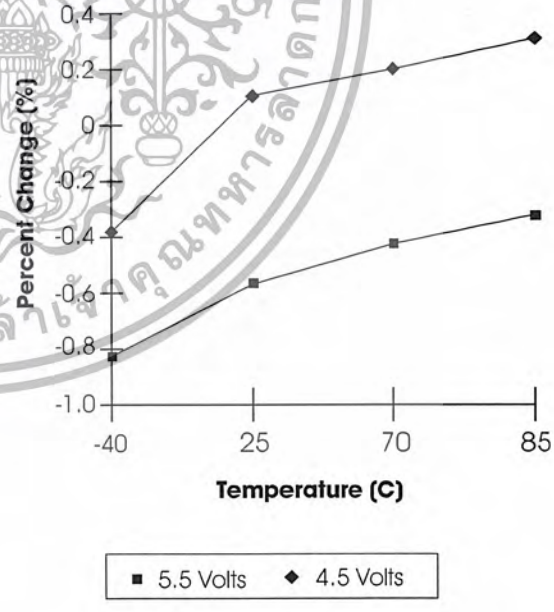


Chart 4: Oscillator Stability



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 8: Absolute Maximum Ratings (Die)[1]

Condition	Value
Junction temperature	150°C
Storage temperature range	–65°C to +150°C
Voltage applied to any pad	(V_{SS} – 0.3 V) to (V_{CC} + 0.3 V)
Voltage applied to any pad (Input current limited to ±20 mA)	(V_{SS} –1.0 V) to (V_{CC} + 1.0 V)
$V_{CC} - V_{SS}$	–0.3 V to +7.0 V

1. *Stresses above those listed may cause permanent damage to the device. Exposure to the absolute maximum ratings may affect device reliability. Functional operation is not implied at these conditions.*

Table 9: Operating Conditions (Die)

Condition	Value
Commercial operating temperature range	0°C to +50°C
Supply voltage (V_{CC})[1]	+4.5 V to +6.5 V
Ground voltage (V_{SS})[2]	0 V

1. $V_{CC} = V_{CCA} = V_{CCD}$.
2. $V_{SS} = V_{SSA} = V_{SSD}$.

Table 10: DC Parameters (Die)

Symbol	Parameters	Min[2]	Typ[1]	Max[2]	Units	Conditions
V_{IL}	Input Low Voltage			0.8	V	
V_{IH}	Input High Voltage	2.0			V	
V_{OL}	Output Low Voltage			0.4	V	I_{OL} = 4.0 mA
V_{OH}	Output High Voltage	V_{CC} – 0.4			V	I_{OH} = –10 µA
V_{OH1}	$\overline{OVF}$ Output High Voltage	2.4			V	I_{OH} = –1.6 mA
V_{OH2}	$\overline{EOM}$ Output High Voltage	V_{CC} – 1.0	V_{CC} – 0.8		V	I_{OH} = –3.2 mA
I_{CC}	V_{CC} Current (Operating)		25	30	mA	$R_{EXT} = \infty$ [3]
I_{SB}	V_{CC} Current (Standby)		1	10	µA	[2]
I_{IL}	Input Leakage Current			±1	µA	
I_{ILPD}	Input Current HIGH with Pull Down			130	µA	Force V_{CC} [4]
R_{EXT}	Output Load Impedance	16			Ω	Speaker Load
R_{MIC}	Preamp In Input Resistance	4	9	15	KΩ	MIC and MIC REF Pads
R_{AUX}	AUX INput Resistance	5	11	20	KΩ	
$R_{ANA\ IN}$	ANA IN Input Resistance	2.3	3	5	KΩ	
A_{PRE1}	Preamp Gain 1	21	24	26	dB	AGC = 0.0 V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 10: DC Parameters (Die)

Symbol	Parameters	Min(2)	Typ (1)	Max(2)	Units	Conditions
A_{PRE2}	Preamp Gain 2		–15	5	dB	AGC = 2.5 V
A_{AUX}	AUX IN/SP+ Gain		0.98	1.0	V/V	
A_{ARP}	ANA IN to SP+/- Gain	21	23	26	dB	
R_{AGC}	AGC Output Resistance	2.5	5	9.5	KΩ	

1. *Typical values @ T_A = 25°C and 5.0 V.*
2. *All Min/Max limits are guaranteed by ISD via electrical testing or characterization. Not all specifications are 100 percent tested.*
3. *V_{CCA} and V_{CCD} connected together.*
4. *XCLK pad only.*

Table 11: AC Parameters (Die)

Symbol	Characteristic		Min(2)	Typ (1)	Max(2)	Units	Conditions
F_S	Sampling Frequency	ISD2560		8.0		KHz	(7)
		ISD2575		6.4		KHz	(7)
		ISD2590		5.3		KHz	(7)
		ISD25120		4.0		KHz	(7)
F_{CF}	Filter Pass Band	ISD2560		3.4		KHz	3 dB Roll-Off Point (3) (8)
		ISD2575		2.7		KHz	3 dB Roll-Off Point (3) (8)
		ISD2590		2.3		KHz	3 dB Roll-Off Point (3) (8)
		ISD25120		1.7		KHz	3 dB Roll-Off Point (3) (8)
T_{REC}	Record Duration	ISD2560	58.1	60.0	62.0	sec	Commercial Operation(7)
		ISD2575	72.6	75.0	77.5	sec	Commercial Operation(7)
		ISD2590	87.1	90.0	93.0	sec	Commercial Operation(7)
		ISD25120	116.1	120.0	123.9	sec	Commercial Operation(7)
T_{PLAY}	Playback Duration	ISD2560	58.1	60.0	62.0	sec	Commercial Operation(7)
		ISD2575	72.6	75.0	77.5	sec	Commercial Operation(7)
		ISD2590	87.1	90.0	93.0	sec	Commercial Operation(7)
		ISD25120	116.1	120.0	123.9	sec	Commercial Operation(7)
T_{CE}	$\overline{CE}$ Pulse Width			100		nsec	
T_{SET}	Control/Address Setup Time			300		nsec	
T_{HOLD}	Control/Address Hold Time			0		nsec	
T_{PUD}	Power-Up Delay	ISD2560	24.1	25.0	27.8	msec	Commercial Operation
		ISD2575	30.2	31.3	34.3	msec	Commercial Operation
		ISD2590	36.2	37.5	40.8	msec	Commercial Operation
		ISD25120	48.2	50.0	53.6	msec	Commercial Operation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 11: AC Parameters (Die)

Symbol	Characteristic		Min(2)	Typ (1)	Max(2)	Units	Conditions
T_{PDR}	PD Pulse Width Record	ISD2560		25		msec	
		ISD2575		31.25		msec	
		ISD2590		37.5		msec	
		ISD25120		50.0		msec	
T_{PDP}	PD Pulse Width Play	ISD2560		12.5		msec	
		ISD2575		15.625		msec	
		ISD2590		18.75		msec	
		ISD25120		25.0		msec	
T_{PDS}	PD Pulse Width Static			100		nsec	(6)
T_{PDH}	Power Down Hold			0		nsec	
T_{EOM}	$\overline{EOM}$ Pulse Width	ISD2560		12.5		msec	
		ISD2575		15.625		msec	
		ISD2590		18.75		msec	
		ISD25120		25.0		msec	
T_{OVF}	Overflow Pulse Width			6.5		µsec	
THD	Total Harmonic Distortion			1	3	%	@ 1 KHz
P_{OUT}	Speaker Output Power			12.2	50	mW	R_{EXT} = 16 Ω(4)
V_{OUT}	Voltage Across Speaker Pins				2.5	V p-p	R_{EXT} = 600 Ω
V_{IN1}	MIC Input Voltage				20	mV	Peak-to-Peak (5)
V_{IN2}	ANA IN Input Voltage				50	mV	Peak-to-Peak
V_{IN3}	Aux Input Voltage				1.25	V	Peak-to-Peak; R_{EXT} = 16 Ω

1. *Typical values @ T_A = 25°C and 5.0 V.*
2. *All Min/Max limits are guaranteed by ISD via electrical testing or characterization. Not all specifications are 100 percent tested.*
3. *Low-frequency cutoff depends upon the value of external capacitors (see Pin Descriptions).*
4. *From AUX IN; if ANA IN is driven at 50 mV p-p, the P_{OUT} = 12.2 mW, typical.*
5. *With 5.1 KΩ series resistor at ANA IN.*
6. *T_{PDS} is required during a static condition, typically overflow.*
7. *Sampling Frequency and playback Duration can vary as much as ±2.25 percent over the commercial temperature range and voltage range. For greater stability, an external clock can be utilized (see Pin Descriptions).*
8. *Filter specification applies to the antialiasing filter and the smoothing filter.*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TYPICAL PARAMETER VARIATION WITH VOLTAGE AND TEMPERATURE (DIE)

Chart 5: Record Mode Operating Current (I_{CC})

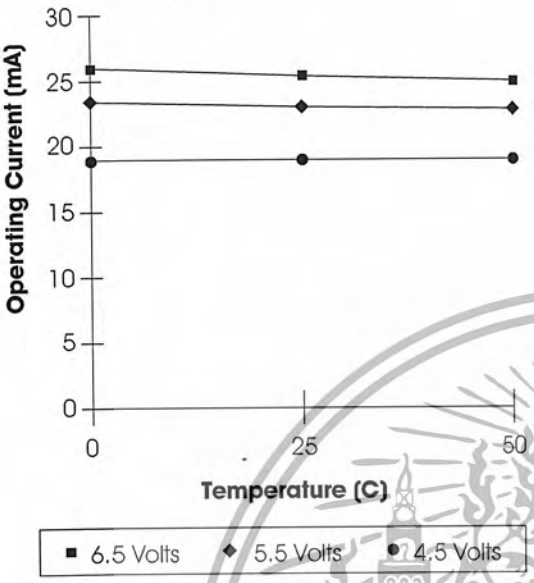


Chart 7: Standby Current (I_{SB})

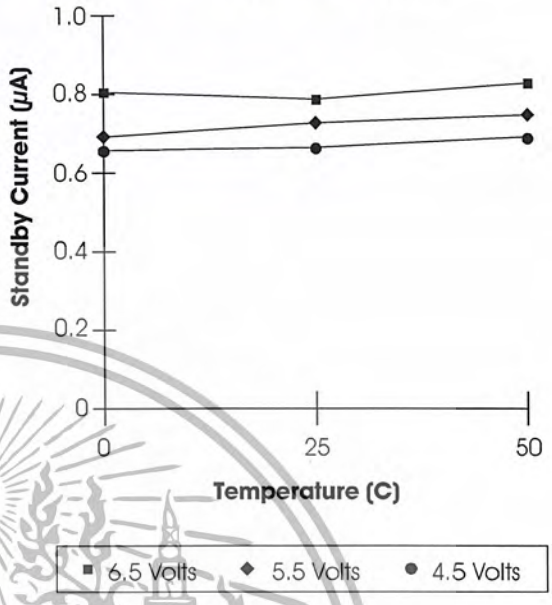


Chart 6: Total Harmonic Distortion

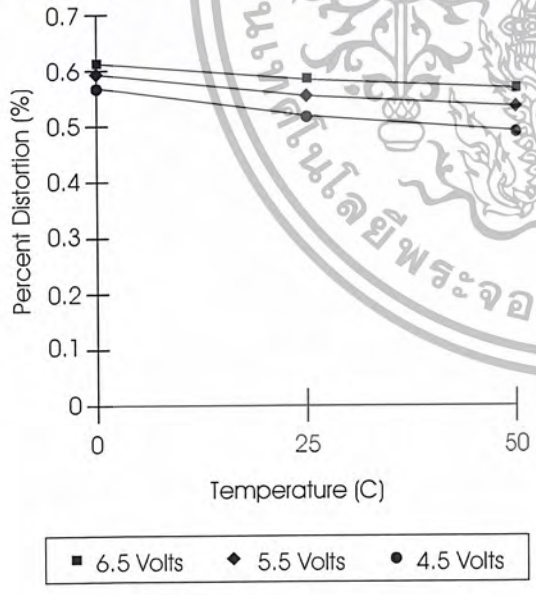
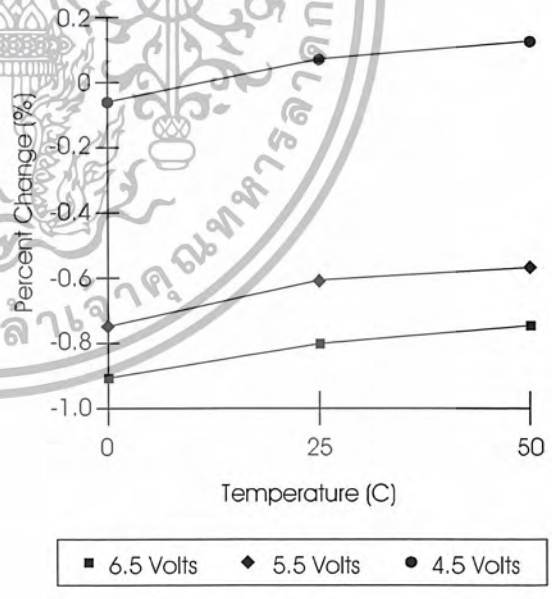
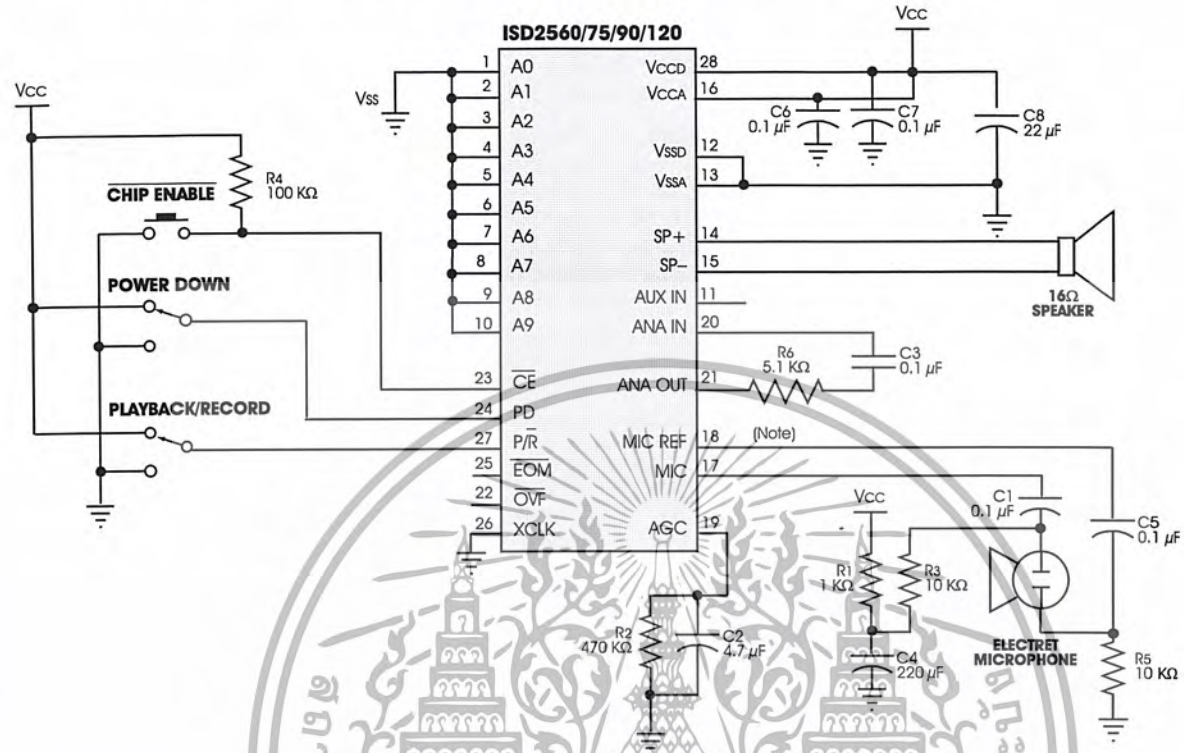


Chart 8: Oscillator Stability



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 4: ISD2560/75/90/120 Application Example—Design Schematic



NOTE: *If desired, pin 18 (PDIP package) may be left unconnected (microphone preamplifier noise will be higher). In this case, pin 18 must not be tied to any other signal or voltage. Additional design example schematics are provided in the Application Notes in this book.*

Table 12: Application Example—Basic Device Control

Control Step	Function	Action
1	Power up chip and select record/playback mode	(1.) PD = LOW, (2.) P/$\overline{R}$ = As desired
2	Set message address for record/playback	Set addresses A0–A9
3A	Begin playback	P/$\overline{R}$ = HIGH, $\overline{CE}$ = Pulsed LOW
3B	Begin record	P/$\overline{R}$ = LOW, $\overline{CE}$ = LOW
4A 4B	End playback End record	Automatic PD or $\overline{CE}$ = HIGH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 13: Application Example—Passive Component Functions

Part	Function	Comments
R1	Microphone power supply decoupling	Reduces power supply noise
R2	Release time constant	Sets release time for AGC
R3, R5	Microphone biasing resistors	Provides biasing for microphone operation
R4	Series limiting resistor	Reduces level to prevent distortion at higher supply voltages.
R6	Series limiting resistor	Reduces level to high supply voltages
C1, C5	Microphone DC–blocking capacitor Low-frequency cutoff	Decouples microphone bias from chip. Provides single-pole low-frequency cutoff and common mode noise rejection.
C2	Attack/Release time constant	Sets attack/release time for AGC
C3	Low-frequency cutoff capacitor	Provides additional pole for low-frequency cutoff
C4	Microphone power supply decoupling	Reduces power supply noise
C6, C7, C8	Power supply capacitors	Filter and bypass of power supply

EXPLANATION

In this simplified block diagram of a microcontroller application, the Push-Button mode and message cueing are used. The microcontroller is a 16-pin version with enough port pins for buttons, an LED, and the ISD2500 series device. The software can be written to use three buttons: one each for play and record, and one for message selection. Because the microcontroller is interpreting the buttons and commanding the ISD2500 device, software can be written for any functions desired in a particular application.

NOTE *ISD does not recommend connecting address lines directly to a microprocessor bus. Address lines should be externally latched.*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 5: ISD2560/75/90/120 Application Example—Microcontroller/ISD2500 Interface

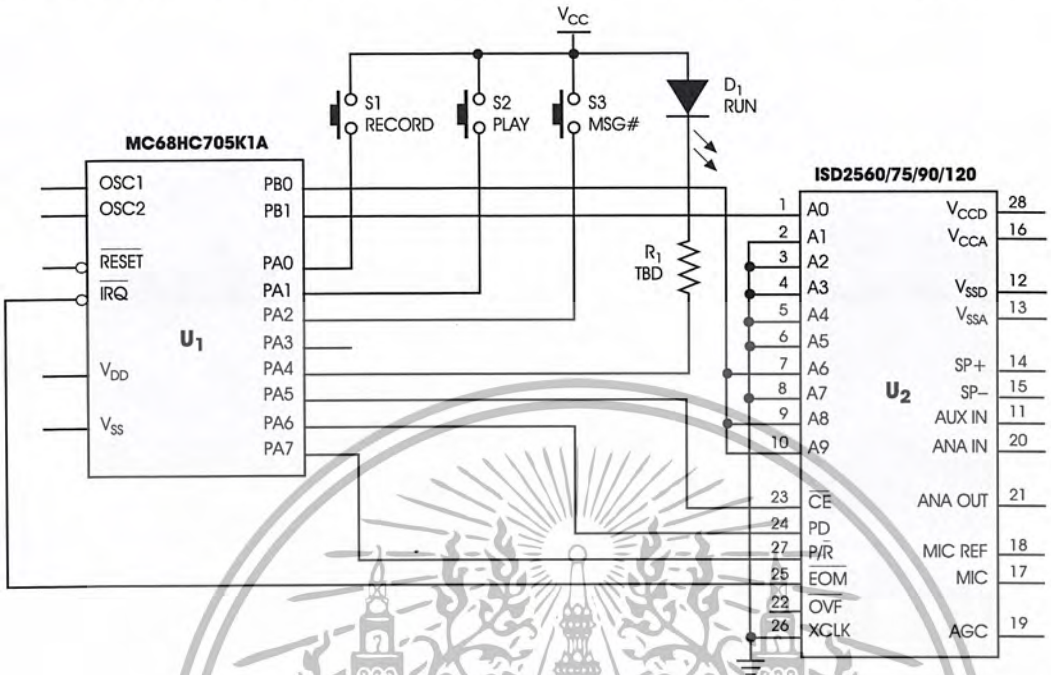
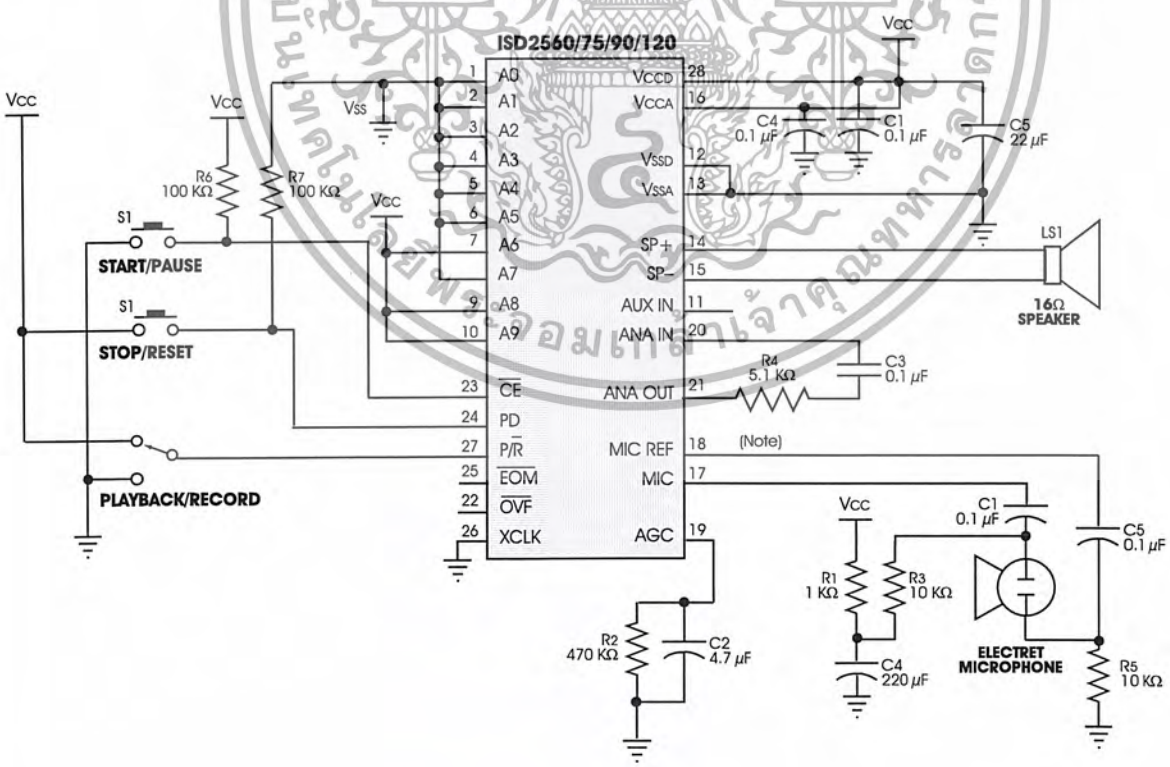


Figure 6: ISD2500 Application Example—Push-Button



NOTE: *Please refer to Application Information.*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 14: Application Example—Push-Button Control

Control Step	Function	Action
1	Select record/playback mode	P/$\overline{R}$ = As desired
2A 2B	Begin playback Begin record	P/$\overline{R}$ = HIGH, $\overline{CE}$ = Pulsed LOW P/$\overline{R}$ = LOW, $\overline{CE}$ = Pulsed LOW
3	Pause record or playback	$\overline{CE}$ = Pulsed LOW
4A 4B	End playback End record	Automatic at $\overline{EOM}$ marker or PD = Pulsed HIGH PD = Pulsed HIGH

Table 15: Application Example—Passive Component Functions

Part	Function	Comments
R2	Release time constant	Sets release time for AGC
R4	Series limiting resistor	Reduces level to prevent distortion at higher supply voltages
R6, R7	Pull-up and pull-down resistors	Defines static state of inputs
C1, C4, C5	Power supply capacitors	Filters and bypass of power supply
C2	Attack/Release time constant	Sets attack/release time for AGC
C3	Low-frequency cutoff capacitor	Provides additional pole for low-frequency cutoff

Table 16: Push-Button Parameters

Symbol	Characteristic		Min	Typ (1)	Max	Units	Conditions
T_{CE}	$\overline{CE}$ Pulse Width [Start/Pause]			300		nsec	
T_{SET}	Control/Address Setup Time			300		nsec	
T_{PUD}	Power-Up Delay	ISD2560		25		msec	
		ISD2575		31.25		msec	
		ISD2590		37.25		msec	
		ISD25120		50.0		msec	
T_{PD}	PD Pulse Width [Stop/Reset]			300		nsec	
T_{RUN}	$\overline{CE}$ to $\overline{EOM}$ HIGH		25		400	nsec	
T_{PAUSE}	$\overline{CE}$ to $\overline{EOM}$ LOW		50		400	nsec	
T_{DB}	$\overline{CE}$ HIGH Debounce	ISD2560	70		105	msec	
		ISD2575	85		135	msec	
		ISD2590	105		160	msec	
		ISD25120	135		215	msec	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PUSH-BUTTON TIMING DIAGRAMS

Figure 7: Push-Button Mode Record

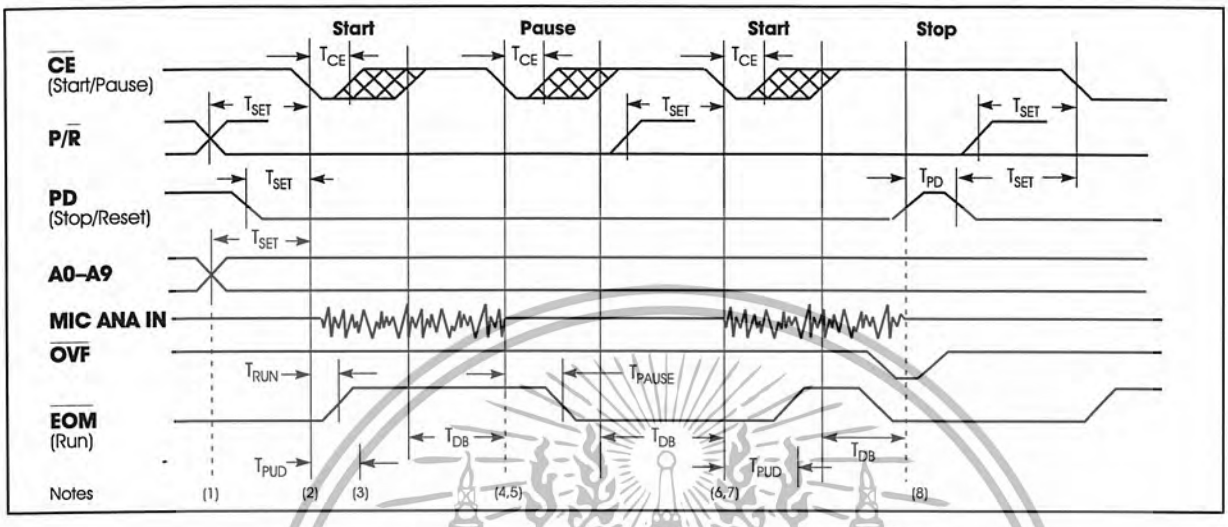
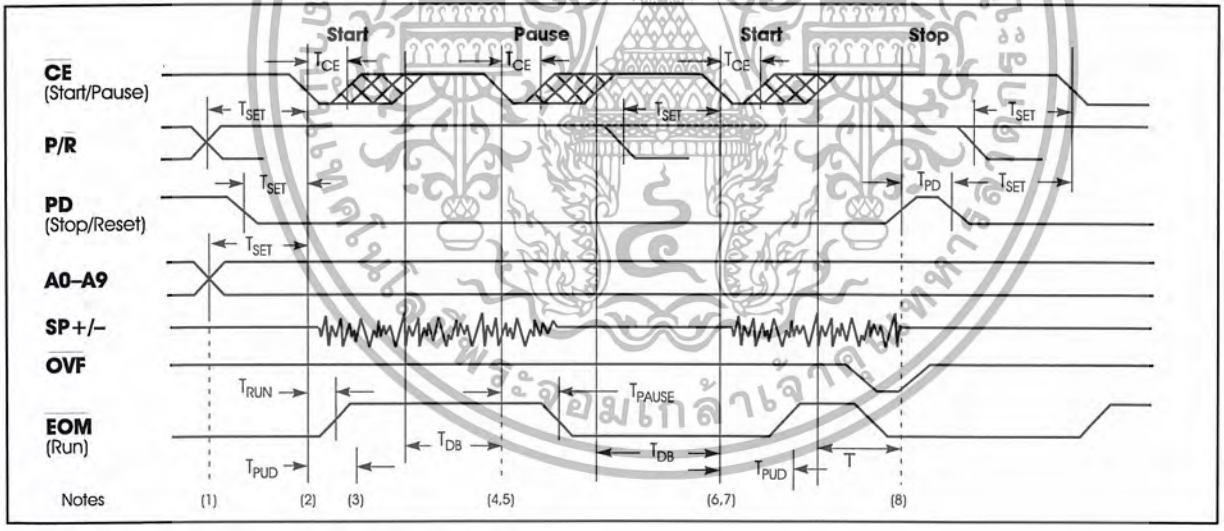


Figure 8: Push-Button Mode Playback



1. *A9, A8, and A6 = 1 for push-button operation.*
2. *The first $\overline{CE}$ LOW pulse performs a Start function.*
3. *The part will begin to play or record after a power-up delay T_{PUD}.*
4. *The part must have $\overline{CE}$ HIGH for a debounce period TDB before it will recognize another falling edge of $\overline{CE}$ and pause.*
5. *The second $\overline{CE}$ LOW pulse, and every even pulse thereafter, performs a Pause function.*
6. *Again, the part must have $\overline{CE}$ HIGH for a debounce period TDB before it will recognize another falling edge of $\overline{CE}$, which would restart an operation. In addition, the part will not do an internal power down until CE is HIGH for the TDB time.*
7. *The third $\overline{CE}$ LOW pulse, and every odd pulse thereafter, performs a Resume function.*
8. *At any time, a HIGH level on PD will stop the current function, reset the address counter, and power down the device.*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DEVICE PHYSICAL DIMENSIONS

Figure 9: 28-Lead 8x13.4mm Plastic Thin Small Outline Package (TSOP) Type I (E)

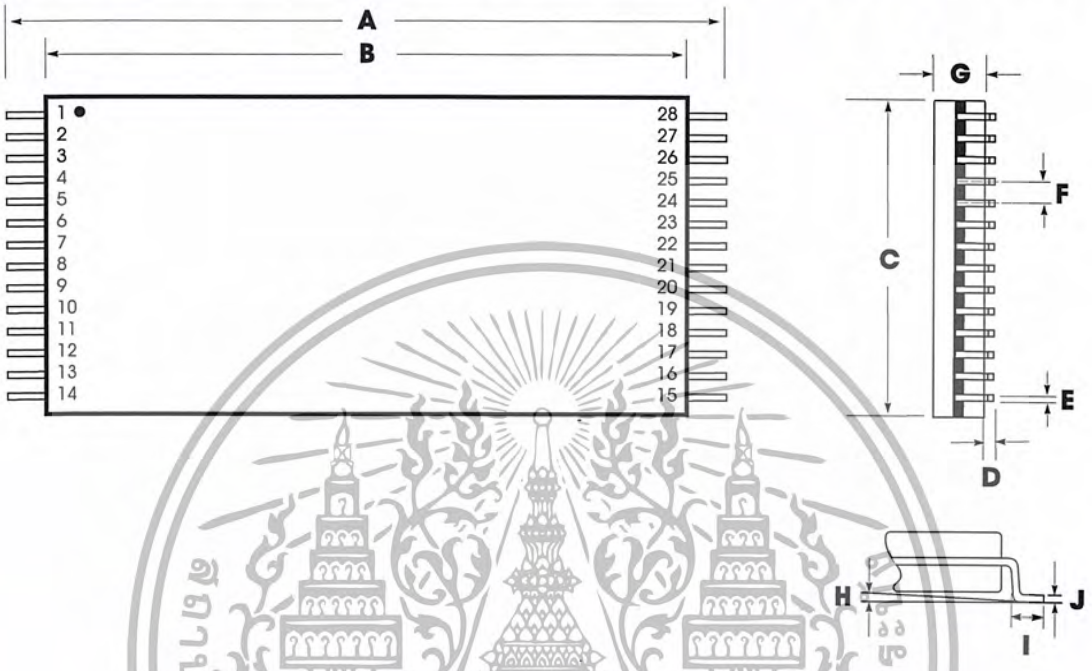


Table 17: Plastic Thin Small Outline Package (TSOP) Type I (E) Dimensions

	INCHES			MILLIMETERS		
	Min	Nom	Max	Min	Nom	Max
A	0.520	0.528	0.535	13.20	13.40	13.60
B	0.461	0.465	0.469	11.70	11.80	11.90
C	0.311	0.315	0.319	7.90	8.00	8.10
D	0.002		0.006	0.05		0.15
E	0.007	0.009	0.011	0.17	0.22	0.27
F		0.0217			0.55	
G	0.037	0.039	0.041	0.95	1.00	1.05
H	0°	3°	6°	0°	3°	6°
I	0.020	0.022	0.028	0.50	0.55	0.70
J	0.004		0.008	0.10		0.21

NOTE: *Lead coplanarity to be within 0.004 inches.*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 10: 28-Lead 0.600-Inch Plastic Dual Inline Package (PDIP) (P)

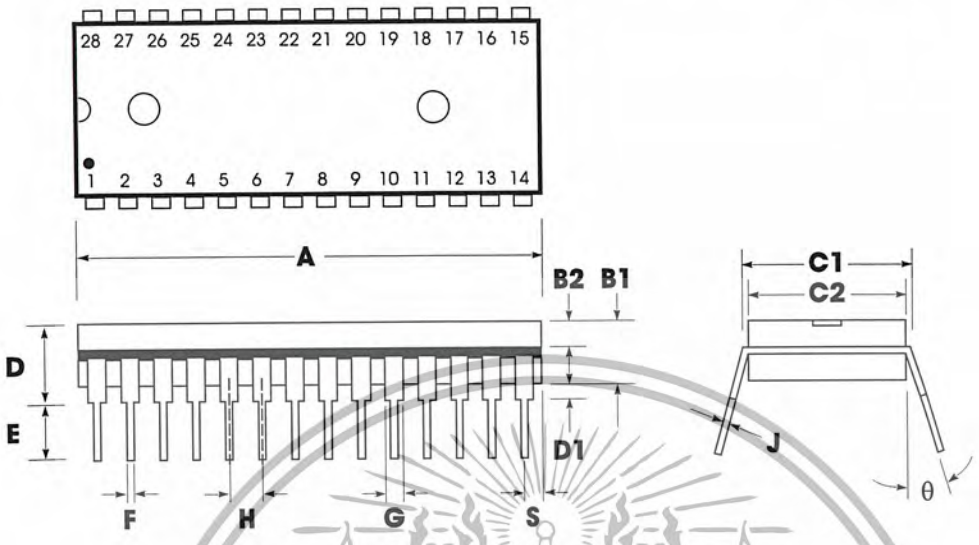


Table 18: Plastic Dual Inline Package (PDIP) (P) Dimensions

	INCHES			MILLIMETERS		
	Min	Nom	Max	Min	Nom	Max
A	1.445	1.450	1.455	36.70	36.83	36.96
B1		0.150			3.81	
B2	0.065	0.070	0.075	1.65	1.78	1.91
C1	0.600		0.625	15.24		15.88
C2	0.530	0.540	0.550	13.46	13.72	13.97
D			0.19			4.83
D1	0.015			0.38		
E	0.125		0.135	3.18		3.43
F	0.015	0.018	0.022	0.38	0.46	0.56
G	0.055	0.060	0.065	1.40	1.52	1.65
H		0.100			2.54	
J	0.008	0.010	0.012	0.20	0.25	0.30
S	0.070	0.075	0.080	1.78	1.91	2.03
q	0°		15°	0°		15°

NOTE: *Lead coplanarity to be within 0.004 inches.*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 11: 32-Lead 8x20mm Plastic Thin Small Outline Package (TSOP) Type I (T)

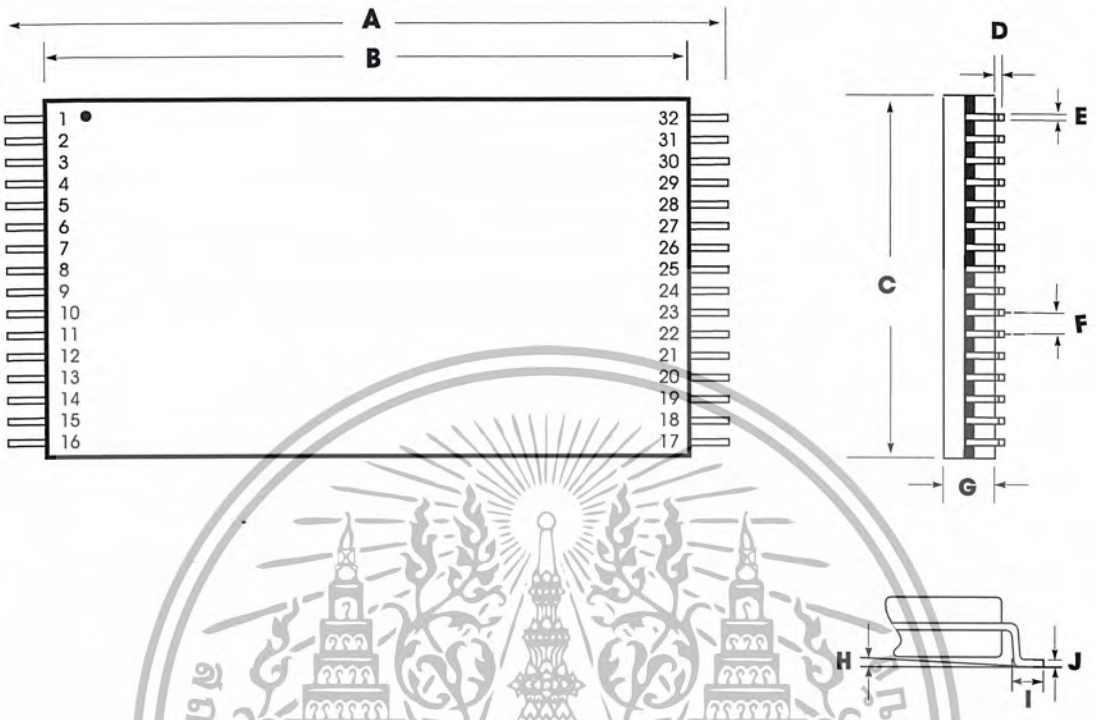


Table 19: Plastic Thin Small Outline Package (TSOP) Type I (T) Dimensions

	INCHES			MILLIMETERS		
	Min	Nom	Max	Min	Nom	Max
A	0.780	0.787	0.795	19.80	20.00	20.20
B	0.720	0.724	0.728	18.30	18.40	18.50
C	0.311	0.315	0.319	7.90	8.00	8.10
D	0.002		0.006	0.05		0.15
E	0.006	0.009	0.011	0.17	0.22	0.27
F		0.0197			0.50	
G	0.037	0.039	0.041	0.95	1.00	1.05
H	0°	3°	5°	0°	3°	5°
I	0.020	0.024	0.028	0.50	0.60	0.70
J	0.004		0.008	0.10		0.21

NOTE: *Lead coplanarity to be within 0.002 inches.*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

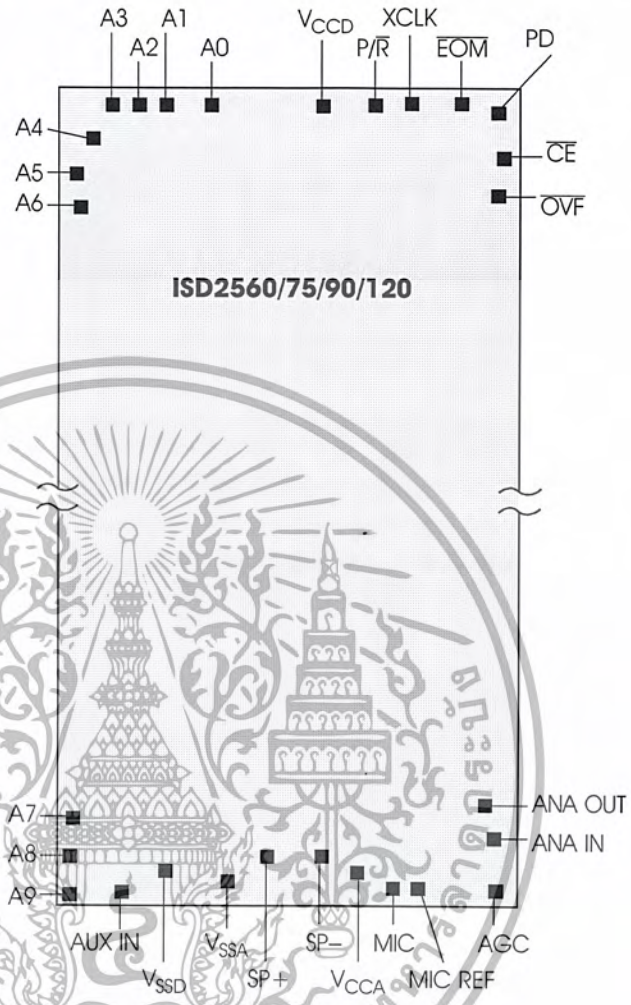
Figure 12: ISD2560/75/90/120 Products *Current* Bonding Physical Layout[1] (Unpackaged Die)

ISD2560/75/90/12[2]

I. Die Dimensions
X: 187 ±1 mils
Y: 399 ±1 mils

II. Die Thickness[2]
17.5 ±1 mils

III. Pad Opening
109 x 109 microns
4.3 x 4.3 mils



1. *The backside of die is internally connected to V_{SS}. It **MUST NOT** be connected to any other potential or damage may occur.*
2. *Die thickness is subject to change, please contact ISD factory for status.*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 20: ISD2560/75/90/120 Products *Current* PIN/PAD Designations, with Respect to Die Center (µm)

Pin	Pin Name	X Axis	Y Axis
A0	Address 0	–1148.9	4898.2
A1	Address 1	–1406.9	4898.2
A2	Address 2	–1661.9	4898.2
A3	Address 3	–1916.9	4898.2
A4	Address 4	–2069.9	4608.2
A5	Address 5	–2194.9	4358.2
A6	Address 6	–2194.9	4108.2
A7	Address 7	–2194.9	–4212.3
A8	Address 8	–2194.9	–4456.3
A9	Address 9	–2076.4	–4897.3
AUX IN	Auxiliary Input	–1607.9	–4868.3
V_{SSD}	V_{SS} Digital Power Supply	–1343.9	–4850.8
V_{SSA}	V_{SS} Analog Power Supply	–551.9	–4884.8
SP+	Speaker Output +	–111.4	–4790.8
SP–	Speaker Output –	425.6	–4790.8
V_{CCA}	V_{CC} Analog Power Supply	865.1	–4848.32
MIC	Microphone Input	1320.7	–4897.3
MIC REF	Microphone Reference	1605.1	–4897.3
AGC	Automatic Gain Control	1877.6	–4871.3
ANA IN	Analog Input	2202.11	–4269.8
ANA OUT	Analog Output	2123.1	–3910.8
$\overline{OVF}$	Overflow Output	2142.6	4154.7
$\overline{CE}$	Chip Enable Input	2202.1	4558.7
PD	Power Down Input	2048.1	4898.2
$\overline{EOM}$	End of Message	1648.1	4865.7
XCLK	No Connect (optional)	1221.1	4898.2
P/$\overline{R}$	Playback/Record	965.6	4898.2
V_{CCD}	V_{CC} Digital Power Supply	646.1	4895.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

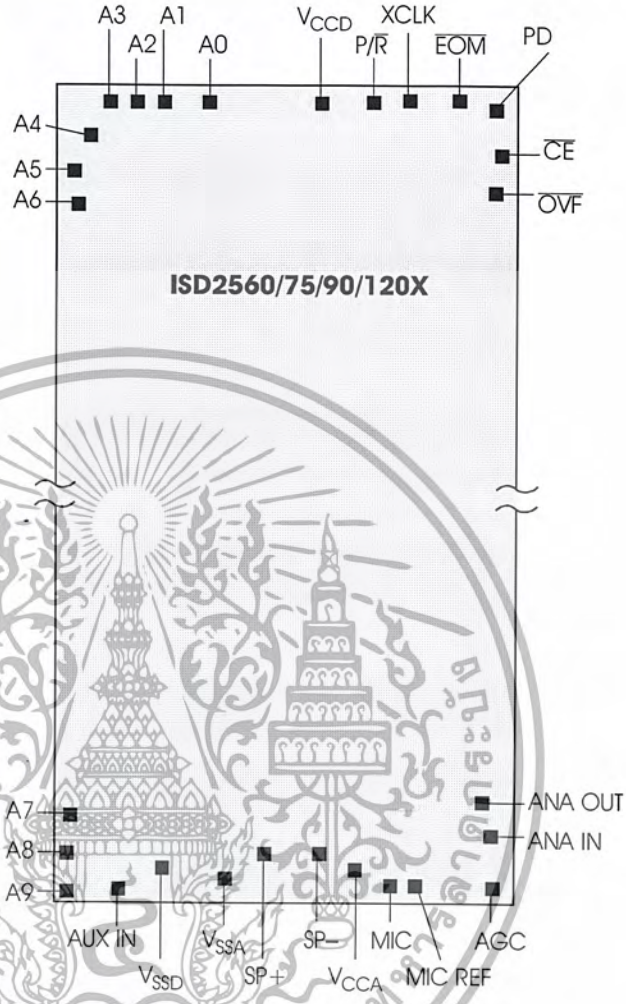
Figure 13: ISD2560/75/90/120 Products *Future* Bonding Physical Layout[1] (Unpackaged Die)

ISD2560/75/90/120X[2]

I. Die Dimensions
X: 149.5 ±1 mils
Y: 262.0 ±1 mils

II. Die Thickness[2]
11.8 ±.4 mils

III. Pad Opening
111 x 111 microns
4.4 x 4.4 mils



1. *The backside of die is internally connected to V_{SS}. It **MUST NOT** be connected to any other potential or damage may occur.*
2. *Die thickness is subject to change, please contact ISD factory for status and availability.*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

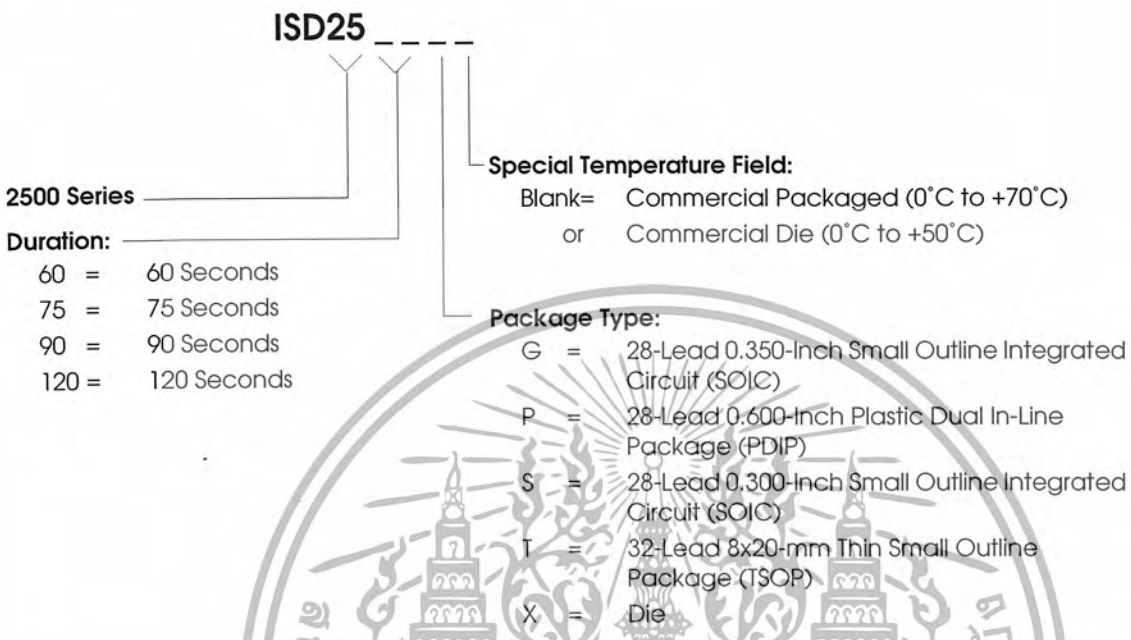
Table 21: ISD2560/75/90/120 Products *Future* PIN/PAD Designations, with Respect to Die Center (µm)

Pin	Pin Name	X Axis	Y Axis
A0	Address 0	–897.9	3135.2
A1	Address 1	–1115.4	3135.2
A2	Address 2	–1331.0	3135.2
A3	Address 3	–1544.0	3135.2
A4	Address 4	–1640.4	2888.9
A5	Address 5	–1698.2	2671.0
A6	Address 6	–1698.2	2441.5
A7	Address 7	–1731.2	–2583.2
A8	Address 8	–1731.2	–2768.4
A9	Address 9	–1731.2	–3050.8
AUX IN	Auxiliary Input	–1410.1	–3115.7
V_{SSD}	V_{SS} Digital Power Supply	–1112.8	–3096.2
V_{SSA}	V_{SS} Analog Power Supply	–407.8	–3138.5
SP+	Speaker Output +	–47.4	–3067.7
SP–	Speaker Output –	386.9	–3067.7
V_{CCA}	V_{CC} Analog Power Supply	746.5	–3110.4
MIC	Microphone Input	1101.2	–3146.0
MIC REF	Microphone Reference	1294.7	–3146.0
AGC	Automatic Gain Control	1666.4	–3130.3
ANA IN	Analog Input	1728.6	–2654.0
ANA OUT	Analog Output	1700.9	–2411.0
$\overline{OVF}$	Overflow Output	1340.9	3121.7
$\overline{CE}$	Chip Enable Input	1726.7	2824.4
PD	Power Down Input	1730.5	3094.0
$\overline{EOM}$	End of Message	1340.9	3121.7
XCLK	No Connect (optional)	986.5	3160.7
P/$\overline{R}$	Playback/Record	807.2	3163.4
V_{CCD}	V_{CC} Digital Power Supply	544.7	3159.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ORDERING INFORMATION

Product Number Descriptor Key



When ordering ISD2560/75/90/120 products, please refer to the following valid part numbers.

Part Number	Part Number	Part Number	Part Number
ISD2560G	ISD2575G	ISD2590G	ISD25120G
ISD2560P	ISD2575P	ISD2590P	ISD25120P
ISD2560S	ISD2575S	ISD2590S	ISD25120X
ISD2560T	ISD2575T	ISD2590T	
ISD2560X	ISD2575X	ISD2590X	

For the latest product information, access ISD's worldwide website at http://www.isd.com.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้