ระบบเซ็นเซอร์หน่วยวัดน้ำอัตโนมัติ

AUTOMATIC WATER SENSOR METER SYSTEM

โดย

นาย ภูเบศวร์ สารทอง

นาย โสภณ มกรครรภ์

นาย เอกศักดิ์ สุวัฒโณ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

ระบบเซ็นเซอร์หน่วยวัดน้ำอัตโนมัติ

AUTOMATIC WATER SENSOR METER SYSTEM

โดย

นาย ภูเบศวร์ สารทอง 46010585

นาย โสภณ มกรครรภ์ 46010897

นาย เอกศักดิ์ สุวัฒโณ 46010999

อาจารย์ที่ปรึกษา

รศ.นิภา ลีลารุจิ

รศ.ณรงค์ เหมกรณ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

ปริญญานิพนธ์ปีการศึกษา 2549

ภาควิชา วิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง **ระบบเซ็นเซอร์หน่วยวัดน้ำอัตโนมัติ**

AUTOMATIC WATER SENSOR METER SYSTEM

ผู้จัดทำ

1. นาย ภูเบศวร์ สารทอง 46010585

2. นายโสภณ มกรครรภ์ 46010897

3. นายเอกศักดิ์ สุวัฒโณ 46010999

นิภา ลีลารุจิ .... อาจารย์ที่ปรึกษา

(รศ. นิภา ลีลารุจิ)

.... อาจารย์ที่ปรึกษา

(รศ. ณรงค์ เหมกรณ์)

ระบบเซ็นเซอร์หน่วยวัดน้ำอัตโนมัติ

AUTOMATIC WATER SENSOR METER SYSTEM

โดย นายภูเบศวร์ สารทอง 46010585
นายโสภณ มกรครรภ์ 46010897
นายเอกศักดิ์ สุวัฒโณ 46010999

อาจารย์ที่ปรึกษา รศ.นิภา ลีลารุจิ
รศ.ณรงค์ เหมกรณ์

บทคัดย่อ

โครงงานนำเสนอระบบหน่วยมาตรวัดน้ำอัตโนมัติ เป็นโครงงานที่นำเซ็นเซอร์มาตรวจจับหน่วยมาตรวัดน้ำของมิเตอร์น้ำ 3 ตัว แล้วนำข้อมูลหน่วยการใช้น้ำที่ได้ไปเข้าไมโครคอนโทลเลอร์ให้เป็นตัวจัดการข้อมูล แล้วทำการส่งข้อมูลดังกล่าวไปยังภาครับ ในส่วนของภาครับจะเก็บข้อมูลที่ได้ไว้เป็นฐานข้อมูลบนคอมพิวเตอร์ เพื่อนำข้อมูลไปวิเคราะห์และแสดงหน่วยการใช้น้ำบนจอแอลซีดีโมดูลตามห้องต่าง ๆ

Abstract

This project present automatic water sensor system that use for detect the unit of 3 water meters. Then move it to microcontroller for contrive and transmit to receiver with radio wave. The receiving path detect data and sent to computer for analyze , management and show the unit of water on LCD Module .

สารบัญ

สารบัญ (ต่อ)

สารบัญรูปภาพ

บทที่ 2

บทที่ 3

สารบัญรูปภาพ (ต่อ)

บทที่ 4

บทที่ 1
บทนำ

1.1 ที่มาของปัญหา

ในปัจจุบันชุมชนได้มีการขยายตัวมากขึ้นทั้งในเมืองหลวงและชนบท ฉะนั้นความต้องการทางด้านสาธารณูปโภคขั้นพื้นฐาน เช่น น้ำ จึงมีความต้องการมากขึ้น โดยจะสังเกตได้จากความต้องการใช้น้ำที่นับวันจะเพิ่มมากขึ้นจนไม่สามารถผลิตได้ทันท่วงที แทบทุกครัวเรือนไม่ว่าจะเป็นในเมืองหลวง ชนบทหรือทุกพื้นที่ที่การประปาสามารถให้บริการได้ถึงนั้น จำเป็นต้องติดตั้งเครื่องมือชนิดหนึ่งซึ่งจะทำหน้าที่ในการตรวจวัดปริมาณน้ำที่แต่ละครัวเรือนได้บริโภคไปในแต่ละเดือน เครื่องมือดังกล่าวก็คือ มิเตอร์น้ำซึ่งจะติดตั้งไว้เพื่อบันทึก ปริมาณน้ำที่แต่ละครัวเรือนได้บริโภคไปในแต่ละเดือนนั้นจะถูกคำนวณเป็นค่าใช้จ่าย ซึ่งประชาชนจะต้องจ่ายค่าบริการดังกล่าวตามสัดส่วนที่ตนเองได้บริโภคไป

ในบริเวณมหาวิทยาลัยต่าง ๆ มีหอพักนักศึกษาเกิดขึ้นมากมาย และในหอพักนักศึกษาที่เกิดขึ้นมานั้นก็จะมีนักศึกษาหรือบุคคลทั่วไป ไปพักอาศัยอยู่เป็นจำนวนมากและในแต่ละเดือนทางหอพักก็จะมีการเก็บค่าเช่าหอพัก ค่าน้ำ ค่าไฟฟ้า และค่าบริการอื่น ๆ แล้วแต่ทางหอพักจะมีการกำหนดค่าไว้ในแต่ละหน่วยที่ใช้บริการ

ดังนั้น จึงได้มีการคิดที่จะหาวิธีการที่จะให้หอพักต่าง ๆ ใช้การเก็บค่าเช่าหอพัก ค่าน้ำ ค่าไฟฟ้า โดยเฉพาะการเก็บค่าน้ำที่มีความสะดวกมากขึ้น โดยที่เจ้าของหอพักไม่ต้องไปคอยเดินจดหน่วยการใช้น้ำตามห้องต่าง ๆ ในแต่ละเดือนให้เสียเวลา

โดยในการทำงานมีขั้นตอนอยู่ว่า เมื่อเริ่มต้นที่เจ้าของหอพักจะต้องทำการคิดค่าน้ำ ก็จะทำได้โดยเรียกใช้โปรแกรมช่วยคิดค่าน้ำ ที่สามารถเรียกดูได้ทีละห้องตามต้องการในเครื่องคอมพิวเตอร์ของเจ้าของหอพักเอง โดยจะมีการติดตั้งอุปกรณ์ที่ใช้อ่านค่าหน่วยมิเตอร์น้ำไว้แล้วในข้างต้น ซึ่งโปรแกรมในคอมพิวเตอร์ก็จะทำการส่งสัญญาณร้องขอข้อมูลการใช้น้ำของแต่ห้องมา เมื่ออุปกรณ์อ่านหน่วยการใช้น้ำประจำมิเตอร์แต่ละตัวได้รับสัญญาณแล้วก็จะทำการส่งข้อมูลการใช้ของตนออกไปให้กับทางโปรแกรมที่ร้องขอมา โปรแกรมก็จะทำการประมวลผลข้อมูลดังกล่าวออกมาเป็นค่าเช่าหอพักที่แต่ละห้องต้องจ่ายในแต่ละเดือนออกมา

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อเพิ่มความสะดวกสบายในการจัดเก็บและคิดค่าน้ำในหอพักต่าง ๆ
2. เพื่อศึกษาการทำงานของระบบไมโครคอนโทรลเลอร์
3. ฝึกทักษะในเรื่องของการออกแบบวงจรต่างๆ
4. สามารถเขียนโปรแกรมเพื่อสร้างระบบการจัดเก็บและคิดค่าน้ำ ได้

1.3 ขอบเขตการดำเนินงาน

ในส่วนแรกเป็นการศึกษาระบบการใช้เซ็นเซอร์ในการตรวจจับการใช้ปริมาณน้ำของมิเตอร์ในแต่หน่วยว่ามีวิธีใดบ้างที่จะใช้ในการดึงข้อมูลที่มิเตอร์น้ำวัดปริมาณน้ำได้ แล้วนำข้อมูลนั้นมาเข้าสู่ระบบการประมวลผลโดยใช้ไมโครคอนโทรเลอร์ รวมทั้งศึกษาระบบการรับส่งข้อมูลโดยอาศัยตัวกลางเป็นคลื่นความถี่วิทยุไปยังคอมพิวเตอร์ซึ่งใช้เป็นฐานข้อมูลในการจัดเก็บ และทำการรวบรวมข้อมูลความถูกต้องของระบบ จากนั้นจึงทำการวิเคราะห์ปัญหาที่จะนำไปสู่ความผิดพลาดที่จะเกิดขึ้น แล้วจึงสรุปและแก้ไขข้อบกพร่องดังกล่าว

ในส่วนที่สองเป็นการออกแบบระบบที่ได้ทำการศึกษามา ทำการประกอบวงจร และทดสอบการใช้งานจริง แล้วทำการสรุปผลการใช้งานระบบเซ็นเซอร์หน่วยวัดน้ำอัตโนมัติและปัญหาที่เกิดขึ้น รวมถึงการประยุกต์การใช้งานในลักษณะอื่นๆ ได้

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. ช่วยเพิ่มความสะดวกสบายในระบบการจัดเก็บข้อมูลการใช้น้ำของผู้ให้บริการ โดยสามารถนำไปใช้งานได้จริงในชีวิตประจำวัน
2. รู้จักการประยุกต์การใช้งานระบบไมโครคอนโทรลเลอร์เพื่อให้เกิดประโยชน์สูงสุด
3. สามารถประยุกต์การใช้งานระบบเซ็นเซอร์ไปใช้ในการตรวจจับการทำงานอย่างอื่นได้ เช่น การตรวจวัดการทำงานของมิเตอร์ไฟฟ้า
4. รู้จักวิธีการทำงานอย่างมีระบบ และสามารถแก้ไขปัญหาอย่างมีหลักการ

บทที่ 2
ทฤษฎีและหลักการ

2.1 ทฤษฎีไมโครคอนโทรลเลอร์ MCS-51

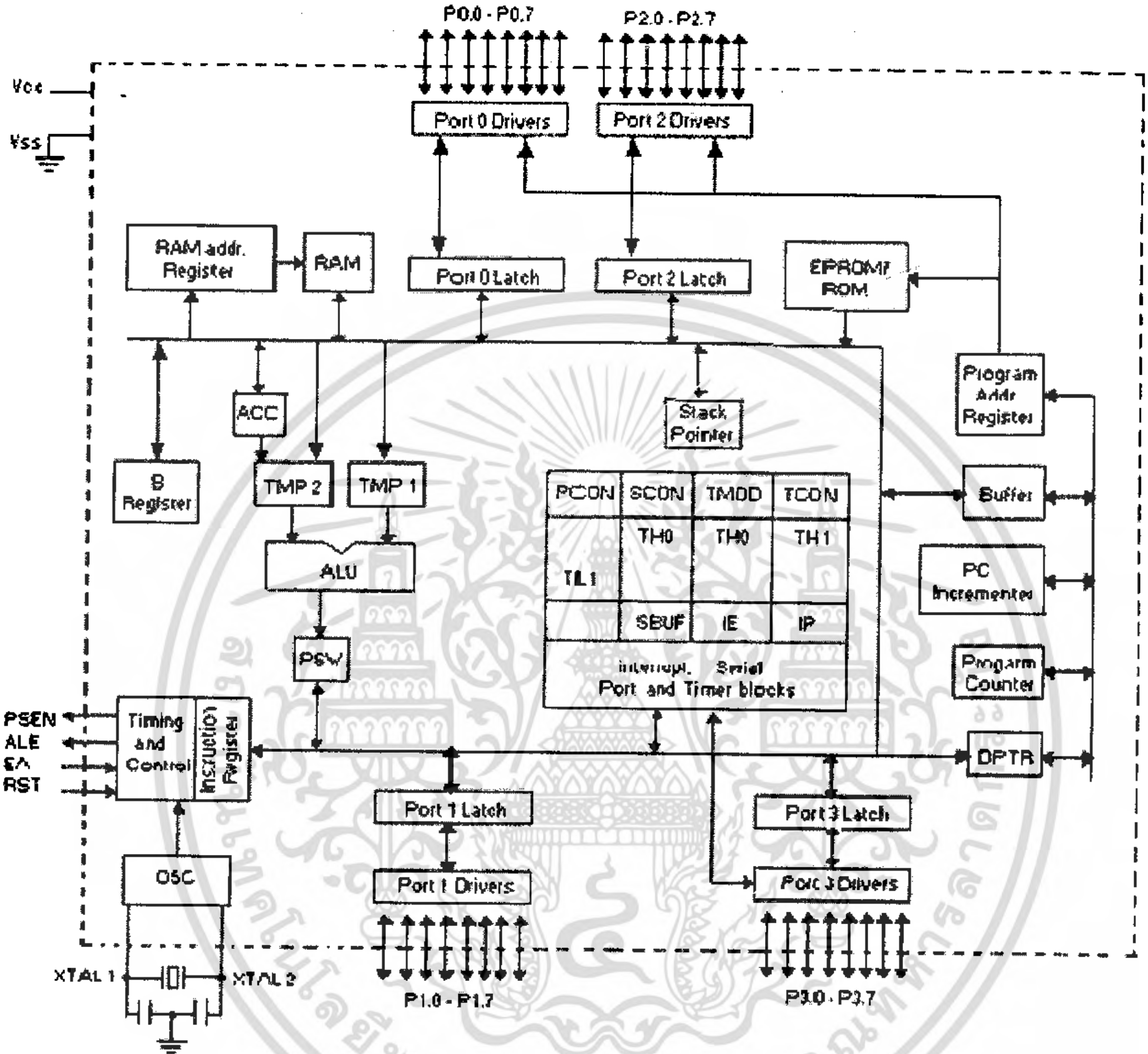
2.1.1 คุณลักษณะพื้นฐานของ MCS-51

คุณสมบัติของไมโครคอนโทรลเลอร์ ตระกูล MCS-51 อนุกรม AT89xx

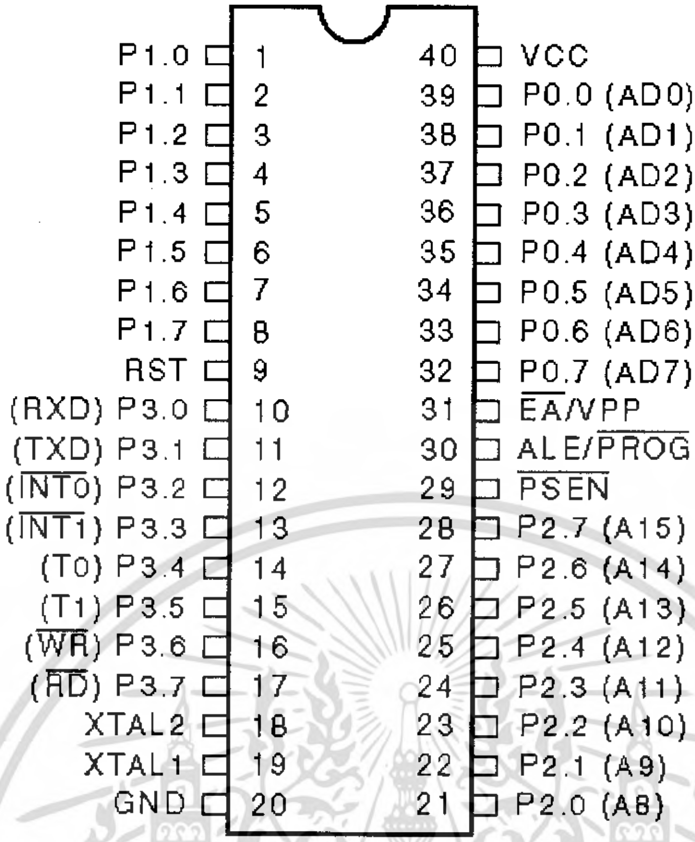
1. เป็นไมโครคอนโทรลเลอร์ที่ใช้ซีพียูขนาด 8 บิต
2. ภายในมีหน่วยความจำโปรแกรมเป็นแบบแฟลช สามารถลบและเขียนใหม่ได้พันครั้ง
3. หน่วยความจำข้อมูลพื้นฐานเป็นหน่วยความจำแบบแรม ในบางเบอร์จะมีหน่วยความจำแบบอีอีพรอมเพิ่มเติม
4. ขาพอร์ตเป็นแบบสองทิศทาง สามารถใช้งานเป็นได้ทั้งอินพุทและเอาท์พุท
5. มีวงจรสื่อสารอนุกรมแบบฟูลดูเพล็กซ์
6. ไทเมอร์/เคาน์เตอร์ขนาด 16 บิต อย่างน้อย 2 ตัว
7. สามารถรองรับแหล่งกำเนิดอินเตอร์รัปต์ได้ 6 ประเภท
8. สามารถขยายหน่วยความจำภายนอกเพิ่มเติมได้สูงสุด 64 กิโลไบต์
9. มีวงจรกำเนิดสัญญาณนาฬิกาอยู่ภายในชิป

2.1.2 ลักษณะการจัดขาของ MCS-51

ไมโครคอนโทรลเลอร์ MCS-51 ทุกเบอร์จะมีสถาปัตยกรรมและขาใช้งานพื้นฐานเหมือนกัน ดังแสดงในรูปที่ 2.1 และ 2.2 โดยมีรายละเอียดขั้นต้นดังนี้



รูปที่ 2.1 แสดงรายละเอียดโครงสร้างหลักของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช ของ ATMEL



รูปที่ 2.2 แสดงการจัดขามาตรฐานของไมโครคอนโทรลเลอร์ MCS-51 ในอนุกรม AT89C5x

ขา Vcc ใช้สำหรับต่อไฟเลี้ยง +5V

ขา GND เป็นขากราวด์ สำหรับต่อกับกราวด์ของระบบ

ขาพอร์ต 0 (P0.0-P0.7) มีขา 8 ขา สามารถกำหนดให้เป็นได้ทั้งอินพุทและเอาท์พุท สำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ต 0 ขาใดขาหนึ่งเป็นอินพุท สามารถทำได้โดยการเขียนข้อมูล “1” ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (float) จึงมีอินพุทอิมพิแดนซ์สูง สามารถใช้งานเป็นขาพอร์ตอินพุทได้ นอกจากนั้นขาพอร์ตนี้ยังถูกใช้งานในการติดต่อกับขาแอดเดรสไบต์ต่ำของหน่วยความจำภายนอก (A0-A7) และขาข้อมูล (D0-D7) โดยใช้กระบวนการมัลติเพล็กซ์เข้าช่วย เพื่อสลับการทำงานเป็นได้ทั้งขาติดต่อแอดเดรส และขาข้อมูล

ขาพอร์ต 1 (P1.0-P1.7) มีขา 8 ขา แต่ละขาสามารถกำหนดให้เป็นทั้งอินพุทและเอาท์พุทสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุท สามารถทำได้โดยการเขียนข้อมูล “1” ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย

ขาพอร์ต 2 (P2.0-P2.7) มีขา 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุทและเอาท์พุทสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุทสามารถทำได้โดยการเขียนข้อมูล “1” ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (float) จึงมีอินพุทอิมพีแดนซ์สูง สามารถใช้งานเป็นขาพอร์ตอินพุทได้ นอกจากนั้นขาพอร์ตนี้ยังถูกใช้งานในการติดต่อกับขาแอดเดรสไบต์สูงของหน่วยความจำภายนอก (A8-A15)

ขาพอร์ต 3 (P3.0-P3.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุทและเอาท์พุทสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุทสามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (float) จึงมีอินพุทอิมพีแดนซ์สูงสามารถใช้งานเป็นขาพอร์ตอินพุทได้ นอกจากนั้นขาพอร์ต 3 ยังเป็นขาที่มีหน้าที่การใช้งานพิเศษ ดังมีรายละเอียดขั้นตอนต่อไปนี้

P3.0 ใช้เป็นขาอินพุทสำหรับรับข้อมูลจากการสื่อสารแบบอนุกรม หรือขา RxD

P3.1 ใช้เป็นขาอินพุทสำหรับรับข้อมูลจากการสื่อสารแบบอนุกรม หรือขา TxD

P3.2 ใช้เป็นขาอินพุทรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 0 หรือขา $\overline{\text{INT0}}$

P3.3 ใช้เป็นขาอินพุทรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 1 หรือขา $\overline{\text{INT1}}$

P3.4 ใช้เป็นขาอินพุทสำหรับรับสัญญาณไทเมอร์จากภายนอกช่อง 0 หรือขา T0

P3.5 ใช้เป็นขาอินพุทสำหรับรับสัญญาณไทเมอร์จากภายนอกช่อง 1 หรือขา T1

P3.6 ใช้เป็นขาสัญญาณ $\overline{\text{WR}}$ ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก

P3.7 ใช้เป็นขาสัญญาณ $\overline{\text{RD}}$ ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก

ขารีเซต (Reset) ใช้ในการรีเซตการทำงานของไมโครคอนโทรลเลอร์ โดยใช้การป้อนสัญญาณเพื่อรีเซตสถานะที่ขานี้ต้องอยู่ในระดับรีเซตอย่างน้อย 2 แมชชีนไซเคิล โดยที่วงจรกำเนิดสัญญาณนาฬิกายังคงทำงานต่อเนื่องไปอย่างเป็นปกติ

ขา ALE /$\overline{\text{PROG}}$ (Address Latch Enable/Program pulse input) เป็นขาที่ใช้ในการควบคุมการแลตช์ของขาพอร์ต 0 เมื่อมีการใช้งานหน่วยความจำภายนอก นอกจากนั้นขานี้ยังใช้เป็นขาสำหรับรับพัลส์ของการโปรแกรมสำหรับโปรแกรมข้อมูลลงในไมโครคอนโทรลเลอร์ MCS-51 ในรุ่นที่มีหน่วยความจำโปรแกรมเป็นแบบอีพรอม

ขา $\overline{\text{PSEN}}$ (Program Store Enable) ขานี้ใช้ในการส่งสัญญาณเพื่อร้องขอติดต่อกับหน่วยความจำโปรแกรมภายนอก เมื่อไมโครคอนโทรลเลอร์ต้องการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอก ตัวไมโครคอนโทรลเลอร์ต้องการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอกตัวไมโครคอนโทรลเลอร์จะส่งสัญญาณออกมาที่ขานี้ 2 ครั้ง ในแต่ละแมชชีนไซเคิล แต่ถ้าหากติดต่อกับหน่วยความจำข้อมูลภายนอกขานี้จะไม่มีการส่งสัญญาณใดๆออกมา

ขา $\overline{\text{EA}}$ /Vpp (External Access enable/Programming voltage input) ใช้สำหรับเลือกการติดต่อหน่วยความจำโปรแกรมจากภายนอกหรือภายในตัวไมโครคอนโทรลเลอร์ ถ้าหากขานี้เป็น "0" เป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำโปรแกรมภายนอก แต่ถ้าหากขานี้เป็น "1" เป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำภายในตัวไมโครคอนโทรลเลอร์ นอกจากนี้ที่ขานี้ยังใช้เป็นขาอินพุทสำหรับรับแรงดันไฟสูงสำหรับการโปรแกรมหน่วยความจำภายในไมโครคอนโทรลเลอร์ สำหรับไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชต้องการแรงดันสำหรับการโปรแกรม +5V

ขา XTAL1 และ XTAL2 เป็นขาสำหรับต่อคริสตัลเพื่อสร้างสัญญาณนาฬิกาในการกำหนดจังหวะการทำงานของไมโครคอนโทรลเลอร์

2.1.3 พอร์ตอินพุทและพอร์ตเอาท์พุท

พอร์ต คือ แอดเดรสหนึ่งที่ได้รับการกำหนดไว้เพื่อการโอนย้ายข้อมูลระหว่างไมโครคอนโทรลเลอร์กับอุปกรณ์ภายนอก การกำหนดประเภทของการติดต่อขึ้นอยู่กับทิศทางการไหลของข้อมูล เมื่อพิจารณาจากไมโครคอนโทรลเลอร์เป็นหลัก จากรูปที่ 2.3(a) เป็นการใช้ไมโครคอนโทรลเลอร์เป็นเอาท์พุทพอร์ต และจากรูป 2.3(b) เป็นการใช้ไมโครคอนโทรลเลอร์เป็นอินพุทพอร์ต

รูปที่ 2.3 แสดงการใช้ไมโครคอนโทรลเลอร์เป็นอินพุทและเอาท์พุทพอร์ต

2.1.3.1 การใช้งานพอร์ตเป็นอินพุท

การใช้งานพอร์ตเป็นการอินพุทข้อมูลจะต้องเริ่มต้นด้วยการส่งข้อมูลที่มีค่าเป็น 1 ออกมาทางบิตของพอร์ตสั้นก่อนเป็นอันดับแรก เพื่อหยุดการทำงานของทรานซิสเตอร์ที่ทำหน้าที่ขับสัญญาณเอาท์พุทของบิตนั้น ทำให้ขาสัญญาณของบิตถูกต่อเข้ากับตัวต้านทานซึ่งทำหน้าที่ Pull-up ภายในซึ่งมีผลทำให้บิตนั้นของพอร์ต 1 , 2 และ 3 เป็นสภาวะลอจิกสูง ตัวต้านทานนี้มี

ค่าประมาณ 50 k ohm ซึ่งเป็นค่าที่สูงมาก และทำให้อุปกรณ์ภายนอกสามารถขับสัญญาณของพอร์ตเหล่านี้เป็นลอจิกต่ำได้ง่าย สำหรับบิตของพอร์ต 0 นั้นแม้ว่าจะมีหลักการทำงานที่คล้ายคลึงกันกับบิตของพอร์ตอื่นๆ แต่เนื่องจากไม่มีตัวต้านทานซึ่งทำหน้าที่ Pull-up ภายในไว้ ทำให้เมื่อทรานซิสเตอร์ที่ทำหน้าที่ขับสัญญาณเอาท์พุทนั้นหยุดการทำงาน ก็จะเป็นผลให้สัญญาณนี้อยู่ในสภาวะอิมพีแดนซ์สูงแทน

2.1.3.2 การใช้งานพอร์ตเป็นเอาท์พุท

เมื่อมีการส่งข้อมูลที่มีค่าเป็น 0 ให้กับแต่ละบิตของพอร์ตทุกพอร์ต ข้อมูลนี้จะถูกส่งให้กับฟลิปฟลอป ซึ่งจะค้างค่านี้ไว้และมีผลทำให้ทรานซิสเตอร์ที่ทำหน้าที่ขับสัญญาณเอาท์พุทนั้นทำงาน ดังนั้นขาสัญญาณก็จะมีสภาวะลอจิกเป็นลอจิกต่ำด้วย

ส่วนการส่งข้อมูลที่มีค่าเป็น 1 ออกมานั้น ในกรณีที่เป็นการทำงานในแต่ละบิตของพอร์ต 1 , 2 หรือ 3 จะทำให้ทรานซิสเตอร์ที่ทำหน้าที่ขับสัญญาณเอาท์พุทนั้นหยุดทำงาน มีผลทำให้ขาของสัญญาณเป็นลอจิกสูงด้วยตัวต้านทานที่ Pull-up อยู่ภายในนั้น แต่สำหรับการใช้งานในแต่ละบิตทางพอร์ต 0 นั้นจะมีผลแตกต่างออกไป โดยขาสัญญาณจะมีสภาวะอิมพีแดนซ์สูงแทน เนื่องจากไม่มีตัวต้านทานภายในเชื่อมต่ออยู่นั้นเอง

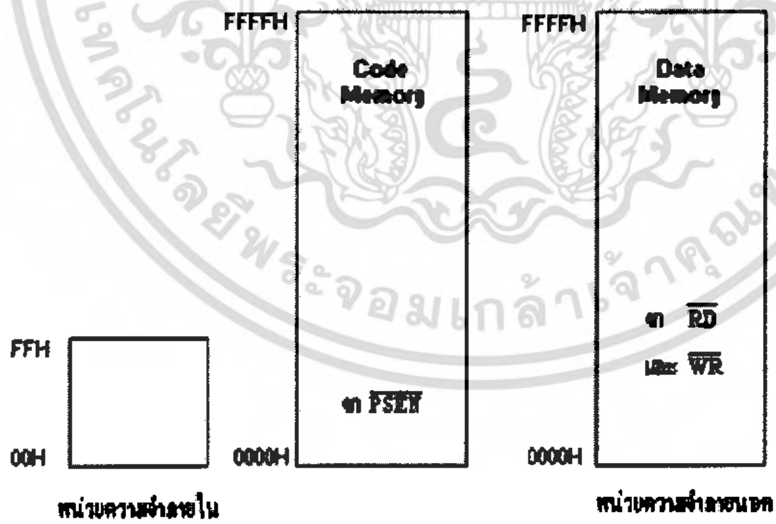
ดังนั้นการใช้งานพอร์ต 0 เป็นการนำข้อมูลออกทางเอาท์พุท จึงจำเป็นต้องใช้ตัวต้านทานภายนอก Pull-up สัญญาณไว้กับลอจิกสูงแทน

2.1.4 หน่วยความจำโปรแกรมของ MCS-51

หน่วยความจำโปรแกรม(Program Memory) มีไว้เพื่อบรรจุคำสั่งหรือโปรแกรมที่ผู้ใช้พัฒนาขึ้นจัดเก็บไว้ในหน่วยความจำ โดยอาจจะประกอบอยู่ภายในตัวของไอซี 89C51 เอง หรือเป็นไอซีหน่วยความจำ EPROM หรือ ROM แยกออกต่างหากได้ ในกรณีหลังจำเป็นต้องมีการใช้พอร์ตอินพุทเอาท์พุท ทำหน้าที่เป็นบัสแอดเดรส และบัสข้อมูลเพื่อให้สามารถติดต่อกับหน่วยความจำมาตรฐานทั่วไปได้

หน่วยความจำโปรแกรมของ 89C51 เป็นบริเวณหน่วยความจำสำหรับเก็บข้อมูลและคำสั่งใช้งานต่างๆ ซึ่งแม้ว่าจะไม่มีการจ่ายกระแสไฟฟ้าให้ระบบข้อมูลเหล่านี้ก็ยังไม่สูญหายโครงสร้างของหน่วยความจำโปรแกรมมีลักษณะเช่นเดียวกับหน่วยความจำที่บรรจุอยู่ในไอซีหน่วยความจำประเภทต่างๆเช่น หน่วยความจำแบบ ROM (Read Only Memory) หรือ EPROM (Erasable Programmable Read Only Memory) การจัดพื้นที่ของหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ 89C51 จะมีการจัดพื้นที่ดังรูปที่ 2.4 โดยที่ไมโครคอนโทรลเลอร์ 89C51 สามารถอ่านข้อมูลหน่วยความจำโปรแกรมสูงสุดได้ไม่เกิน 64 กิโลไบต์ และแยกประเภทของหน่วยความจำโปรแกรมเป็น 2 ลักษณะตาม

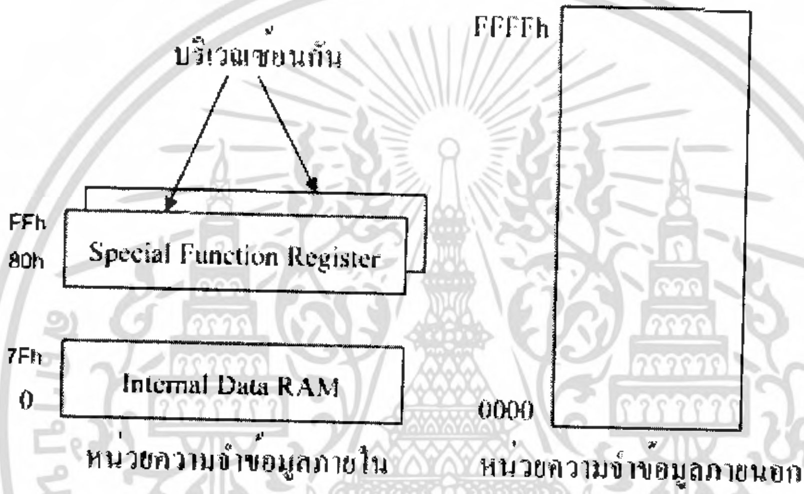
ตำแหน่งของหน่วยความจำนั้น คือ หน่วยความจำโปรแกรมภายใน (Internal Program Memory) ซึ่งเป็นหน่วยความจำ ROM หรือ EPROM ที่อยู่ภายในตัวไอซี ไมโครคอนโทรลเลอร์เองและหน่วยความจำภายนอก (External Program Memory) ซึ่งเป็นการใช้ไอซีหน่วยความจำมาทำหน้าที่เป็นหน่วยความจำโปรแกรมของระบบ



รูปที่ 2.4 แสดงการจัดพื้นที่ของหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ MCS-51

2.1.5 หน่วยความจำข้อมูลของ MCS-51

หน่วยความจำข้อมูลมีหน้าที่สำหรับเก็บข้อมูลหรือตัวแปรที่เกิดขึ้นในขณะที่กำลังประมวลผลโปรแกรมไว้เป็นการชั่วคราว โดยที่หน่วยความจำข้อมูลจะมีลักษณะเป็นหน่วยความจำ RAM แบบสแตติก (Static) ดังนั้นเมื่อไม่มีการจ่ายไฟให้กับระบบ ก็จะมีผลทำให้ข้อมูลที่จัดเก็บไว้ในหน่วยความจำนี้สูญหายไป พื้นที่ของหน่วยความจำข้อมูลของ 89C51 สามารถมีได้ไม่เกิน 64 กิโลไบต์ และแยกประเภทออกเป็นสองลักษณะตำแหน่งที่ตั้งของหน่วยความจำนั้นดังลักษณะในรูปที่ 2.5 คือ หน่วยความจำโปรแกรมภายใน (Internal Date Memory) หรือ RAM ที่อยู่ภายในตัวไมโครคอนโทรลเลอร์เอง และหน่วยความจำข้อมูลภายนอก (External Data Memory) ซึ่งเป็นการใช้ไอซีหน่วยความจำ RAM มาเพิ่มเติมเข้าไปในวงจร ลักษณะเดียวกับการนำไอซี EPROM มาใช้งานเป็นหน่วยความจำโปรแกรมนั่นเอง



รูปที่ 2.5 แสดงลักษณะตำแหน่งที่ตั้งของหน่วยความจำข้อมูล

2.1.5.1 หน่วยความจำข้อมูลภายใน

หน่วยความจำข้อมูลภายใน 89C51 มีจำนวนทั้งหมด 256 ไบต์ โดยจำแนกออกได้เป็นสองลักษณะ คือ พื้นที่เฉพาะสำหรับตัวประมวลผลกลางใช้งานเท่านั้นซึ่งเรียกว่ารีจิสเตอร์ และพื้นที่ใช้งานทั่วไปสำหรับโปรแกรมใช้งานที่ผู้ใช้สร้างขึ้นมา จากรูปที่ 2.6 แสดงถึงการจัดพื้นที่ของหน่วยความจำข้อมูลภายในของ 89C51 ซึ่งแบ่งเป็นสองส่วน คือ หน่วยความจำขนาด 128 ไบต์แรก และหน่วยความจำขนาด 128 ไบต์ถัดไป

2.1.5.2 หน่วยความจำขนาด 128ไบต์แรก

บริเวณนี้จะมีตำแหน่งแอดเดรสอยู่ในช่วง 00H ซึ่งแบ่งได้เป็นอีกสามส่วนดังนี้

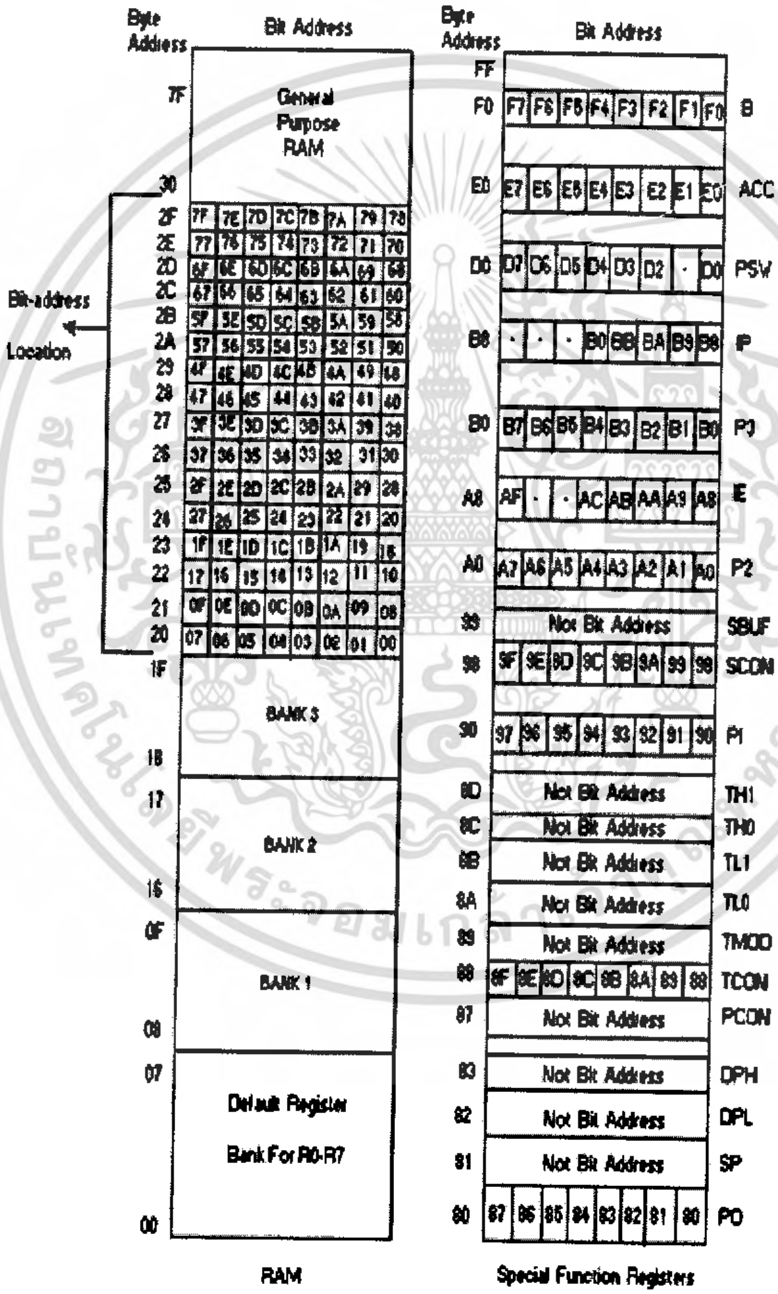
* บริเวณแอดเดรส 00H-1FH จำนวน 32 ไบต์ จำแนกออกเป็นกลุ่มหรือแบงก์ข้อมูลจำนวน 8 ไบต์ รวมทั้งหมดสี่กลุ่ม พื้นที่ข้อมูลในแต่ละกลุ่มจะถูกใช้งานในฐานะของรีจิสเตอร์ทั่วไป เรียกว่า รีจิสเตอร์ R0-R7

* บริเวณแอดเดรส 20H-2FH จำนวน 16 ไบต์ จะเป็นพื้นที่ส่วนสำหรับผู้ใช้ที่สามารถอ้างถึงข้อมูลได้ทั้งแบบไบต์และแบบบิต

* บริเวณแอดเดรส 30H-7FH เป็นบริเวณที่สามารถนำไปใช้งานได้อย่างอิสระ โดยสามารถอ้างอิงได้เฉพาะในลักษณะของไบต์ข้อมูลตามปกติเท่านั้น

2.1.5.3 หน่วยความจำขนาด 128 ไบต์ถัดไป

พื้นที่ตั้งแต่แอดเดรส 80H-FFH เป็นหน่วยความจำที่นำมาใช้งานเป็นรีจิสเตอร์หน้าที่พิเศษ แต่ก็ยังมีบริเวณของหน่วยความจำที่อยู่บริเวณนี้ที่ผู้ใช้สามารถเก็บข้อมูลได้ แต่การเรียกใช้งานจะต้องมีการเข้าถึงข้อมูลแบบโดยอ้อมเท่านั้น



รูปที่ 2.6 แสดงตำแหน่งของหน่วยความจำแบบไบต์และแบบบิต

2.1.6 รีจิสเตอร์หน้าที่พิเศษ

เป็นรีจิสเตอร์ที่ทำหน้าที่ควบคุมการทำงานของอุปกรณ์หรือพอร์ตของ 89C51 ทั้งหมด โดยมีตำแหน่งอยู่ในบริเวณแอดเดรส 80H-FFH การใช้งานรีจิสเตอร์หน้าที่พิเศษเหล่านี้สามารถทำได้ทั้งการระบุถึงชื่อรีจิสเตอร์หรือตำแหน่งแอดเดรสที่เป็นของรีจิสเตอร์นั้นก็ได้

2.1.6.1 แอกคิวมูเลเตอร์ (Accumulator)

เป็นรีจิสเตอร์ขนาด 8 บิต ทำหน้าที่เก็บข้อมูลที่ส่งให้หน่วยทำงานในซีพียูและเก็บผลลัพธ์ที่ได้จากการทำงานนั้น การใช้งานในโปรแกรมจะเรียกว่า รีจิสเตอร์ A

2.1.6.2 รีจิสเตอร์ B

เป็นรีจิสเตอร์ที่ใช้สำหรับการทำคำสั่งการคูณหารตัวเลข ในกรณีที่ไม่ใช้การคำนวณทางด้านคณิตศาสตร์ ก็สามารถนำไปใช้งานเช่นเดียวกับรีจิสเตอร์ทั่วไปได้

2.1.6.3 โปรแกรมเคาน์เตอร์(Program Counter)

เป็นรีจิสเตอร์ที่ใช้สำหรับการชี้ตำแหน่งแอดเดรสของหน่วยความจำโปรแกรม ซึ่งจะต้องไปทำงานในลำดับถัดไป การใช้งานในโปรแกรมจะเรียกว่า รีจิสเตอร์ PC

2.1.6.4 สแต็กพอยน์เตอร์(Stack Pointer)

เป็นรีจิสเตอร์ขนาด 8 บิต ทำหน้าที่เก็บตำแหน่งของตัวชี้หรือพอยน์เตอร์ของบริเวณสแต็กสำหรับเก็บข้อมูลแอกคิวมูเลเตอร์ รีจิสเตอร์ต่างๆ รวมทั้งข้อมูลจากโปรแกรม ค่าเริ่มต้นของสแต็กจะอยู่ที่ตำแหน่ง 07H การใช้งานในโปรแกรมจะเรียกว่ารีจิสเตอร์ SP

2.1.6.5 ตัวชี้ข้อมูลหรือดาต้าพอยน์เตอร์(Data Pointer)

เป็นรีจิสเตอร์ขนาด 16 บิต ซึ่งเรียกว่า รีจิสเตอร์ DPTR และสามารถใช้งานแยกออกเป็นรีจิสเตอร์ขนาด 8 บิต สองตัว คือ รีจิสเตอร์ DPH และ DPL เพื่อเก็บค่าแอดเดรสของหน่วยความจำที่จะต้องใช้งานภายในโปรแกรม หรืออาจเป็นแอดเดรสของอุปกรณ์ภายนอก

2.1.6.6 โปรแกรมสเตตัสเวิร์ด(PSW)

รีจิสเตอร์นี้ทำหน้าที่บอกถึงแฟลกสภาวะการทำงานต่างๆ รวมทั้งบิตสำหรับการกำหนดเลือกแบงก์ (Bank) ของรีจิสเตอร์ที่ใช้งานด้วย

2.1.6.7 รีจิสเตอร์ที่เกี่ยวข้องกับพอร์ต(Port Register)

รีจิสเตอร์เหล่านี้จะมีความเกี่ยวข้องกับการทำงานของพอร์ตอินพุทเอาท์พุทโดยตรงซึ่งจะเป็นรีจิสเตอร์ขนาด 8 บิต สามารถใช้งานได้ทั้งในลักษณะการอินพุทหรือการเอาท์พุทข้อมูลได้

2.1.6.8 รีจิสเตอร์ SBUF

เป็นบัฟเฟอร์ขนาด 8 บิต สำหรับการสื่อสารข้อมูลแบบอนุกรมทั้งการรับและการส่งข้อมูล

2.1.6.9 รีจิสเตอร์ PCON

เป็นรีจิสเตอร์ควบคุมการทำงานในสามลักษณะ ซึ่งได้แก่ การควบคุมการทำงานของโปรแกรมเซ็นเซอร์ การกำหนดอัตราทวีคูณของอัตราเร็วในการสื่อสารข้อมูลอนุกรมและแฟลก สภาวะการทำงานทั่วไป

2.1.6.10 รีจิสเตอร์ IP,IE,TMOD,SCON

เป็นกลุ่มรีจิสเตอร์ที่ทำหน้าที่ควบคุมการทำงานของอินเตอร์รัปต์ต่างๆ

2.2 การแสดงผลทางจอ LCD

จอ LCD เป็นอุปกรณ์ที่ใช้แสดงผลโดยในโครงงานนี้เราเลือกใช้ LCD ที่มีขนาด 16 ตัวอักษร 1บรรทัด ซึ่งทำหน้าที่รับข้อมูลจากไมโครคอนโทรลเลอร์เพื่อแสดงผลออกทางหน้าจอเราเลือกใช้จอ LCDรุ่น BC1602HGREH เพราะว่ามีจำนวนตัวอักษรและมีขนาดที่เหมาะสมรวมทั้งราคาที่ไม่แพงมากจนเกินไป

2.2.1 การใช้งานจอ LCD

ในโมดูลของจอ LCD นั้นประกอบไปด้วย 3 ส่วนหลักๆ คือ

ตัวแสดงผล (display) ซึ่งในโครงสร้างภายในนั้นเป็นผลึกเหลวที่สามารถแสดงผลให้เห็นโดยอาศัยแสงจากภายนอก ดังนั้นการมองจอ LCD ต้องอาศัยมุมมองที่เหมาะสมเพื่อที่จะมองข้อมูลที่แสดงอยู่บนจอ LCD

ตัวควบคุม (controller) เป็นส่วนที่ทำหน้าที่ในการรับข้อมูลจากภายนอกมาควบคุมการทำงานของโมดูล LCD เช่น การลบจอภาพ การแสดงตัวอักษร หรือ การเลื่อนเคอร์เซอร์ เป็นต้น ซึ่งส่วนควบคุมนี้จะใช้ชิปที่ทำหน้าที่ในการควบคุม โดยเฉพาะ ซึ่งเป็นที่นิยมก็ได้แก่ ชิปเบอร์ HD44780 และ HD61830 โดยHD44780 นั้นจะใช้ควบคุม LCD แบบอักขระ ส่วน HD61830 จะใช้ในการควบคุม LCD แบบกราฟิก

ตัวขับ (driver) เป็นส่วนที่ทำหน้าที่รับสัญญาณจากตัวควบคุมมาทำการขับให้ตัวแสดงผลข้อมูลตามที่ต้องการ ซึ่งชิปที่ทำหน้าที่เป็นตัวขับนี้ได้แก่ ชิปเบอร์ HD44100H และ MSM5259 เป็นต้น

2.2.2 โครงสร้างภายในของตัวควบคุมโมดูล LCD

โครงสร้างภายในที่จะนำเสนอนี้เป็นโครงสร้างของ LCD แบบอักขระซึ่งชิปที่ควบคุมคือ HD44780ซึ่งจะประกอบไปด้วย

บัฟเฟอร์อินพุตและเอาต์พุต เป็นส่วนที่ใช้ในการติดต่อรับส่งข้อมูลกับอุปกรณ์ภายนอก เพื่อที่จะถ่ายทอดข้อมูลเข้าออกภายในตัวควบคุม

รีจิสเตอร์คำสั่ง (Instruction Register : IR) เป็นรีจิสเตอร์ใช้รับข้อมูลคำสั่งจากอุปกรณ์ภายนอกเพื่อนำไปควบคุมการแสดงผล

รีจิสเตอร์ข้อมูล (Data Register : DR) เป็นรีจิสเตอร์ที่ใช้รับข้อมูลจากภายนอก เพื่อถ่ายต่อไปยังหน่วยความจำที่ทำหน้าที่เก็บข้อมูลแสดงผล หรือนำข้อมูลไปสร้างตัวอักษรเพิ่มเติมในแรมเก็บตัวอักษร

แรมเก็บข้อมูลแสดงผล (Display Data Ram : DDRAM) เป็นหน่วยความจำแรมที่ทำหน้าที่เก็บข้อมูลที่ได้จากรีจิสเตอร์ DR ตัวควบคุมจะนำข้อมูลใน DDRAM ไปเปิดตารางของตัวอักษรที่เก็บไว้ในหน่วยความจำรอม และ แรมเก็บตัวอักษร เพื่อนำไปแสดงที่ตัวแสดงผล

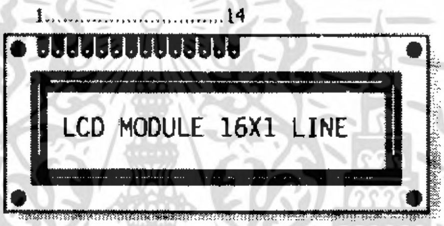
รอมเก็บตัวอักษร (Character Generator ROM : CGROM) เป็นหน่วยความจำรอมที่ใช้เก็บข้อมูลตัวอักษรหรือสัญลักษณ์ ที่สามารถอ่านออกไปแสดงที่ตัวแสดงผลได้ มีขนาด 7,200 บิตโดยจะถูกอ่านออกไปด้วยค่าของข้อมูลใน DDRAM

แรมเก็บตัวอัก ษร (Character Generator RAM : CGRAM) เป็นหน่วยความจำแรมที่ใช้เก็บตัวอักษรที่มีการสร้างเพิ่มเติมขึ้นใหม่ ในกรณีที่ตัวอักษรใน CGROM ไม่เพียงพอ มีขนาด 512 บิตการเขียนและอ่านค่าไปใช้นั้นทำเช่นเดียวกับ CGROM คือ จะทำการเขียนข้อมูลลงใน DDRAM ก่อน จากนั้นจะเป็นหน้าที่ของตัวควบคุมที่จะมาอ่านค่าจาก CGRAM เอง

แฟลก BUSY เป็นส่วนที่ทำหน้าที่แจ้งสถานะการทำงานของตัวควบคุมให้อุปกรณ์ภายนอกได้ทราบว่า ตัวควบคุมพร้อมที่จะรับข้อมูลหรือคำสั่งหรือไม่ ดังนั้นก่อนการส่งข้อมูลหรือคำสั่งมายังตัวควบคุมต้องตรวจสอบสถานะของแฟลก BUSY นี้เสียก่อน

2.2.3 โมดูลของจอ LCD ขนาด 16 ตัวอักษร

ในจอ LCD ชนิดนี้มีขาในการต่อใช้งานทั้งหมด 14 ขาดังรูปที่ 2.7 โดยการทำงานของขาแต่ละขาเป็นดังต่อไปนี้



ขา 1 : GND

ขา 2 : +V

ขา 3 : Brightness ปรับความสว่าง

ขา 4 : RS

ขา 5 : R/$\overline{W}$

ขา 6 : E

ขา 7-14 : D0-D7

รูปที่ 2.7 แสดงรูปร่างและการจัดขาโมดูล LCD แบบอักขระ

VSS (ขา1) : ต่อลงกราวด์

VDD (ขา2) : ต่อไฟเลี้ยง +5 โวลต์

VO (ขา3) : เป็นขาอินพุตรับแรงดันเพื่อปรับความเข้าของการแสดงผล

RS (ขา4) : เป็นขาอินพุตที่ใช้ในการแยกชนิดของข้อมูลที่ ำการประมวลผลในขณะนั้นว่าเป็นคำสั่งสำหรับรีจิสเตอร์ IR หรือ เป็นข้อมูลสำหรับรีจิสเตอร์ DR โดยถ้าขานี้เป็น “0” ข้อมูลที่ส่งมาจะเป็นคำสั่ง แต่ถ้าขานี้เป็น “1” ข้อมูลที่ส่งมาจะเป็นข้อมูลที่ใช้สำหรับการแสดงผล

R/W(ขา5) : เป็นขาที่ใช้เลือกเพื่อการอ่านหรือการเขียนข้อมูลกับ LCD ถ้าเป็น “0” เป็นการกำหนดให้เขียนข้อมูล แต่ถ้าเป็น “1” จะเป็นการอ่านข้อมูล

E (ขา6) : เป็นขา Enableจอ LCD ให้ทำงาน

D0-D7 (ขา 7-14) : เป็นขาที่ใช้ในการส่งข้อมูล 8 บิตเข้ามาสู่จอ LCD

2.2.4 การใช้งานด็อทเมตริกซ์แอลซีดีโมดูล (Dot Matrix LCD Module)

ปัจจุบันแอลซีดีโมดูลเป็นที่นิยมกันอย่างมาก สำหรับการแสดงผลในเครื่องใช้ต่างๆ ทั้งนี้เนื่องจากมีความเหมาะสมในหลายๆด้าน เช่นการใช้กระแสต่ำ สามารถแสดงผลเป็นตัวอักษรและตัวเลขหรือแสดงเป็นกราฟฟิก (เฉพาะรุ่น) ในขณะนี้ผู้ผลิตแอลซีดีโมดูลจะผลิตแอลซีดีรุ่นที่เป็นโมดูลที่มีตัวแอลซีดีและวงจรควบคุมมาให้พร้อมออกมา ทำให้ผู้ใช้สามารถต่อเข้ากับระบบไมโครคอนโทรลเลอร์ ได้ง่ายและสะดวกสำหรับการเขียนโปรแกรม

แอลซีดีโมดูลมีอยู่หลายรุ่นและคุณสมบัติต่างกันออกไป แบ่งได้ 2 กลุ่มคือ แบบด็อทเมทริกซ์ และกราฟฟิก โดยแบบด็อทเมตริกซ์จะแสดงผลเป็นตัวอักษรขนาด 5 x 8 ดอท (Dot) และมีจำนวนตัวอักษรและบรรทัดแตกต่างกันออกไปในแต่ละรุ่น ส่วนแบบกราฟฟิกจะแสดงผลในบิตแมพ (Bit Map) คือสร้างเป็นภาพใดๆก็ได้ตามต้องการ แนวทางในการใช้งานของทั้ง 2 แบบต่างก็มีลักษณะใกล้เคียงกัน การใช้งานโดยทั่วไปมักจะเป็นแบบด็อทเมทริกซ์มากกว่า เนื่องจากราคาถูกกว่าและเพียงพอต่องานส่วนใหญ่

คุณสมบัติของ ด็อทเมทริกซ์แอลซีดีโมดูล และสามารถสรุปได้เป็นข้อๆดังนี้

- ตัวอักษรแสดงด้วยด็อทเมทริกซ์ขนาด 5 x 8 ด็อท
- สามารถต่อเข้ากับระบบ ไมโครคอนโทรลเลอร์ได้ 2 ลักษณะ คือแบบเมโมรีแมพ และแบบเมโมรีผ่านพอร์ตซึ่งจะใช้ขาสัญญาณทั้งหมด 14 ขา (ขั้วต่อ 16 ขา)
- การใช้งานสะดวกกับระบบไมโครคอนโทรลเลอร์ เพียงแค่ส่งข้อมูลให้กับแอลซีดีโมดูล เท่านั้นข้อความก็จะปรากฏบนแผงแสดงและจะค้างไว้ตลอดทำให้ไม่ต้องเสียเวลาของระบบ
- มีคำสั่งพิเศษสำหรับอำนวยความสะดวกมากมาย เช่น Clear Display, Home Cursor on, Off Cursor, Blank Characteristic ฯลฯ
- สามารถแสดงผลตัวอักษรภาษาอังกฤษและตัวเลขได้ 160 ตัวและสัญลักษณ์พิเศษอีก 32 ตัว
- กินกระแสน้อยและมีน้ำหนักเบา รวมทั้งทำงานได้ด้วยไฟเลี้ยงระดับ 5 โวลต์

2.2.5 การต่อเข้ากับระบบไมโครคอนโทรลเลอร์

แอลซีดีโมดูลจะต่อเข้ากับระบบ ไมโครคอนโทรลเลอร์ ได้ 2 ลักษณะ ดังได้กล่าวไปแล้วซึ่งทั้ง 2 แบบนี้จะมีข้อดีข้อเสียต่างกันออกไป โดยแต่ละแบบมีหลักการดังนี้

การต่อแบบเมโมรีแมพ(Memory Map)

- สามารถต่อเข้ากับชิพเบอร์ทั่วๆ ไปได้ เช่น 8051 หรือ Z80 โดยจะทำให้ระบบที่ใช้ตัวไมโครคอนโทรลเลอร์สามารถมองเห็นแอลซีดีโมดูลในลักษณะเมโมรีได้ทันที
- ผู้ใช้สามารถเขียนและอ่านข้อมูลจากแอลซีดีได้ ซึ่งจะทำให้มองเสมือนเป็นเมโมรีบัฟเฟอร์(Memory Buffer) ไปในตัว

- เนื่องจากสามารถอ่านข้อมูลกลับได้จึงทำให้สามารถตรวจสอบแฟลก ความพร้อมในขณะที่แอลซีดีโมดูลกำลังทำงานได้
- ใช้ได้กับบอร์ดที่มีแอลซีดีบัสมาให้พร้อมเท่านั้น
- ทำให้กินพื้นที่ของหน่วยความจำไปส่วนหนึ่งและต้องมีการดีโค๊ดละเอียดพอสมควร
- การจัดขาสัญญาณจะต้องเป็นไปตามแบบของชิพ

2.2.6 แบบไอโอพอร์ท (I/O Port)

- สามารถต่อเข้ากับไอโอพอร์ทใดๆก็ได้โดยใช้สายสัญญาณจำนวน 11 เส้นและใช้โปรแกรมเป็นตัวสร้างสัญญาณขึ้นมา ให้ตรงกับข้อกำหนดของแอลซีดีโมดูล
- ผู้ใช้เขียนข้อมูลให้แอลซีดีโมดูลอย่างเดียว ซึ่งผู้ใช้ควรจะกำหนดเมโมรีส่วนหนึ่งให้เสมือนบัฟเฟอร์ให้กับแอลซีดีโมดูล
- เนื่องจากไม่สามารถอ่านข้อมูลกลับได้ จึงต้องใช้การหน่วงเวลาของระบบเอง เพื่อรอให้แอลซีดีโมดูลกระทำขบวนการต่างๆ
- ใช้ได้กับบอร์ดทั่วๆไปที่มีพอร์ท
- ไม่เปลืองส่วนของเมโมรีในการใช้งาน

2.2.7 คำสั่งการควบคุมจอ LCD

ในการเขียนคำสั่งลงในตัวควบคุมนั้น ต้องทำการกำหนดค่าขา RS และ R/W เป็น "0" แล้วจึงเขียนคำสั่งตามไป คำสั่งที่ทำการควบคุมโมดูล LCD ของชิปควบคุม HD44780 ที่สำคัญๆ นั้นมีดังต่อไปนี้

1. คำสั่งเคลียร์ตัวแสดงผล (clear display)

มีข้อมูลคำสั่งเป็น 01H เป็นคำสั่งที่ใช่เขียนข้อมูลช่องว่างหรือ space เข้าไปใน DDRAM ทั้งหมดเมื่อตัวควบคุมทำการ execute คำสั่งนี้ จะทำการกำหนดแอดเดรสของ DDRAM เป็น 0 เคอร์เซอร์จะกลับไปอยู่ที่ตำแหน่งซ้ายมือสุดของจอแสดงผล แล้วเซตบิต I/D ให้เป็น "1"

2. คำสั่ง Return Home

ต้องกำหนดให้บิต 1 ของข้อมูลเป็น "1" เป็นคำสั่งให้เคอร์เซอร์เคลื่อนที่กลับไปยังตำแหน่งซ้ายสุดของจอแสดงผล แต่ข้อมูลที่อยู่บนจอแสดงผลอยู่นั้นจะไม่มีการเปลี่ยนแปลง นั่นคือ ข้อมูลคำสั่งของคำสั่งนี้จะเป็น 02H หรือ 03H ก็ได้

3. คำสั่งเลือกโหมดการป้อนข้อมูล (Entry mode Set)

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
0	0	0	0	0	1	I/D	S

บิต S เป็นบิตที่ใช้ในการกำหนดลักษณะของการแสดงผล เมื่อมีการป้อนข้อมูล ถ้าหากบิต S เป็น"1" เมื่อเกิดข้อมูลใหม่บนจอแสดงผล ตัวเคอร์เซอร์จะอยู่กับที่ แต่ตัวอักษรข้อมูลเดิมจะถูกดันไปทางซ้าย แต่ถ้าเป็น "0" เมื่อเกิดข้อมูลใหม่ตัวของเคอร์เซอร์จะเลื่อนไปทางขวามือ

บิต I/D เป็นบิตที่ใช้ในการกำหนดว่า เมื่อมีการเขียนหรืออ่านข้อมูลแล้ว ทำให้แอดเดรสของDDRAM เพิ่มขึ้นหรือลดลง 1 แอดเดรส โดยถ้าบิตนี้เป็น "1" แอดเดรสของ DDRAM จะเพิ่มขึ้น แต่ถ้าเป็น "0" แอดเดรสจะลดลง

ดังนั้นข้อมูลคำสั่งที่เกิดขึ้นนี้สำหรับคำสั่งนี้ได้แก่ 04H-07H และที่นิยมใช้คือ 06H หมายถึงกำหนดให้เมื่อมีการเกิดข้อมูลใหม่ เคอร์เซอร์จะเลื่อนไปทางขวามือ และแอดเดรสของ DDRAM จะมีการเพิ่มขึ้น

4. คำสั่งควบคุมการแสดงผล

มีรายละเอียดของคำสั่งดังต่อไปนี้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
0	0	0	0	1	D	C	B

บิต D ใช้ควบคุมการเปิดปิดจอแสดงผล ถ้าเป็น "1" จะเป็นการเปิดจอแสดงผล แต่ถ้าเป็น "0" จะเป็นการปิดจอแสดงผล

บิต C ใช้ควบคุมการแสดงตัวเคอร์เซอร์บนจอแสดงผล ถ้าต้องการให้มีเคอร์เซอร์แสดงอยู่บนจอก็กำหนดเป็น "1" แต่ถ้าไม่ต้องการก็ทำการกำหนดบิตนี้เป็น "0"

บิต B ใช้ควบคุมการกระพริบของเคอร์เซอร์ ถ้าต้องการให้มีการกระพริบก็ทำการกำหนดบิตนี้เป็น "1"

ดังนั้นข้อมูลคำสั่งนี้จะมีค่าคำสั่งตั้งแต่ 08H-0FH แต่ที่นิยมใช้ คือ คำสั่ง 0CH เป็นการให้มีการเปิดจอแสดงผล แต่ไม่มีการแสดงเคอร์เซอร์และ0FH เป็นการเปิดจอแสดงผล แสดงเคอร์เซอร์รวมทั้งมีการกระพริบด้วย

5. คำสั่งควบคุมการเลื่อนเคอร์เซอร์และข้อมูลตัวอักษร

มีรายละเอียดของคำสั่งดังต่อไปนี้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
0	0	0	1	S/C	R/L	*	*

การควบคุมเคอร์เซอร์และการควบคุมตัวอักษรบนจอแสดงผลจะขึ้นอยู่กับการกำหนดบิต S/C และ R/L ซึ่งจะมีลักษณะดังต่อไปนี้

S/C	R/L	ลักษณะการเลื่อน	ข้อมูลคำสั่ง
0	0	เลื่อนเคอร์เซอร์ไปทางซ้าย	10H-13H
0	1	เลื่อนเคอร์เซอร์ไปทางขวา	14H-17H
1	0	เลื่อนตัวอักษรใหม่ไปทางซ้าย	18H-1BH
1	1	เลื่อนตัวอักษรใหม่ไปทางขวา	1CH-1FH

6. คำสั่งกำหนดฟังก์ชั่นการทำงาน

มีรายละเอียดของคำสั่งดังต่อไปนี้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
0	0	1	D/L	N	F	*	*

บิต DL เป็นบิตที่ใช้เพื่อทำการกำหนดจำนวนบิตที่ใช้ในการส่งข้อมูลเมื่อ เป็น “0” จะเป็นการติดต่อแบบ 4 บิต แต่ถ้าเป็น “1” จะเป็นการติดต่อแบบ 8 บิต

บิต N เป็นการกำหนดบรรทัดของการแสดงข้อมูลโดย “0” เป็นการแสดงแบบ 1 บรรทัดของจอLCD ขนาด 1 บรรทัด แต่ถ้าจอ LCD เป็นประเภทที่มากกว่า 1 บรรทัดก็กำหนดให้เป็น “1”

บิต F ใช้เลือกความละเอียดของตัวอักษรในการแสดงผลโดย “0” เป็นการแสดงขนาน 5x7 จุด “1” เป็นการแสดงขนาด 5x10 จุด

7. คำสั่งเลือกแอดเดรสของ CGRAM

เมื่อต้องการกำหนดแอดเดรสของ CGRAM ต้องกำหนดบิต 7 เป็น “0” บิต 6 เป็น “1” ส่วนอีก 6 บิตที่เหลือจะเป็นค่าของแอดเดรสโดยแอดเดรสจะอยู่ระหว่างค่า 00H-3FH

8. คำสั่งเลือกแอดเดรสของ DDRAM

ใช้ในการเลือกแอดเดรสของ DDRAM ก่อนที่จะทำการอ่านหรือเขียนข้อมูล โดยบิตที่ 7 ต้องเป็น “1” และข้อมูลอีก 7 บิตที่เหลือจะเป็นค่าแอดเดรสของ DDRAM โดยจะอยู่ระหว่าง 8CH-0FFH โดยจำนวนแอดเดรสยังขึ้นอยู่กับสถานะของบิต N โดยที่ถ้าบิต N เป็น “1” แอดเดรสจะมีค่าอยู่ระหว่าง 80H-0CFH และถ้าเป็น “1” จะมีค่า2ช่วงอยู่ระหว่าง 8CH-87H และ 0C0H-0C7H

9. คำสั่งอ่านแฟลก BUSY และแอดเดรส

มีรายละเอียดของคำสั่งดังต่อไปนี้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
BF	A	A	A	A	A	A	A

เป็นคำสั่งที่ใช้ในการอ่านแฟลก BUSY (BF) โดยแฟลกนี้จะเป็นตัวที่บอกสถานะของจอ LCD ว่าพร้อมที่จะรับคำสั่งหรือข้อมูลหรือไม่โดยถ้าบิต BF เป็น "0" แสดงว่า LCD พร้อมที่จะรับข้อมูลหรือคำสั่งแต่ถ้าเป็น "1" นั่นหมายความว่า LCD กำลังทำงานใดงานหนึ่งอยู่ยังไม่พร้อมที่จะรับคำสั่งไปทำงาน

2.2.8 การเขียนคำสั่งและข้อมูลให้แก่จอ LCD

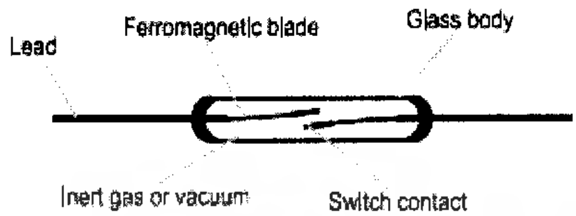
ในการเขียนข้อมูลเพื่อควบคุมให้แก่ LCD นั้นต้องทำการส่งคำสั่งในการ instruction แล้วทำการกำหนดโหมดการทำงานให้แก่โมดูล LCD ก่อน จากนั้นจึงค่อยทำการส่งข้อมูลที่ต้องการแสดงผลเนื่องจากบัสข้อมูลของ LCD นั้นมี 8 เส้นที่ใช้เป็นตัวส่งข้อมูลและคำสั่ง ดังนั้นในการส่งคำสั่งและข้อมูลจึงต้องทำการเซตลอจิกที่ขา RS ว่าถ้าเป็น "0" จะเป็นการส่งข้อมูลที่เป็นคำสั่ง แ ละถ้าเป็น"1" นั่นหมายความว่าข้อมูลนั้นเป็นข้อมูลที่ต้องการให้แสดงผลออกทางจอ LCD

เมื่อต้องการเขียนข้อมูลหรืออ่านข้อมูลใน CGRAM และ DDRAM เริ่มจากต้องทำการกำหนดแอดเดรสที่ต้องการอ่านหรือเขียนก่อน โดยใช้คำสั่งเลือกแอดเดรส จากนั้นกำหนดให้ขา RS เป็น "1" เพื่อแจ้งให้ตัวควบคุมภายใน LCD รู้ว่าข้อมูลนี้เป็นข้อมูลที่ไม่ใช่คำสั่ง

ในกรณีที่ต้องการอ่านข้อมูลต้องกำหนดให้ขา R/W เป็น "1" ข้อมูลขนาด 8 บิตก็จะอยู่บนบัสข้อมูลโดยที่ข้อมูลที่อ่านออกมานั้นจะเป็นข้อมูลที่มาจากแอดเดรสของ CGRAM หรือ DDRAM ตามที่ต้องการ

ในกรณีที่ต้องการเขียนข้อมูล เมื่อกำหนดแอดเดรสและทำการเซตลอจิกที่ขา RS เป็น "1" แล้วนั้นต้องทำการกำหนดขา R/W เป็น "0" ข้อมูลที่อยู่บนบัสจะถูกเขียนในรีจิสเตอร์ DR จากนั้นจะเป็นการถ่ายข้อมูลไปที่ DDRAM เป็นลำดับต่อไป

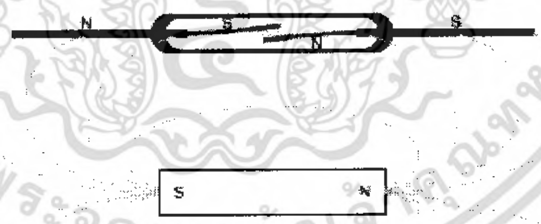
2.3 Reed switch



รูปที่ 2.8 แสดงส่วนประกอบของ Reed switch

Reed switch ประกอบด้วยโลหะ 2 ชิ้น (โดยทั่วไปจะประกอบไปด้วยธาตุเหล็กและนิเกิล) ที่มีลักษณะคล้ายกับใบมีดบางๆ วางอยู่ใกล้ๆ กัน ซึ่งบรรจุอยู่ภายในหลอดแก้วเล็กๆ ที่ด้านในประกอบด้วยก๊าซไนโตรเจน หรือบางครั้งอาจจะเป็นก๊าซเฉื่อย

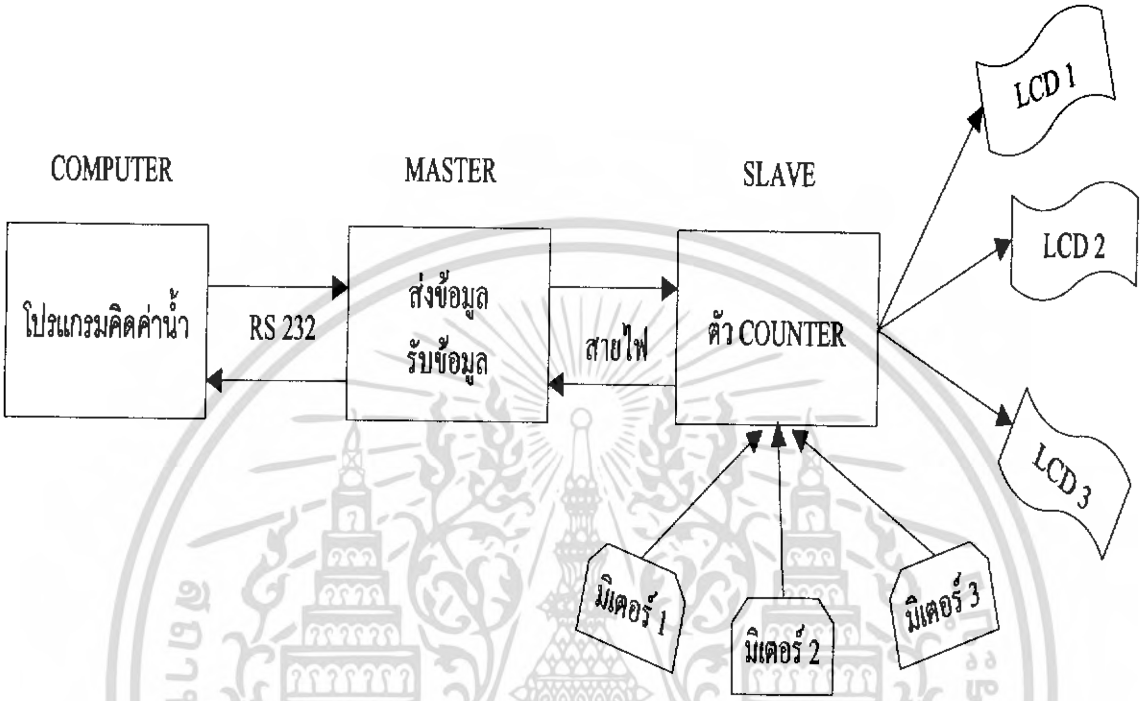
Reed switch จะทำงานก็ต่อเมื่อมีการนำแท่งแม่เหล็กถาวรเข้ามาใกล้ โดยสนามแม่เหล็กจะส่งผลทำให้โลหะทั้งสองชิ้นที่อยู่ภายในดูดกันทำให้โลหะทั้ง 2 ชิ้นนี้มีการแตะกัน เมื่อโละหะทั้ง 2 ชิ้นแตะกัน Switch จะทำงานแบบ ON เมื่อนำแท่งแม่เหล็กห่างออกไปโลหะ 2 ชิ้นนี้ก็จะแยกจากกัน Switch จะทำงานแบบ OFF ทันที ซึ่งสามารถแสดงลักษณะการทำงานได้ดังรูป



รูปที่ 2.9 แสดงการทำงานของ Reed switch เมื่อมีแท่งแม่เหล็กเข้ามาใกล้

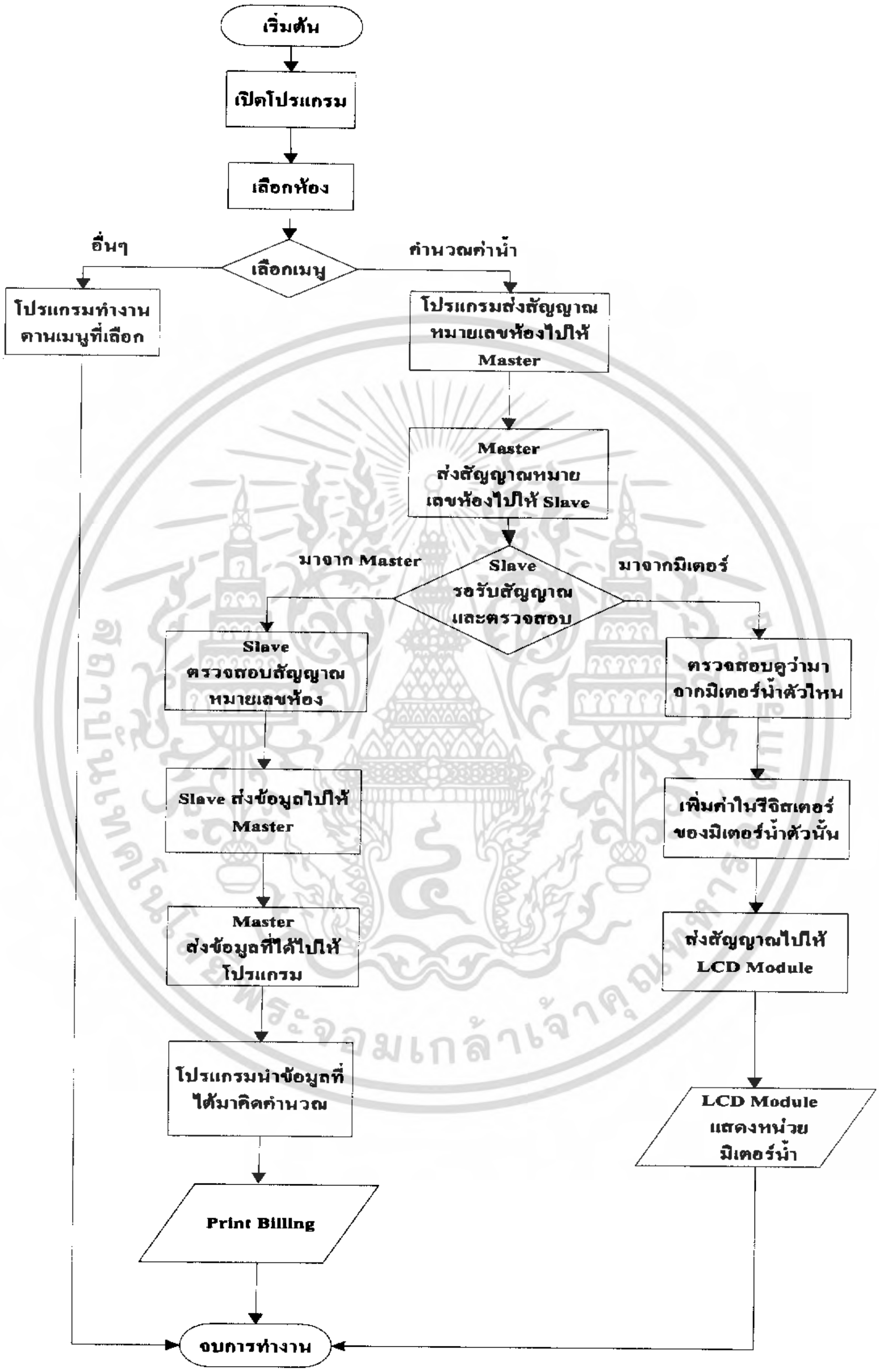
บทที่ 3
การคำนวณและการสร้าง

ในการทำงานของโครงงานนี้สามารถแสดงได้ตามบล็อกไดอะแกรมต่อไปนี้



รูปที่ 3.1 แสดงบล็อกไดอะแกรมการทำงานรวมของระบบ

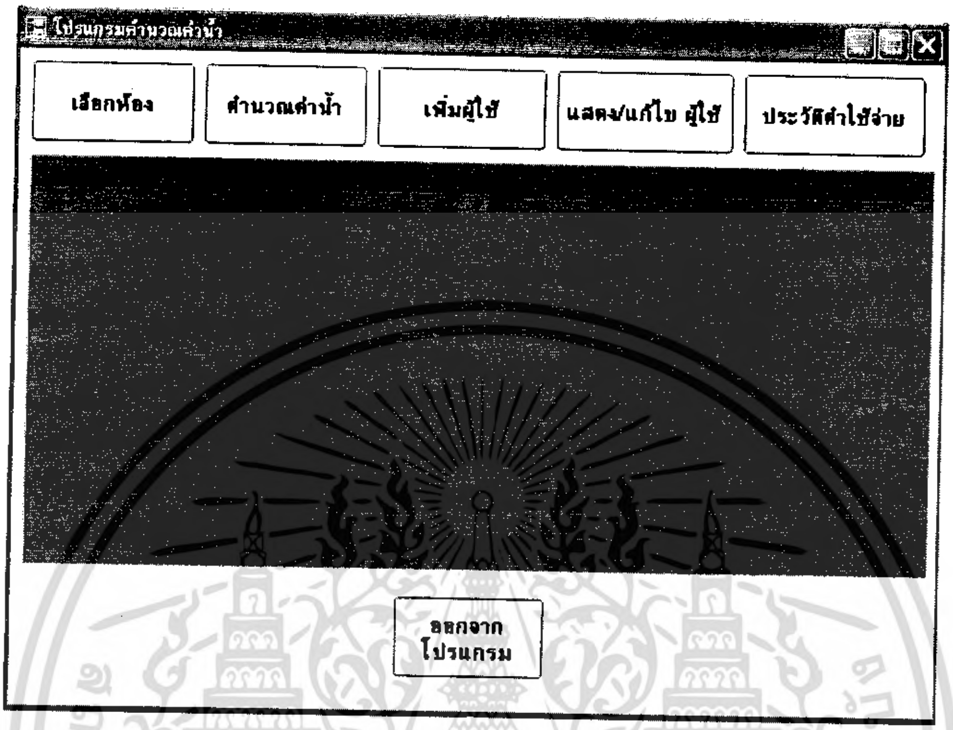
แผนผังการทำงานรวมของระบบ



รูปที่ 3.2 แสดงแผนผังการทำงานรวมของระบบ

3.1 โปรแกรมการคิดค่าน้ำ

หน้าที่หลักของโปรแกรมนี้คือ ทำการคิดค่าน้ำของแต่ละห้องตามความต้องการของเจ้าของหอพัก และทำการเก็บข้อมูลดังกล่าวไว้เป็นฐานข้อมูลการใช้ โดยโปรแกรมที่ใช้มีรูปร่างของหน้าต่าง ดังนี้



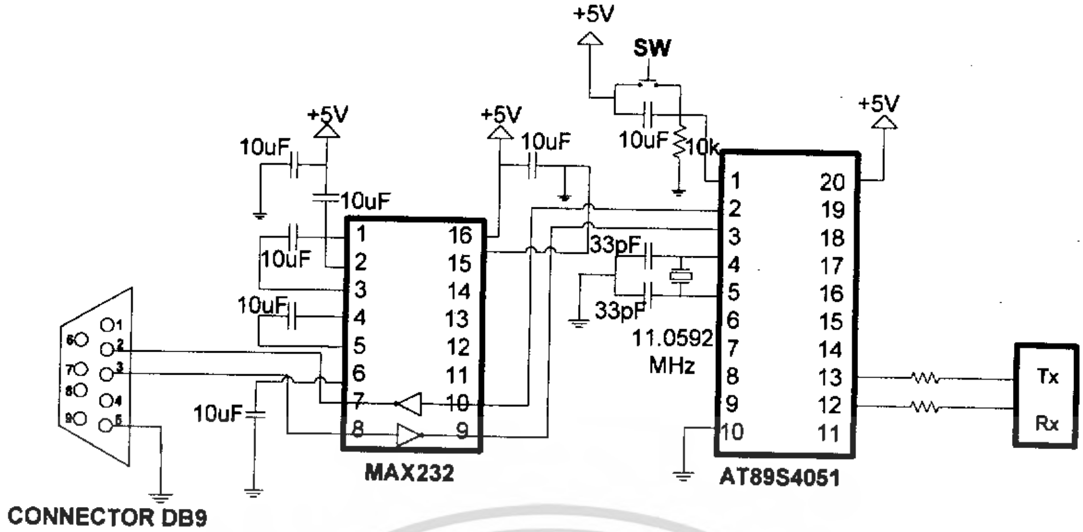
รูปที่ 3.3 แสดงหน้าหลักของโปรแกรม

จากรูปที่ 3.3 เมื่อเปิดโปรแกรมการคิดค่าน้ำขึ้นมา โปรแกรมการคิดค่าน้ำจะประกอบไปด้วยฟังก์ชั่นการทำงานต่าง ๆ ดังนี้

- เลือกห้อง : เป็นฟังก์ชั่นแรกที่ต้องกด เพื่อใส่หมายเลขห้องที่ต้องการคิดค่าน้ำ
- คำนวณค่าน้ำ : เป็นฟังก์ชั่นที่จะทำการคิดค่าน้ำ
- เพิ่มผู้ใช้ : เป็นฟังก์ชั่นที่ใช้ในกรณีที่มีการเข้ามาอยู่ใหม่ของผู้เช่าหรือมีการย้ายออก
- แสดง/แก้ไข ผู้ใช้ : เป็นการเรียกดูข้อมูลของผู้เช่าในห้องนั้น ๆ
- ประวัติค่าใช้จ่าย : เป็นฟังก์ชั่นการเรียกดูข้อมูลการใช้น้ำของห้องนั้น ๆ ตามที่ได้ทำการเลือกไว้

3.2 Master

เป็นส่วนที่คอยติดต่อกับคอมพิวเตอร์โดยจะทำหน้าที่ในการส่งสัญญาณร้องขอข้อมูลการใช้น้ำของห้องพักต่าง ๆ ตามสัญญาณที่รับมาจากโปรแกรมที่ใช้ในคอมพิวเตอร์ไปให้กับ Slave และจะคอยรับสัญญาณตอบกลับของสัญญาณข้อมูลที่ร้องขอไปจาก Slave ด้วย โดยจะมีการต่อวงจรดังรูป

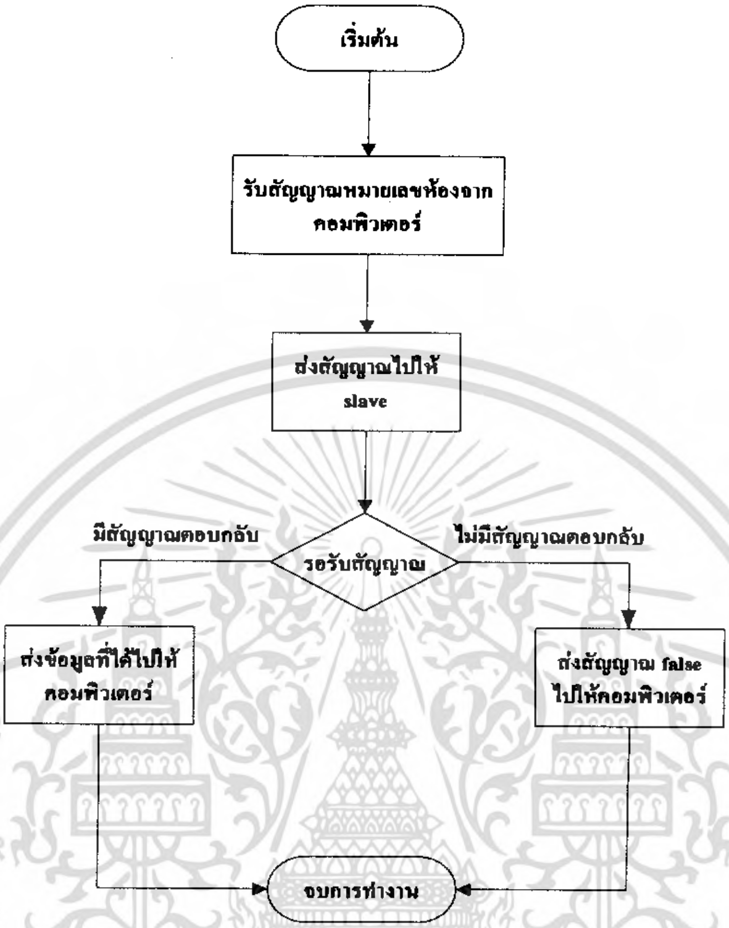


รูปที่ 3.4 แสดงรูปวงจรรวมทางด้านส่ง (Master)

โดยมีขั้นตอนการทำงานดังนี้

- เริ่มต้นการทำงานของวงจร วงจรทางด้าน Master จะอยู่ในสถานการณ์ทำงานแบบ T_x หรือเป็นวงจรที่ทำหน้าที่ในการส่งสัญญาณ
- ตัวโปรแกรมการทำงานของ Master จะคอยทำการตรวจเช็คสัญญาณว่ามีสัญญาณร้องขอข้อมูลการใช้น้ำของห้องพักจากโปรแกรมในคอมพิวเตอร์ โดยจะคอยทำการตรวจเช็คอยู่ตลอดเวลา
- ถ้ามีสัญญาณสัญญาณดังกล่าวจะเป็นสัญญาณ Address หรือหมายเลขของห้องที่ต้องการข้อมูล จากนั้น Master ก็จะทำการส่งสัญญาณดังกล่าวนี้ไปให้กับ Slave
- หลังจากนั้น Master ก็จะเปลี่ยนสถานภาพการทำงานของวงจรเป็นแบบ R_x ทันทีเพื่อรอรับสัญญาณตอบกลับข้อมูลจาก Slave
- เมื่อ Master รับสัญญาณตอบกลับได้แล้วก็จะทำการส่งข้อมูลดังกล่าวไปให้กับโปรแกรมในคอมพิวเตอร์ทันที
- แต่ถ้าหากไม่มีสัญญาณตอบกลับมาจาก Slave หรือสัญญาณมาไม่ถึง Master ก็จะส่งสัญญาณไปที่โปรแกรมในคอมพิวเตอร์ว่าไม่มีสัญญาณตอบกลับจาก Slave
- หลังจากที่ Master รับสัญญาณมาและทำการส่งสัญญาณไปที่โปรแกรมในคอมพิวเตอร์แล้ว ก็จะทำการเปลี่ยนสถานะการทำงานของวงจรเป็น T_x ทันที เพื่อจะคอยรับสัญญาณร้องขอข้อมูลเพื่อที่จะส่งอีกครั้ง

3.2.1 แผนผังการทำงานของทางด้าน Master

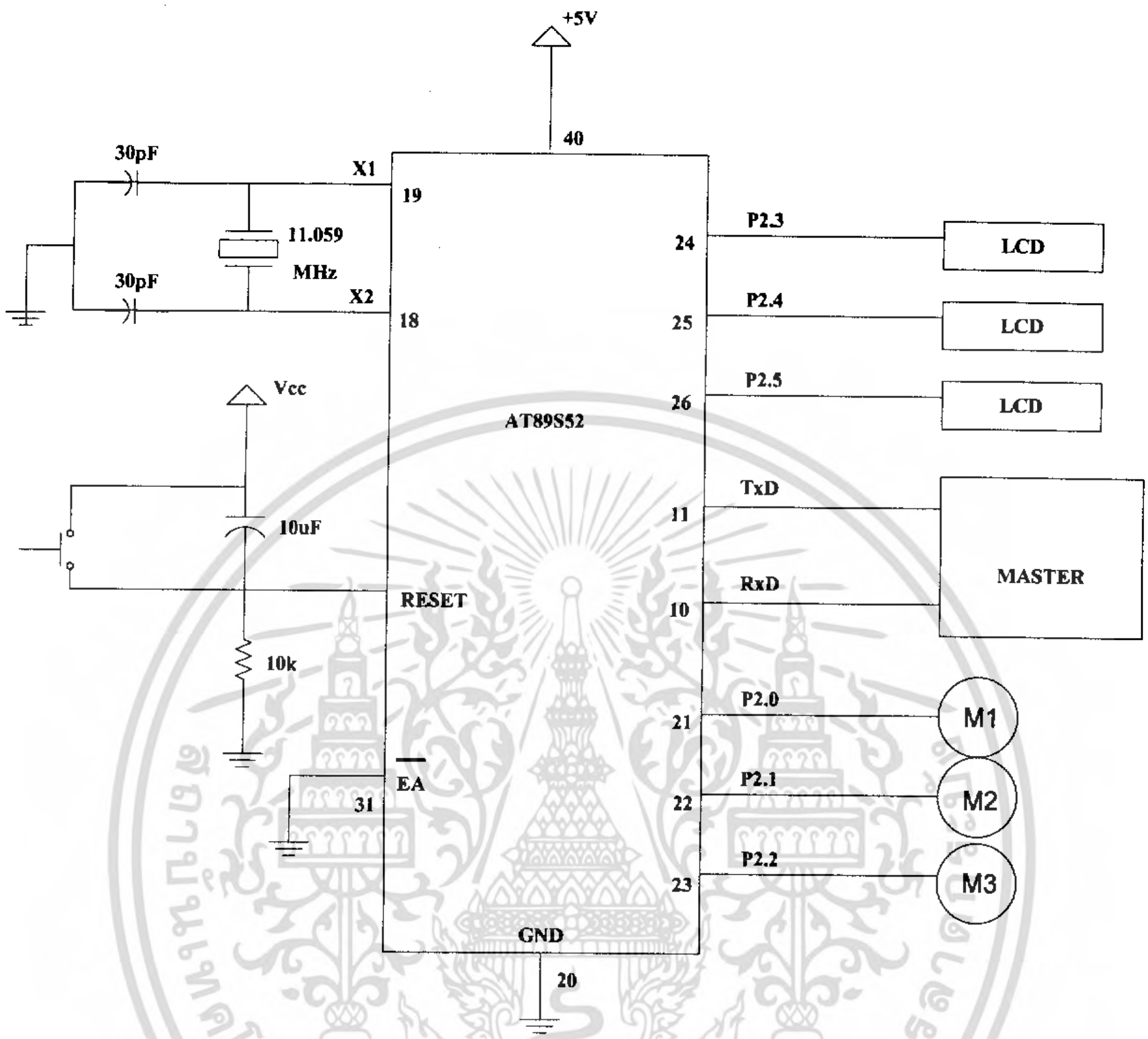


รูปที่ 3.5 แสดงแผนผังการทำงานของทางด้าน Master

3.3 Slave

เป็นวงจรที่ต่ออยู่กับมิเตอร์น้ำของแต่ละห้องซึ่ง Slave 1 ตัวสามารถที่จะทำการต่อกับมิเตอร์น้ำได้หลายตัว การทำงานของวงจรนี้จะเป็นตัวที่ทำการบันทึกข้อมูลการใช้น้ำของแต่ละห้องเอาไว้และคอยส่งข้อมูลดังกล่าวไปให้กับ Master เมื่อได้รับสัญญาณร้องขอมา โดยวงจรทางด้าน Slave นี้มีด้วยกัน 2 ส่วน คือ ส่วนแรกเป็นส่วนที่ทำหน้าที่เก็บข้อมูลการใช้น้ำทั้งหมดของมิเตอร์น้ำแต่ละตัว ส่วนที่ 2 คือ ส่วนที่ทำการแสดงผลการใช้น้ำของผู้ใช้น้ำในแต่ละห้องโดยจะแสดงผ่านจอ LCD โมดูล ซึ่งจะติดไว้ตามห้องของผู้ใช้

3.3.1 ส่วนของวงจรนับจำนวนการใช้น้ำของมิเตอร์



รูปที่ 3.6 แสดง MCS-51 บัสเมื่อติดต่อกับหน่วยความจำภายนอก

การทำงาน

ในส่วนของตัวนับ (Counter mode) ในโครงงานนี้ จะทำการสั่งงานวงจรโดยใช้โปรแกรมเป็นตัวสั่งการทำงาน ซึ่งจะมีหน้าที่หลักๆ อยู่ 3 ภาคด้วยกันดังนี้

ภาคที่ 1 : ภาคการนับจำนวนพัลส์ของสัญญาณที่เข้ามาทีละตัว โดยนับจากตัวเซ็นเซอร์ที่ติดและทำการนับค่าอยู่ที่มิเตอร์วัดน้ำ และส่งสัญญาณเป็นจำนวนพัลส์เข้ามายังพอร์ท P2.0, P2.1 และ P2.2 ตามลำดับ

ในส่วนของภาคที่ 1 นี้ ได้ทำการเขียนโปรแกรมนับจำนวนพัลส์ที่เข้ามาจากมิเตอร์ทั้ง 3 ตัว ทีละลูก เมื่อมีสัญญาณจากเซ็นเซอร์เข้ามาทีละลูก มายังส่วนของวงจรนับที่มีการโปรแกรมไว้และจะส่งสัญญาณนี้ไปแสดงผลยังจอ LCD โมดูลในแต่ละห้องของผู้เช่าในทันที

สัญญาณที่นับได้จากมิเตอร์ตัวที่ 1 จะถูกส่งมายัง P2.3 ซึ่งจะเข้ามาทางขาที่ 24 ของไมโครคอนโทรลเลอร์

สัญญาณที่นับได้จากมิเตอร์ตัวที่ 2 จะถูกส่งมายัง P2.4 ซึ่งจะเข้ามาทางขาที่ 25 ของไมโครคอนโทรลเลอร์

สัญญาณที่นับได้จากมิเตอร์ตัวที่ 3 จะถูกส่งมายัง P2.5 ซึ่งจะเข้ามาทางขาที่ 26 ของไมโครคอนโทรลเลอร์

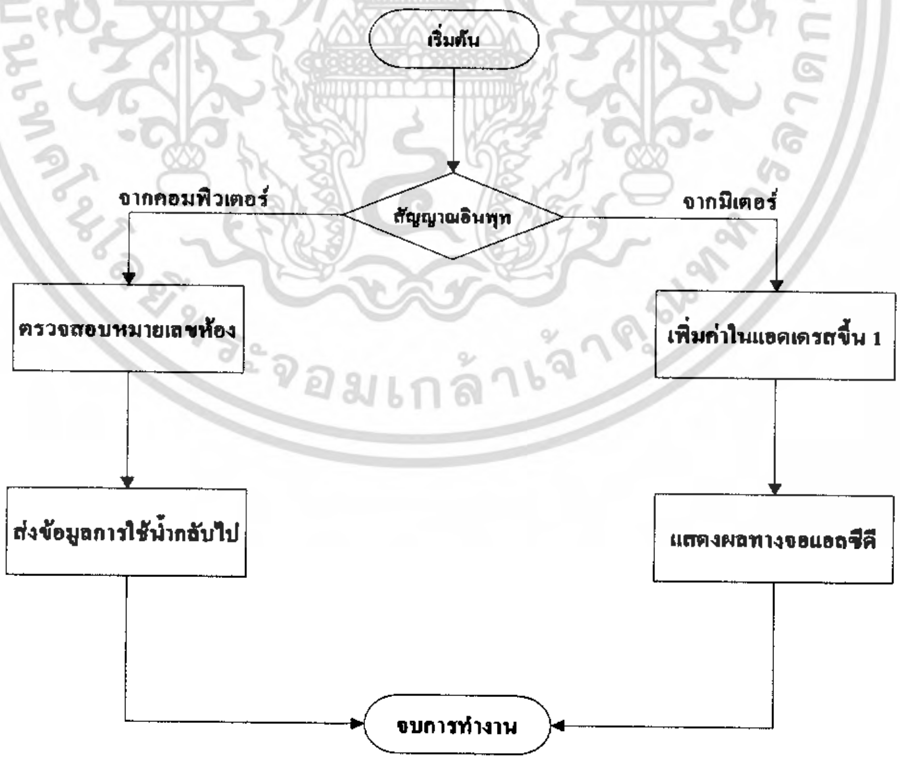
ภาคที่ 2 : เป็นภาคที่ทำงานต่อจากภาคที่ 1 นั่นคือ เมื่อวงจรนับได้รับสัญญาณพัลส์แล้ว โปรแกรมคำสั่งที่เขียนคือจะให้ส่งค่าสัญญาณนี้ไปแสดงผลออกทางจอ LCD โมดูล โดยที่สัญญาณพัลส์ที่ได้จากมิเตอร์วัดน้ำทั้ง 3 ตัวนี้ จะถูกส่งมาอย่างอิสระต่อกัน นั่นคือถูกส่งมาแบบแยกแอดเดรสของข้อมูลกัน ดังนั้นเมื่อสัญญาณพัลส์ออกจากภาคที่ 2 ไปแล้ว ก็จะมีไมโครคอนโทรลเลอร์คอยควบคุมการแสดงผลอีก 3 ตัวเช่นเดียวกัน

ภาคที่ 3 : เป็นภาคการทำงานแรกสุดและท้ายสุดในส่วนของโปรแกรมนับ ซึ่งมีการทำงาน 2 แบบประกอบด้วย 2 ส่วนหลักๆ คือ

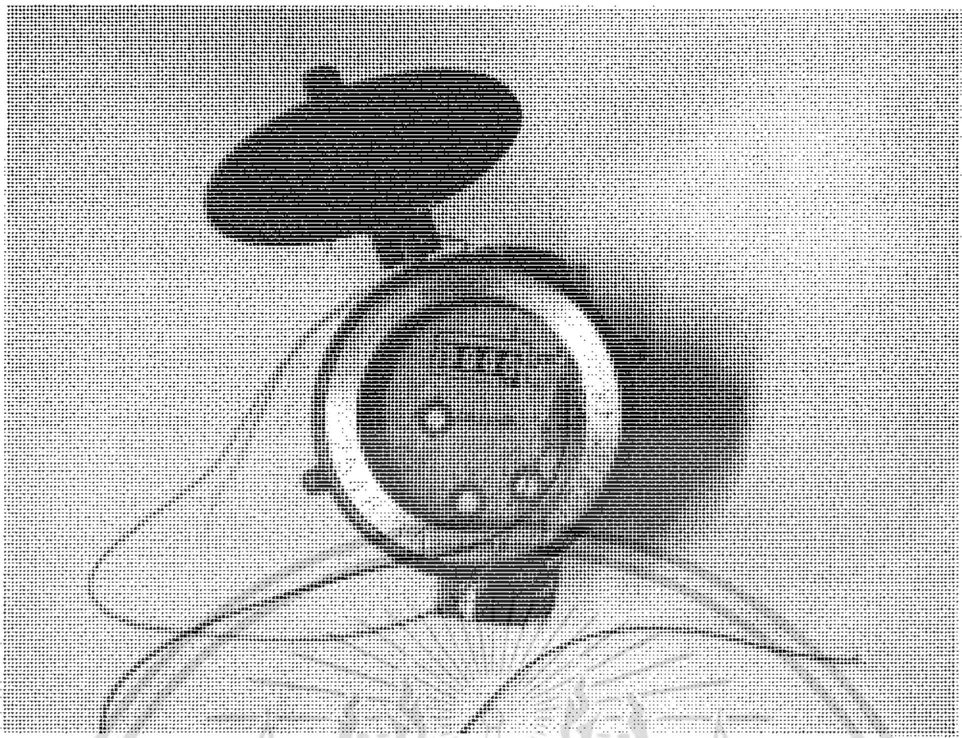
ส่วนแรกในด้านของภาครับ RxD ของโปรแกรมนับทางด้านรับสัญญาณจะเรียกใช้งานโปรแกรมจากไมโครคอนโทรลเลอร์ตัวกลาง ซึ่งสัญญาณจะถูกส่งมาและเข้ามายังขาที่ 10 ของไมโครคอนโทรลเลอร์

ส่วนที่สอง คือด้านของภาคส่ง TxD เมื่อโปรแกรมนับทำงานเสร็จสมบูรณ์ทั้งในภาคที่ 1 และภาคที่ 2 จะทำการส่งข้อมูลการทำงานทั้งหมดกลับออกไปยังไมโครคอนโทรลเลอร์ตัวกลาง ผ่านทางขาที่ 11ของไมโครคอนโทรลเลอร์

3.3.2 แผนผังการทำงานของทางด้าน Slave

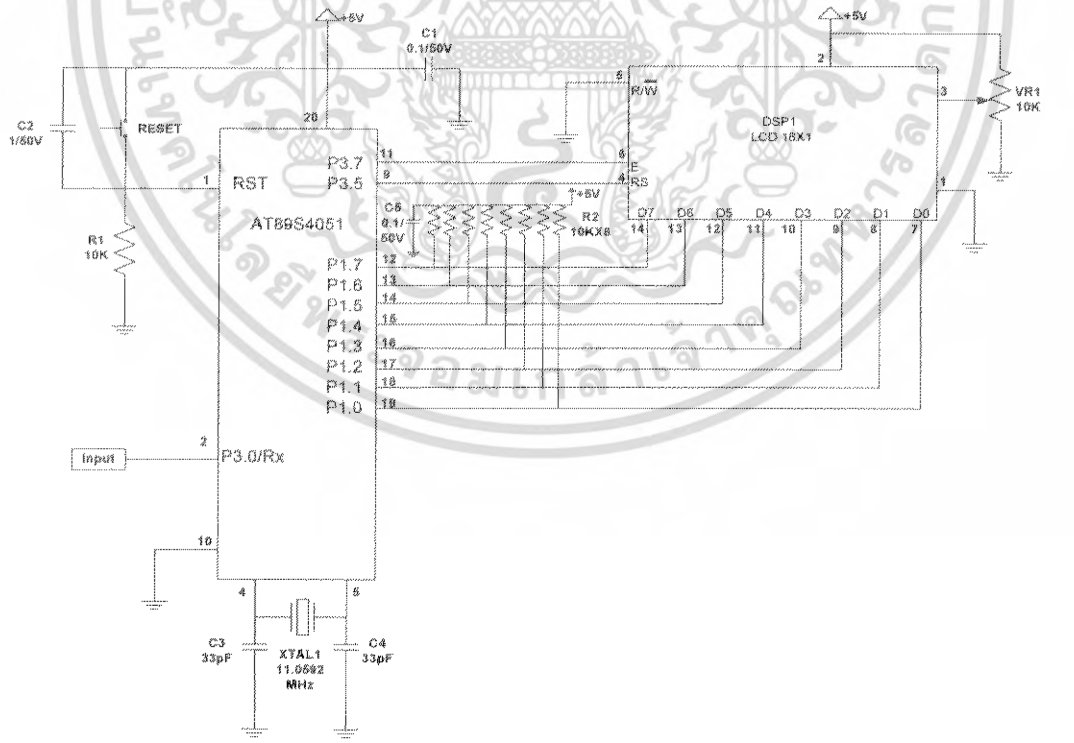


รูปที่ 3.7 แสดงแผนผังการทำงานของทางด้าน Slave



รูปที่ 3.8 แสดงมิเตอร์น้ำที่ทำการติดตั้ง Reed Switch แล้ว

3.3.3 วงจรส่วนแสดงผล (LCD module)



รูปที่ 3.9 แสดงวงจรส่วนแสดงผล (LCD module) ปริมาณการใช้น้ำในแต่ละหน่วย

การทำงาน

เป็นวงจรส่วนที่ติดต่ออยู่กับวงจรทางด้านฝั่ง Slave เพื่อทำการแสดงผลไปยังห้องของผู้เช่าในแต่ละห้องว่ามีการใช้น้ำไปปริมาณเท่าไร ในส่วนของวงจรประกอบไปด้วยจอ LCD และ ไมโครคอนโทรลเลอร์ที่เป็นตัวรับข้อมูลมาจากวงจรทางด้านฝั่ง Slave แล้วมาทำการควบคุมการทำงานของโมดูล LCD

ในการทำงานของวงจรจะใช้ Mcs-51 ในการควบคุมการทำงานทั้งหมดและแสดงผลการทำงานที่ได้ผ่านทางส่วนแสดงผล คือ โมดูล LCD โดยวงจรส่วนแสดงผลนี้จะเป็นส่วนที่คอยรับข้อมูลปริมาณการใช้น้ำจากวงจรทางด้านฝั่ง Slave ที่ส่งมาให้ผ่านตามสายไฟ โดย 1 พัลส์ที่ส่งมาจะแสดงถึงการใช้น้ำ 1 ลูกบาศก์เมตร หรือ 1 ยูนิต ซึ่งในการแสดงผลนั้นเมื่อวงจรทางด้านฝั่ง Slave ส่งข้อมูลมา 1 ครั้ง วงจรทางด้านส่วนแสดงผลจะมีการแสดงผลบนจอโมดูล LCD 1 ครั้ง จนกว่าวงจรทางด้านฝั่ง Slave จะมีการส่งข้อมูลมาอีกครั้ง ข้อมูลที่แสดงผลจึงจะมีการเปลี่ยนแปลงไป 1 หน่วย

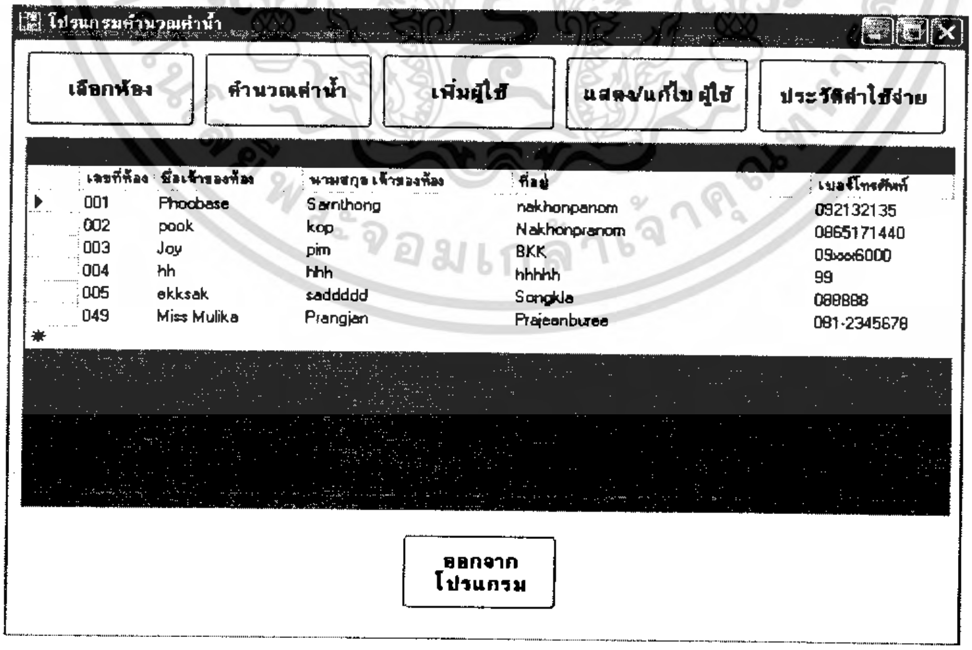
บทที่ 4
การทดลองและผลการทดลอง

การทำงานของโครงงานนี้จะแบ่งออกเป็น 3 ส่วน คือ ส่วนของ Master , ส่วนของ Slave และส่วนของการแสดงผลทางโมดูล LCD ซึ่งมีการทำงานและผลการทำงานดังนี้

ส่วนของ Master

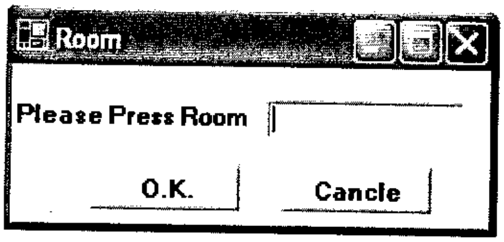
1. เริ่มต้นการทำงานจะทำการเรียกใช้โปรแกรมคิดค่าน้ำในคอมพิวเตอร์

รูปที่ 4.1 แสดงหน้าต่างโปรแกรมเมื่อเข้าไปในโปรแกรม



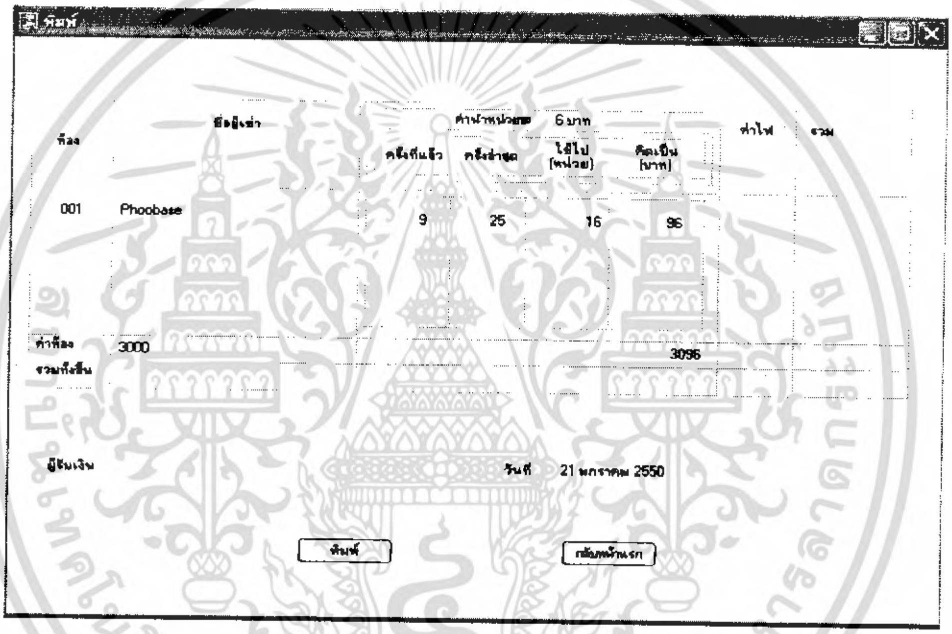
รูปที่ 4.2 แสดงหน้าต่างหลักของโปรแกรม

2. หลังจากที่เข้าโปรแกรมไปแล้ว จะต้องทำการเลือกห้องที่จะทำการคิดค่าน้ำ



รูปที่ 4.3 แสดงหน้าต่างที่จะใส่หมายเลขห้อง

3. เมื่อเลือกห้องแล้วก็จะให้โปรแกรมทำการคิดค่าน้ำ ซึ่งโปรแกรมก็จะส่งสัญญาณไปให้กับ Master เพื่อที่จะส่งสัญญาณดังกล่าวไปให้กลับ Slave



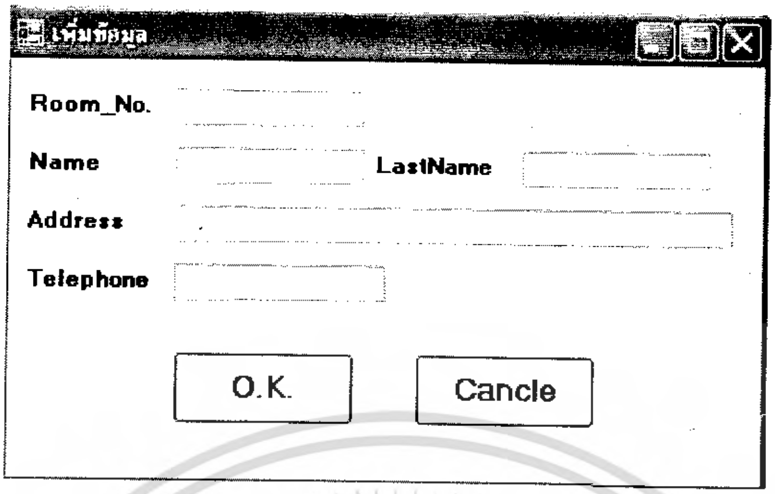
รูปที่ 4.4 แสดงหน้าต่างการคิดค่าน้ำ

ห้อง	ชื่อผู้เช่า	ค่าน้ำหน่วยละ 6.00 บาท				ค่าไฟ	รวม (บาท)
		ครั้งที่แล้ว	ครั้งล่าสุด	ใช้ไป (หน่วย)	คิดเป็น (บาท)		
001	Phoobase	9	25	16.00	96.00		
ค่าห้อง	3,000.00				3,096.00		
รวมทั้งสิ้น							

ผู้รับเงิน .... วันที่

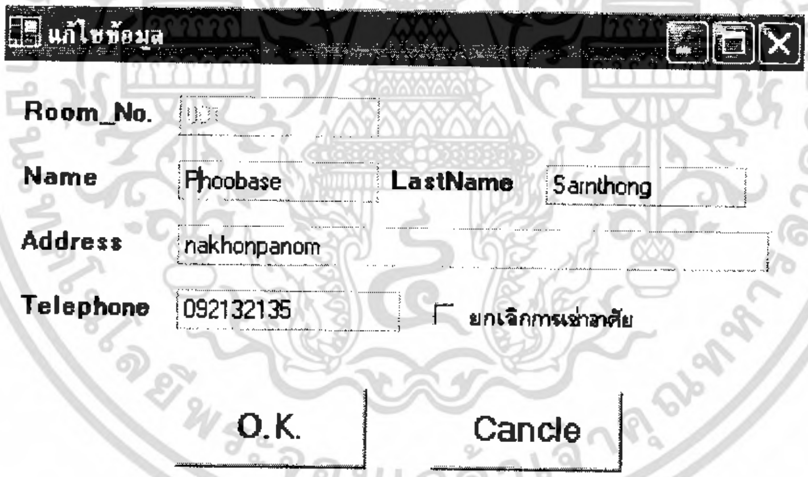
รูปที่ 4.5 แสดงใบเสร็จที่ทำการพิมพ์แล้ว

3.1 หรือสามารถเข้าเมนูอื่น ๆ นอกจาก เมนูการคิดค่าน้ำได้ ดังนี้



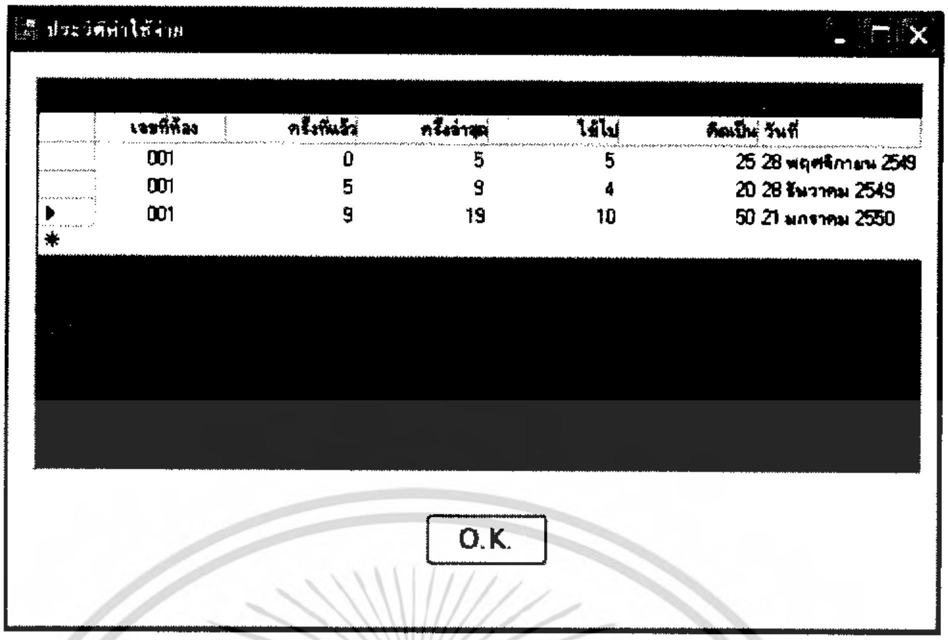
รูปที่ 4.6 แสดงหน้าต่างที่ใช้ในการเพิ่มข้อมูลของผู้เช่าห้อง

จากรูปที่ 4.6 หน้าต่างที่ใช้ในการเพิ่มข้อมูลของผู้เช่าห้อง โดยสามารถที่จะทำการเพิ่มข้อมูลของผู้เช่าห้องในหน้านี้ได้ ตามข้อมูลของผู้เช่า



รูปที่ 4.7 แสดงหน้าต่างที่ใช้ในการแก้ไขข้อมูลของผู้เช่าห้อง

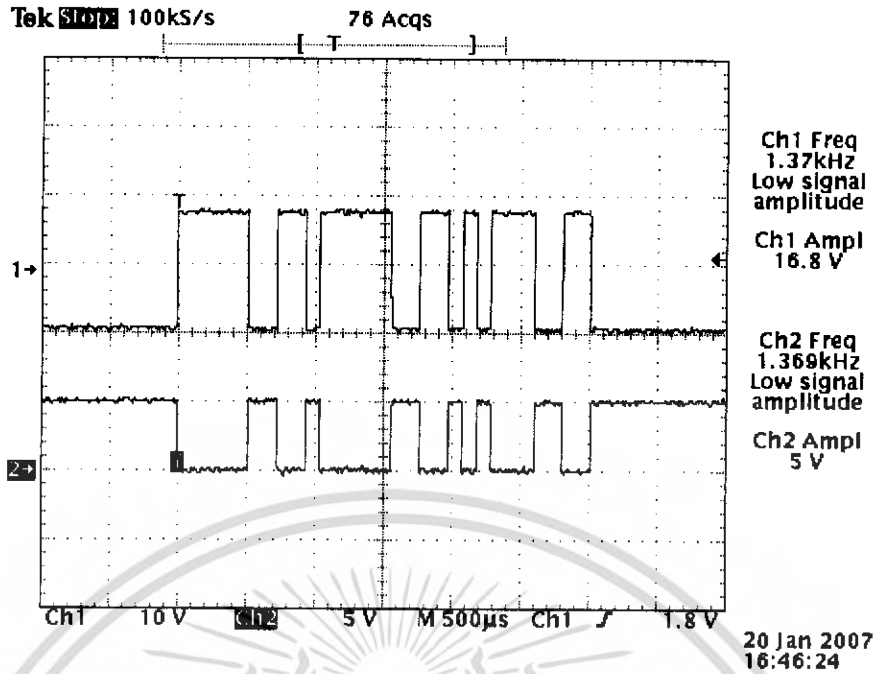
จากรูปที่ 4.7 เป็นหน้าต่างที่คล้ายกับหน้าต่างที่ใช้ในการเพิ่มข้อมูลของผู้เช่าห้อง ใช้ในการเปลี่ยนแปลงและแก้ไขข้อมูลของผู้เช่าห้อง



รูปที่ 4.8 แสดงประวัติการใช้น้ำของห้องพักในแต่ละห้อง

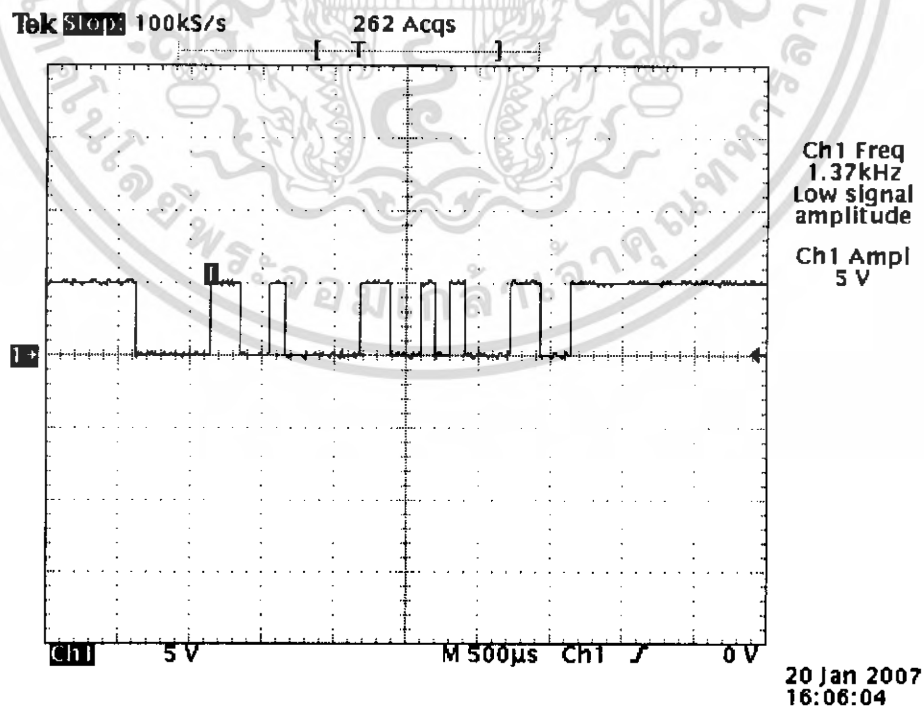
จากรูปที่ 4.8 เป็นเมนูที่ใช้ในการดูข้อมูลการใช้น้ำย้อนหลังของการใช้น้ำในแต่ละห้อง ซึ่งสามารถเข้ามาดูเพื่อใช้เป็นข้อมูลในการคิดค่าน้ำในกรณีที่ กระบวนการคิดค่าน้ำเกิดการผิดพลาดที่โปรแกรมไม่สามารถที่จะทำการเรียกข้อมูลได้

4. หลังจากที่ทำการเลือกเมนูให้คิดค่าน้ำแล้วนั้น ก็จะมีข้อมูลของหมายเลขห้องที่ต้องการติดต่อด้วยส่งมาให้ที่ Master เพื่อให้ Master ส่งสัญญาณดังกล่าวไปให้กับ Slave เพื่อร้องขอข้อมูลการใช้น้ำ ณ ขณะนั้นของห้องนั้น ๆ มาให้ก็จะได้รูปของสัญญาณหมายเลขห้องที่ส่งออกมาจากคอมพิวเตอร์และสัญญาณที่ได้ก็จะผ่านไปที่ IC MAX 232 ดังนี้

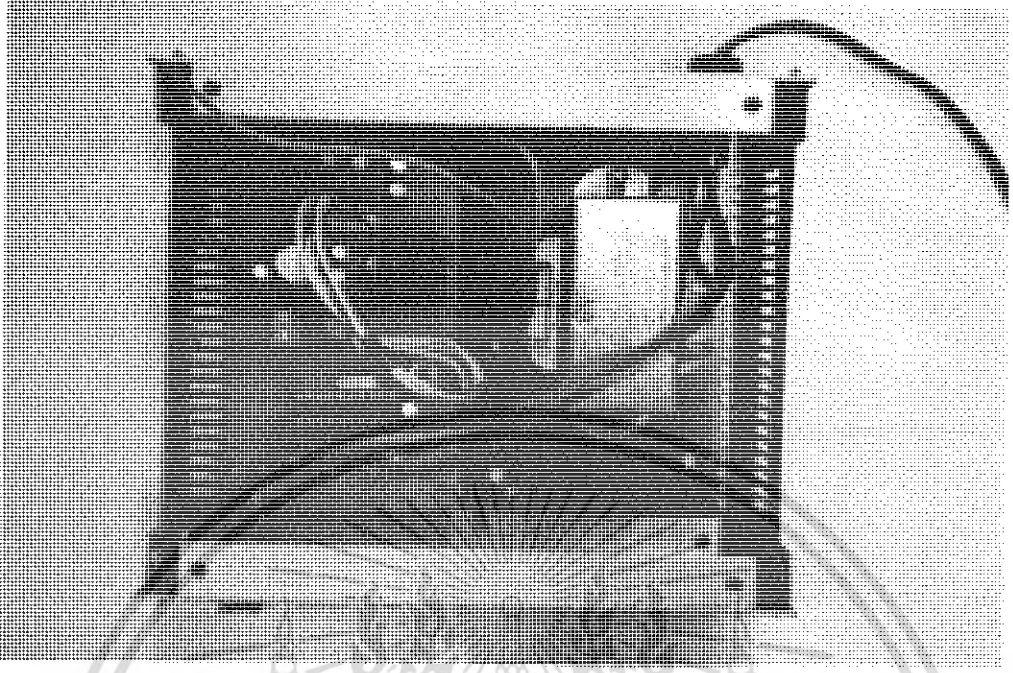


รูปที่ 4.9 แสดงสัญญาณหมายเลขห้อง (001) ที่ส่งออกมาจากคอมพิวเตอร์ (บน)
สัญญาณหมายเลขห้องที่ผ่าน IC MAX 232 (ล่าง)

จากรูปที่ 4.9 จะเห็นว่าสัญญาณที่ได้จากคอมพิวเตอร์จะเป็นสัญญาณไฟฟ้าขนาด 9 โวลต์ แต่เมื่อผ่านไปที่ตัว IC MAX 232 แล้วสัญญาณไฟฟ้าจะลดลงเหลือเพียง 5 โวลต์ เพื่อให้เป็นสัญญาณที่ได้สอดคล้องกับระบบไมโครคอนโทรลเลอร์ที่ใช้



รูปที่ 4.10 แสดงสัญญาณหมายเลขห้อง (001) ที่ส่งออกมาจากฝั่ง Master ไปให้ทางด้านฝั่ง Slave



รูปที่ 4.11 แสดงวงจรรวมทางด้าน Master

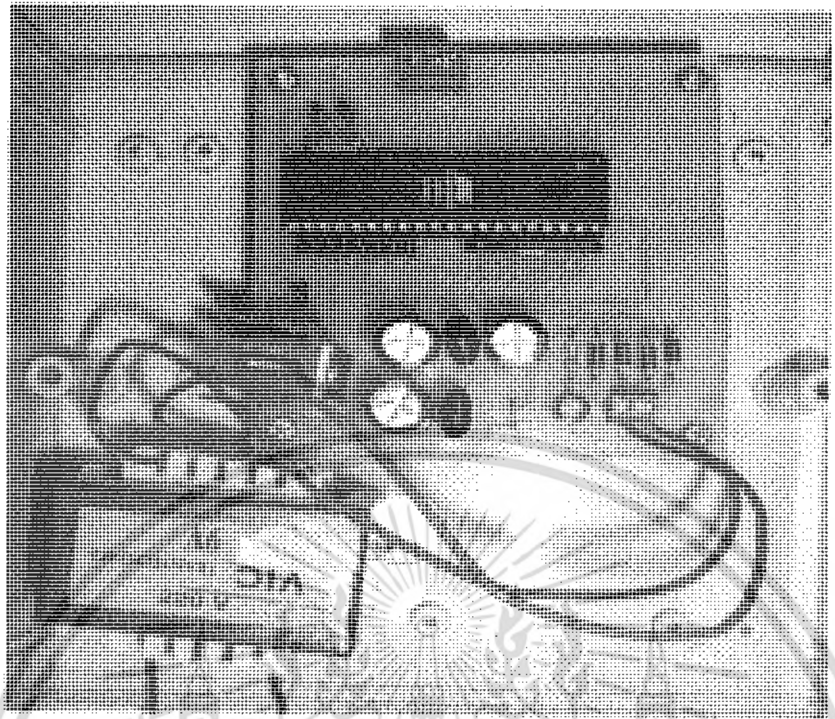
ส่วนของ Slave

1. ทางด้านการทำงานของ Slave นั้นจะเปิดการทำงานอยู่ตลอดเวลาเพื่อที่จะทำการบันทึกข้อมูลการใช้น้ำของแต่ละห้อง และคอยตรวจสอบสัญญาณติดต่อจาก Master

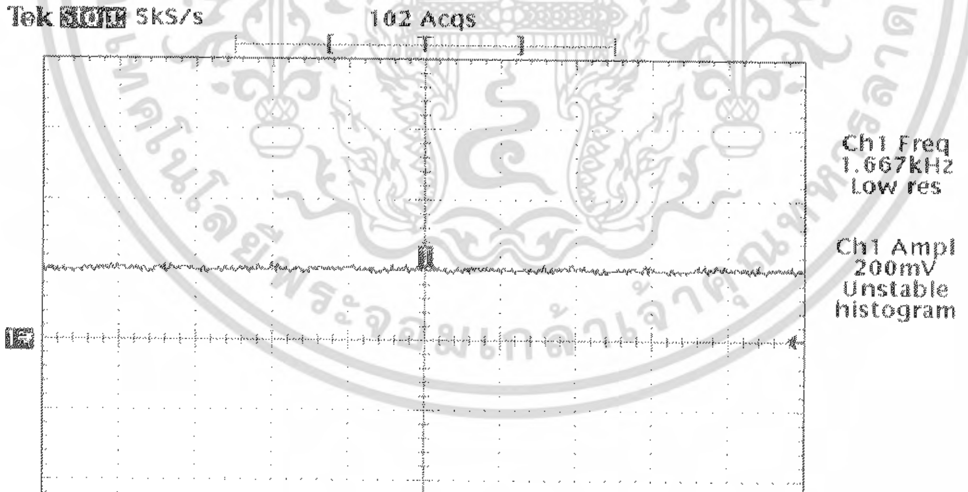
2. เมื่อมีสัญญาณอินพุทมาจากทางด้าน Reed switch ส่วนของวงจร Mcs-51 ก็จะทำการเพิ่มค่าในหน่วยความจำขึ้นอีก 1 ค่าที่ Reed switch นั้นต่อเป็นอินพุทอยู่

3. ในกรณีที่ทาง Slave ทำการตรวจพบว่ามีสัญญาณติดต่อมาจาก Master ก็จะทำการตรวจสอบว่าสัญญาณที่ส่งมานั้นเป็นสัญญาณของหมายเลขห้องใดที่ทาง Master ต้องการข้อมูลกลับไป เพื่อที่ทาง Slave จะได้ทำการส่งข้อมูลได้ไปถูกต้องตามความต้องการ

4. เมื่อ Slave ทำการตรวจทราบแล้วว่าทาง Master ต้องการข้อมูลของห้องใดแล้วก็จะทำการส่งข้อมูลของห้องนั้น ๆ ไปให้กับ Master ตามความต้องการ และส่งข้อมูลไปแสดงผลตามแต่ละห้องของผู้เช่าได้



รูปที่ 4.12 แสดงวงจรรวมทางด้าน Slave



รูปที่ 4.13 แสดงสัญญาณอินพุทของวงจรนับจาก Reed switch

ส่วนของการแสดงผล (LCD module)

เมื่อทางด้านฝั่ง Slave ได้รับข้อมูลการใช้น้ำในแต่ละห้องแล้ว ขณะเดียวกัน ฝั่ง Slave ก็จะทำการส่งข้อมูลที่ได้นั่นไปแสดงผลที่จอโมดูล LCD ในแต่ละห้องทันที

รูปที่ 4.14 แสดงส่วนแสดงผลของ LCD โมดูล

บทที่ 5
สรุปและวิจารณ์ผลการทดลอง

สรุป

หลังจากที่ได้ทำการศึกษาหัวข้อโครงงานจนเข้าใจหลักการทำงานและขอบเขตการทำงานทั้งหมดของอุปกรณ์ แล้วจึงเริ่มแบ่งแยกการทำงาน ซึ่งจะแบ่งได้ดังนี้

1. ส่วนของ Master ที่ต่ออยู่กับคอมพิวเตอร์
2. ส่วนของ Slave ที่ต่ออยู่กับมิเตอร์น้ำ
3. ส่วนของเซ็นเซอร์หน่วยวัดมิเตอร์น้ำ
4. ส่วนแสดงผลด้วยโมดูล LCD
5. ส่วนโปรแกรมในคอมพิวเตอร์

เมื่อแบ่งแยกส่วนย่อยได้แล้ว จึงเขียนไดอะแกรมแสดงการเชื่อมต่อในแต่ละส่วน หลังจากนั้นจึงเขียนแผนผังแสดงขั้นตอนการทำงานตามลำดับตั้งแต่การส่งสัญญาณร้องขอจากโปรแกรมในคอมพิวเตอร์ในส่วนของ Master ไปยังส่วนของ Slave แล้วอ่านข้อมูลที่ต้องการส่งกลับไปยัง Master เพื่อนำข้อมูลที่ได้ไปประมวลผลคิดเป็นค่าใช้จ่ายของผู้เช่าในคอมพิวเตอร์ ผลที่ได้มีดังนี้

1. โปรแกรมที่ใช้สามารถทำงานได้ตามที่ต้องการ ทำให้สะดวกต่อระบบการจัดเก็บข้อมูลได้ดี
2. การใช้งานของไมโครคอนโทรลเลอร์สามารถที่จะช่วยควบคุมให้ระบบมีการทำงานที่ดี
3. สามารถนำระบบที่ได้ไปใช้ในการคิดค่าน้ำได้จริง

ปัญหาและแนวทางการพัฒนาต่อ

โครงงานนี้เป็นการศึกษาการประยุกต์ใช้ไมโครคอนโทรลเลอร์, เซ็นเซอร์ และมิเตอร์น้ำ เพื่อนำมาจัดสร้างระบบกระบวนการคิดคำนวณหน่วยมิเตอร์น้ำ โดยได้มีการใช้โปรแกรม Visual Studio C#.net เพื่อมาช่วยในกระบวนการดังกล่าวรวมทั้งในส่วนของการจัดเก็บข้อมูลด้วย แต่ก็ยังคงมีปัญหาในส่วนของระบบเซ็นเซอร์ที่จะทำการตรวจจับการหมุนของตัวมิเตอร์น้ำ ในที่นี้ได้ใช้มิเตอร์น้ำที่เป็นระบบอะนาล็อกที่มีขายอยู่ทั่วไปตามท้องตลาด ดังนั้นจึงต้องมีการแก้ไขในตัวมิเตอร์น้ำบางส่วนเพื่อให้เซ็นเซอร์ที่ใช้สามารถทำการตรวจจับการหมุนของตัวมิเตอร์น้ำได้ ในส่วนของการแก้ไขตัวมิเตอร์น้ำนี้เป็นการแก้ไขในส่วนของหน้าปัดของมิเตอร์น้ำโดยไม่ได้ทำการเปลี่ยนแปลงในส่วนของกลไกการทำงานของมิเตอร์น้ำแต่อย่างใด ดังนั้นจึงทำให้การนับจำนวนหน่วยการใช้น้ำยังคงมีความถูกต้องและแม่นยำตามมาตรฐานเดิม แต่อาจจะมีความยุ่งยากในการแก้ไขบ้างเล็กน้อย

บรรณานุกรม

1. ปรเมษฐ์ ประนยานันท์ และ ปิยพงษ์ เผ่าวนิช "คู่มือและการประยุกต์ใช้งานไมโครคอนโทรเลอร์ MCS – 51" กรุงเทพมหานคร : ซีเอ็ดยูเคชั่น , 2536
2. ผศ.ปัญญาพล หอระตะ. "หลักการเขียนโปรแกรมภาษา C"กรุงเทพมหานคร : ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยขอนแก่น,2545
3. รัชนัย อินทุใส และ ไตรภพ อินทุใส. "ไมโครคอนโทรลเลอร์ 8051"กรุงเทพมหานคร : ฟิสิกส์เซ็นเตอร์ , 2537
4. สุนทร วิฑูสุรพจน์. "การใช้งานไมโครคอนโทรลเลอร์ตระกูล 8051" กรุงเทพมหานคร : ซีเอ็ดยูเคชั่น , 2536
5. รองศาสตราจารย์ สมยศ จุณณะปิยะ. การประยุกต์ใช้งานไมโครคอนโทรลเลอร์ MICROCONTROLLER APPLICATION. ภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ภาคผนวก

```
############################ Addmember ################################
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;
using System.Data.SqlClient;

namespace WindowsApplication1
{
	/// <summary>
	/// Summary description for AddMember.
	/// </summary>
	public class AddMember : System.Windows.Forms.Form
	{
		private System.Windows.Forms.Label label1;
		private System.Windows.Forms.Label label2;
		private System.Windows.Forms.Label label3;
		private System.Windows.Forms.Label label4;
		private System.Windows.Forms.Label label5;
		private System.Windows.Forms.Button btnOK;
		private System.Windows.Forms.Button btnCancle;
		public System.Windows.Forms.TextBox txRoom;
		private System.Windows.Forms.TextBox txName;
		private System.Windows.Forms.TextBox txLname;
		private System.Windows.Forms.TextBox txTel;
		private System.Windows.Forms.TextBox txAddress;
		/// <summary>
		/// Required designer variable.
		/// </summary>
		private System.ComponentModel.Container components = null;
		decimal			deRecord	= 0 ;
		SqlConnection myConnection ;
		private System.Windows.Forms.CheckBox chkActive;
		DataSet ds ;

		public AddMember()
		{
			//
			// Required for Windows Form Designer support
			//
			InitializeComponent();

			//
			// TODO: Add any constructor code after
InitializeComponent call
			//
		}

		/// <summary>
		/// Clean up any resources being used.
		/// </summary>
		protected override void Dispose( bool disposing )
		{
			if( disposing )
			{
```

```
                    if(components != null)
                    {
                          components.Dispose();
                    }
              }
              base.Dispose( disposing );
        }

        #region Windows Form Designer generated code
        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
              this.label1 = new System.Windows.Forms.Label();
              this.txRoom = new System.Windows.Forms.TextBox();
              this.label2 = new System.Windows.Forms.Label();
              this.label3 = new System.Windows.Forms.Label();
              this.txName = new System.Windows.Forms.TextBox();
              this.txLname = new System.Windows.Forms.TextBox();
              this.txAddress = new System.Windows.Forms.TextBox();
              this.label4 = new System.Windows.Forms.Label();
              this.txTel = new System.Windows.Forms.TextBox();
              this.label5 = new System.Windows.Forms.Label();
              this.btnOK = new System.Windows.Forms.Button();
              this.btnCancle = new System.Windows.Forms.Button();
              this.chkActive = new System.Windows.Forms.CheckBox();
              this.SuspendLayout();
              //
              // label1
              //
              this.label1.Font = new System.Drawing.Font("Microsoft
Sans Serif", 9F, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((System.Byte)(0)));
              this.label1.Location = new System.Drawing.Point(8,
16);
              this.label1.Name = "label1";
              this.label1.Size = new System.Drawing.Size(72, 23);
              this.label1.TabIndex = 0;
              this.label1.Text = "Room_No.";
              //
              // txRoom
              //
              this.txRoom.Location = new System.Drawing.Point(88,
16);
              this.txRoom.Name = "txRoom";
              this.txRoom.TabIndex = 1;
              this.txRoom.Text = "";
              //
              // label2
              //
              this.label2.Font = new System.Drawing.Font("Microsoft
Sans Serif", 9F, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((System.Byte)(0)));
              this.label2.Location = new System.Drawing.Point(8,
48);
```

```
            this.label2.Name = "label2";
            this.label2.Size = new System.Drawing.Size(72, 23);
            this.label2.TabIndex = 2;
            this.label2.Text = "Name";
            //
            // label3
            //
            this.label3.Font = new System.Drawing.Font("Microsoft
Sans Serif", 9F, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((System.Byte)(0)));
            this.label3.Location = new System.Drawing.Point(192,
48);
            this.label3.Name = "label3";
            this.label3.Size = new System.Drawing.Size(64, 16);
            this.label3.TabIndex = 3;
            this.label3.Text = "LastName";
            //
            // txName
            //
            this.txName.Location = new System.Drawing.Point(88,
48);
            this.txName.Name = "txName";
            this.txName.TabIndex = 4;
            this.txName.Text = "";
            //
            // txLname
            //
            this.txLname.Location = new System.Drawing.Point(272,
48);
            this.txLname.Name = "txLname";
            this.txLname.TabIndex = 5;
            this.txLname.Text = "";
            //
            // txAddress
            //
            this.txAddress.Location = new
System.Drawing.Point(88, 80);
            this.txAddress.Name = "txAddress";
            this.txAddress.Size = new System.Drawing.Size(296,
20);
            this.txAddress.TabIndex = 7;
            this.txAddress.Text = "";
            //
            // label4
            //
            this.label4.Font = new System.Drawing.Font("Microsoft
Sans Serif", 9F, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((System.Byte)(0)));
            this.label4.Location = new System.Drawing.Point(8,
80);
            this.label4.Name = "label4";
            this.label4.Size = new System.Drawing.Size(72, 23);
            this.label4.TabIndex = 6;
            this.label4.Text = "Address";
            //
            // txTel
            //
```

```
                this.txTel.Location = new System.Drawing.Point(88,
112);
                this.txTel.Name = "txTel";
                this.txTel.Size = new System.Drawing.Size(112, 20);
                this.txTel.TabIndex = 9;
                this.txTel.Text = "";
                //
                // label5
                //
                this.label5.Font = new System.Drawing.Font("Microsoft
Sans Serif", 9F, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((System.Byte)(0)));
                this.label5.Location = new System.Drawing.Point(8,
112);
                this.label5.Name = "label5";
                this.label5.Size = new System.Drawing.Size(72, 23);
                this.label5.TabIndex = 8;
                this.label5.Text = "Telephone";
                //
                // btnOK
                //
                this.btnOK.Font = new System.Drawing.Font("Microsoft
Sans Serif", 12F, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((System.Byte)(0)));
                this.btnOK.Location = new System.Drawing.Point(88,
160);
                this.btnOK.Name = "btnOK";
                this.btnOK.Size = new System.Drawing.Size(96, 40);
                this.btnOK.TabIndex = 10;
                this.btnOK.Text = "O.K.";
                this.btnOK.Click += new
System.EventHandler(this.btnOK_Click);
                //
                // btnCancle
                //
                this.btnCancle.Font = new
System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
((System.Byte)(0)));
                this.btnCancle.Location = new
System.Drawing.Point(216, 160);
                this.btnCancle.Name = "btnCancle";
                this.btnCancle.Size = new System.Drawing.Size(96,
40);
                this.btnCancle.TabIndex = 11;
                this.btnCancle.Text = "Cancle";
                this.btnCancle.Click += new
System.EventHandler(this.btnCancle_Click);
                //
                // chkActive
                //
                this.chkActive.Location = new
System.Drawing.Point(216, 112);
                this.chkActive.Name = "chkActive";
                this.chkActive.Size = new System.Drawing.Size(136,
24);
                this.chkActive.TabIndex = 12;
```

```
			this.chkActive.Text = "ยกเลิกการเช่าอาศัย";
			this.chkActive.Visible = false;
			//
			// AddMember
			//
			this.AutoScaleBaseSize = new System.Drawing.Size(5,
13);
			this.ClientSize = new System.Drawing.Size(400, 230);
			this.Controls.Add(this.chkActive);
			this.Controls.Add(this.btnCancle);
			this.Controls.Add(this.btnOK);
			this.Controls.Add(this.txTel);
			this.Controls.Add(this.txAddress);
			this.Controls.Add(this.txLname);
			this.Controls.Add(this.txName);
			this.Controls.Add(this.txRoom);
			this.Controls.Add(this.label5);
			this.Controls.Add(this.label4);
			this.Controls.Add(this.label3);
			this.Controls.Add(this.label2);
			this.Controls.Add(this.label1);
			this.Name = "AddMember";
			this.StartPosition =
System.Windows.Forms.FormStartPosition.CenterScreen;
			this.Text = "เพิ่มข้อมูล";
			this.Load += new
System.EventHandler(this.AddMember_Load);
			this.Activated +=new
EventHandler(AddMember_Activated);
			this.Closed += new
System.EventHandler(this.AddMember_Closed);
			this.ResumeLayout(false);

		}
		#endregion

		private void AddMember_Load(object sender, System.EventArgs
e)
		{
			myConnection = new
SqlConnection(DataClass.CommonConnection);

			if(DataClass.CommonRoom != "" )   //Check ว่าเป็นการแก้ไขหรือไม่
			{
				chkActive.Visible = true;
				this.Text = "แก้ไขข้อมูล";

				SqlDataAdapter SelectList     =     new
SqlDataAdapter("select name_member,lname_member,address,tel from member
where status =1 and rec_no_room ="+DataClass.CommonRoom,myConnection) ;
				ds = new DataSet() ;
				SelectList.Fill(ds,"listMember") ;
				//Assign Value
				txName.Text      =
	ds.Tables["listMember"].Rows[0][0].ToString() ;
```

```
                    txLname.Text        =
        ds.Tables["listMember"].Rows[0][1].ToString() ;
                    txAddress.Text      =
        ds.Tables["listMember"].Rows[0][2].ToString() ;
                    txTel.Text          =
        ds.Tables["listMember"].Rows[0][3].ToString() ;
                }
            }

            private void btnCancle_Click(object sender,
System.EventArgs e)
            {
                this.Owner.Show();
                this.Hide();
            }

            private void AddMember_Closed(object sender,
System.EventArgs e)
            {
                Application.Exit();
            }

            private void btnOK_Click(object sender, System.EventArgs e)
            {
                if(txRoom.Text  == "")          //Check Empty Room
                {
                    MessageBox.Show("กรุณากรอกเลขที่ห้องด้วยครับ") ;
                    txRoom.Focus() ;
                }
                else if(txName.Text  == "")  //Check Empty Name
                {
                    MessageBox.Show("กรุณากรอกชื่อด้วยครับ") ;
                    txName.Focus() ;
                }
                else if(txLname.Text == "")  //Check Empty Lname
                {
                    MessageBox.Show("กรุณากรอกนามสกุลด้วยครับ") ;
                    txLname.Focus() ;
                }
                else if(txAddress.Text == "") //Check Empty Address
                {
                    MessageBox.Show("กรุณากรอกที่อยู่ด้วยครับ") ;
                    txAddress.Focus() ;
                }
                else if(txTel.Text == "")   //Check Empty Tel
                {
                    MessageBox.Show("กรุณากรอกเบอร์โทรศัพท์ด้วยครับ") ;
                    txTel.Focus() ;
                }
                else
                {
                    if(DataClass.CommonRoom != "")
                        //Update Data
                        UpdateMember();
                    else
                    {
```

```
                    //Insert Data
                    SqlDataAdapter SelectCus = new
SqlDataAdapter("select Rec_no_room  from member where status = 1 and
rec_no_room ="+txRoom.Text,myConnection) ;
                    ds = new DataSet() ;
                    SelectCus.Fill(ds,"Generate") ;
                    if(ds.Tables["Generate"].Rows.Count == 0)
                    {
                         InsertMember();

                    }
                    else
                    {
                         MessageBox.Show("หมายเลขห้องนี้มีอยู่แล้วในระบบ");
                         txRoom.Clear();
                         txRoom.Focus();
                    }
               }
          }
     }

     private void InsertMember()
     {
          try
          {
               SqlDataAdapter SelectCus = new
SqlDataAdapter("select Max(Rec_no_member)  from member ",myConnection)
;
               ds = new DataSet() ;
               SelectCus.Fill(ds,"Generate") ;

               if(ds.Tables["Generate"].Rows[0][0] !=
System.DBNull.Value)              // Check status ID
               {
                    deRecord       =
     Convert.ToDecimal(ds.Tables["Generate"].Rows[0][0]) ;
                    deRecord       =      deRecord +1 ;
               }
               else
               {
                    deRecord       =      1 ;
               }

               string cmdInsertCus = "insert into
Member(Rec_no_member,Rec_no_room,Name_member,Lname_member,Address,Tel,s
tatus,Remark)"+

     "values(@Rec_no_member,@Rec_no_room,@Namemember,@Lname,@Address,@
Tel,@status,@Remark)";

               SqlCommand InsertCus     =      new
SqlCommand(cmdInsertCus,myConnection) ;

               InsertCus.Parameters.Add(new
SqlParameter("@Rec_no_member",SqlDbType.Int)) ;
               InsertCus.Parameters["@Rec_no_member"].Value =
Convert.ToInt32(deRecord) ;
```

```
                        InsertCus.Parameters.Add(new
SqlParameter("@Rec_no_room",SqlDbType.VarChar,10)) ;
                        InsertCus.Parameters["@Rec_no_room"].Value =
txRoom.Text;

                        InsertCus.Parameters.Add(new
SqlParameter("@Namemember",SqlDbType.VarChar,50)) ;
                        InsertCus.Parameters["@Namemember"].Value =
txName.Text ;

                        InsertCus.Parameters.Add(new
SqlParameter("@Lname",SqlDbType.VarChar,50)) ;
                        InsertCus.Parameters["@Lname"].Value =
txLname.Text ;

                        InsertCus.Parameters.Add(new
SqlParameter("@Address",SqlDbType.VarChar,50)) ;
                        InsertCus.Parameters["@Address"].Value =
txAddress.Text ;

                        InsertCus.Parameters.Add(new
SqlParameter("@Tel",SqlDbType.VarChar,20)) ;
                        InsertCus.Parameters["@Tel"].Value = txTel.Text
;

                        InsertCus.Parameters.Add(new
SqlParameter("@status",SqlDbType.Int)) ;
                        InsertCus.Parameters["@status"].Value = 1 ;

                        InsertCus.Parameters.Add(new
SqlParameter("@Remark",SqlDbType.VarChar,50)) ;
                        InsertCus.Parameters["@Remark"].Value = "" ;

                        myConnection.Open() ;
                        InsertCus.ExecuteNonQuery() ;
                        myConnection.Close() ;

                        MessageBox.Show("เพิ่มเรียบร้อย");
                        this.Owner.Show();
                        this.Hide();
                  }
                  catch(Exception e)
                  {
                        MessageBox.Show(e.Message) ;
                  }
                  }
            private void UpdateMember()
            {
                  try
                  {
                        int         iStatusactive     = 0 ;

                        if(chkActive.Checked != true) //ยังเช่าอยู่
                        {
                              iStatusactive = 1 ;
```

```
                    }

                    string cmdUpdateCus = "update member set
rec_no_room = @rec_no_room,name_member = @namemember"+
                          ",lname_member = @lnamemember,address =
@address,tel = @tel,status = @status,remark = @remark"+
                          " where rec_no_room
="+DataClass.CommonRoom  ;

                    SqlCommand UpdateCus    =     new
SqlCommand(cmdUpdateCus,myConnection) ;

                    UpdateCus.Parameters.Add(new
SqlParameter("@rec_no_room",SqlDbType.VarChar,10)) ;
                    UpdateCus.Parameters["@rec_no_room"].Value =
txRoom.Text ;

                    UpdateCus.Parameters.Add(new
SqlParameter("@namemember",SqlDbType.VarChar,50)) ;
                    UpdateCus.Parameters["@namemember"].Value =
txName.Text ;

                    UpdateCus.Parameters.Add(new
SqlParameter("@lnamemember",SqlDbType.VarChar,50)) ;
                    UpdateCus.Parameters["@lnamemember"].Value =
txLname.Text ;

                    UpdateCus.Parameters.Add(new
SqlParameter("@address",SqlDbType.VarChar,50)) ;
                    UpdateCus.Parameters["@address"].Value =
txAddress.Text ;

                    UpdateCus.Parameters.Add(new
SqlParameter("@tel",SqlDbType.VarChar,20)) ;
                    UpdateCus.Parameters["@tel"].Value = txTel.Text
;

                    UpdateCus.Parameters.Add(new
SqlParameter("@status",SqlDbType.Int)) ;
                    UpdateCus.Parameters["@status"].Value =
iStatusactive ;

                    UpdateCus.Parameters.Add(new
SqlParameter("@remark",SqlDbType.VarChar,50)) ;
                    UpdateCus.Parameters["@remark"].Value = "" ;

                    myConnection.Open() ;
                    UpdateCus.ExecuteNonQuery() ;
                    myConnection.Close() ;

                    MessageBox.Show("แก้ไขรายการเรียบร้อย") ;
                    this.Owner.Show();
                    this.Hide();
               }
               catch(SqlException)
               {
                    MessageBox.Show("Error SQL At UpdateCus") ;
```

```
                    }
                    catch(Exception)
                    {
                          MessageBox.Show("Error Exception At UpdateCus")
;
                    }
               }

               private void AddMember_Activated(object sender, EventArgs
e)
               {
//                  if(DataClass.CommonRoom != "" )
//                  {
//                        chkActive.Visible = true;
//                        this.Text = "แก้ไขข้อมูล";
//
//                        SqlDataAdapter SelectList      =      new
SqlDataAdapter("select name_member,lname_member,address,tel from member
where status =1 and rec_no_room ="+DataClass.CommonRoom,myConnection) ;
//                        ds = new DataSet() ;
//                        SelectList.Fill(ds,"listMember") ;
//                        //Assign Value
//                        txName.Text          =
     ds.Tables["listMember"].Rows[0][0].ToString() ;
//                        txLname.Text         =
     ds.Tables["listMember"].Rows[0][1].ToString() ;
//                        txAddress.Text       =
     ds.Tables["listMember"].Rows[0][2].ToString() ;
//                        txTel.Text           =
     ds.Tables["listMember"].Rows[0][3].ToString() ;
//                  }
               }
          }
}
```

```
############################## Config ###############################
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace WindowsApplication1
{
    public partial class Config : Form
    {
        public Config()
        {
            InitializeComponent();
        }

        private void btnOK_Click(object sender, EventArgs e)
        {
            if (txWaterUnit.Text == "")
            {
                MessageBox.Show("กรุณาระบุราคาค่าน้ำต่อหน่วยด้วยครับ");
                txWaterUnit.Focus();
            }
            else if (txRent.Text == "")
            {
                MessageBox.Show("กรุณาระบุราคาค่าเช่าห้องด้วยครับ");
                txRent.Focus();
            }
            else
            {
                if (Convert.ToDecimal(txWaterUnit.Text) == 0)
                {
                    MessageBox.Show("ค่าน้ำไม่สามารถระบุค่า 0 ได้");
                    txWaterUnit.Text = "";
                    txWaterUnit.Focus();
                }
                else if (Convert.ToDecimal(txRent.Text) == 0)
                {
                    MessageBox.Show("ค่าเช่าห้องไม่สามารถระบุค่า 0 ได้");
                    txRent.Text = "";
                    txRent.Focus();
                }
                else
                    GetMain();
            }

        }

        private void txRent_KeyDown(object sender, KeyEventArgs e)
        {
            if (e.KeyValue == 13)
            {
                txWaterUnit.Focus();
            }else if(e.KeyValue<48 || e.KeyValue > 57)
            {
                MessageBox.Show("กรอกได้เฉพาะตัวเลขเท่านั้น");
                txRent.Clear();
                txRent.Focus();
```

```
            }
        }

        private void txWaterUnit_KeyDown(object sender, KeyEventArgs
e)
        {
            if (e.KeyValue == 13)
            {
                GetMain();
            }
            else if (e.KeyValue < 48 || e.KeyValue > 57)
            {
                MessageBox.Show("กรอกได้เฉพาะตัวเลขเท่านั้น");
                txWaterUnit.Clear();
                txWaterUnit.Focus();
            }
        }
        private void GetMain()
        {
            DataClass.CommonRent = Convert.ToInt32(txRent.Text);
            DataClass.CommonUnit = Convert.ToInt32(txWaterUnit.Text);

            MainMenu main = new MainMenu();
            this.AddOwnedForm(main);
            main.Show();
            this.Hide();
        }

    }
}
```

```
################################ History ##############################
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;
using System.Data.SqlClient;
using System.IO.IsolatedStorage;

namespace WindowsApplication1
{
	/// <summary>
	/// Summary description for History.
	/// </summary>
	public class History : System.Windows.Forms.Form
	{
		private System.Windows.Forms.Button btnOK;
		/// <summary>
		/// Required designer variable.
		/// </summary>
		private System.ComponentModel.Container components = null;

		SqlConnection myConnection ;
		DataSet ds ;
		private DataTable dtCus = new DataTable() ;
		DataRow drCus ;
		private System.Windows.Forms.DataGrid listHistory;

		public struct ColumnMemberInfo
		{
			public string columnname ;
			public HorizontalAlignment          columnaligncol ;
			public object columntype ;
			public int columnwidth ;
			public ColumnMemberInfo(string columnname,object
columntype ,HorizontalAlignment columnaligncol , int columnwidth)
			{
				this.columnname             =     columnname
;
				this.columntype             =     columntype
;
				this.columnaligncol         =
	columnaligncol ;
				this.columnwidth        =     columnwidth ;
			}
		}
		public static ColumnMemberInfo[] AddMemColumn = {

			new ColumnMemberInfo("เลขที่ห้อง",          "int",
	HorizontalAlignment.Center,   80),

			new ColumnMemberInfo("ครั้งที่แล้ว",          "string",
	HorizontalAlignment.Right,    80),
```

```
                new ColumnMemberInfo("ครั้งล่าสุด",                "string",
        HorizontalAlignment.Right,     80),

                new ColumnMemberInfo("ใช้ไป",
        "string",    HorizontalAlignment.Right,     80),

                new ColumnMemberInfo("คิดเป็น",                    "string",
        HorizontalAlignment.Right,     85),


                new ColumnMemberInfo("วันที่",
        "string",    HorizontalAlignment.Left,      100),

            } ;

            public History()
            {
                //
                // Required for Windows Form Designer support
                //
                InitializeComponent();
                CreateTableCus()  ;

                //
                // TODO: Add any constructor code after
    InitializeComponent call
                //
            }

            /// <summary>
            /// Clean up any resources being used.
            /// </summary>
            protected override void Dispose( bool disposing )
            {
                if( disposing )
                {
                    if(components != null)
                    {
                        components.Dispose();
                    }
                }
                base.Dispose( disposing );
            }

            #region Windows Form Designer generated code
            /// <summary>
            /// Required method for Designer support - do not modify
            /// the contents of this method with the code editor.
            /// </summary>
            private void InitializeComponent()
            {
            this.listHistory = new System.Windows.Forms.DataGrid();
            this.btnOK = new System.Windows.Forms.Button();
```

```
((System.ComponentModel.ISupportInitialize)(this.listHistory)).BeginIni
t();
            this.SuspendLayout();
            //
            // listHistory
            //
            this.listHistory.DataMember = "";
            this.listHistory.HeaderForeColor =
System.Drawing.SystemColors.ControlText;
            this.listHistory.Location = new System.Drawing.Point(16,
16);
            this.listHistory.Name = "listHistory";
            this.listHistory.Size = new System.Drawing.Size(544, 240);
            this.listHistory.TabIndex = 0;
            //
            // btnOK
            //
            this.btnOK.Font = new System.Drawing.Font("Microsoft Sans
Serif", 12F, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.btnOK.Location = new System.Drawing.Point(256, 280);
            this.btnOK.Name = "btnOK";
            this.btnOK.Size = new System.Drawing.Size(75, 32);
            this.btnOK.TabIndex = 1;
            this.btnOK.Text = "O.K.";
            this.btnOK.Click += new
System.EventHandler(this.btnOK_Click);
            //
            // History
            //
            this.AutoScaleBaseSize = new System.Drawing.Size(5, 13);
            this.ClientSize = new System.Drawing.Size(568, 350);
            this.Controls.Add(this.btnOK);
            this.Controls.Add(this.listHistory);
            this.Name = "History";
            this.StartPosition =
System.Windows.Forms.FormStartPosition.CenterScreen;
            this.Text = "ประวัติค่าใช้จ่าย";
            this.Closed += new
System.EventHandler(this.History_Closed);
            this.Load += new System.EventHandler(this.History_Load);

((System.ComponentModel.ISupportInitialize)(this.listHistory)).EndInit(
);
            this.ResumeLayout(false);

            }
            #endregion

            private void dataView1_ListChanged(object sender,
System.ComponentModel.ListChangedEventArgs e)
            {

            }

            private void btnOK_Click(object sender, System.EventArgs e)
```

```
		{
			this.Owner.Show();
			this.Hide();
		}

		private void History_Closed(object sender, System.EventArgs
e)
		{
			Application.Exit();
		}

		private void History_Load(object sender, System.EventArgs
e)
		{
			myConnection = new
SqlConnection(DataClass.CommonConnection);
			StatusCusList();
		}
		private void CreateTableCus()
		{
			//ReportList//
			dtCus.TableName = "AddGridrCus" ;
			DataGridTableStyle cusstyle = new
DataGridTableStyle();
			cusstyle.MappingName = "AddGridrCus";

			for (int i =0; i < 6; i ++)
			{
				dtCus.Columns.Add(new
DataColumn(AddMemColumn[i].columnname)) ;

				DataGridColumnStyle MemCol          =      new
DataGridTextBoxColumn();
				MemCol.MappingName                       =
	AddMemColumn[i].columnname;
				MemCol.HeaderText                   =
	AddMemColumn[i].columnname;
				MemCol.Width                             =
	AddMemColumn[i].columnwidth ;
				MemCol.Alignment                    =
	AddMemColumn[i].columnaligncol ;

				cusstyle.GridColumnStyles.Add(MemCol);
				listHistory.TableStyles.Add(cusstyle);

			}
			listHistory.DataSource = dtCus.DefaultView ;

			listHistory.Show() ;
		}
		private void StatusCusList()

		{
			try
			{
				dtCus.Clear() ;
```

```
                        SqlDataAdapter SelectList     =     new
SqlDataAdapter("select
rec_no_room,meter_unit_old,meter_unit_new,water_use,price,rec_date from
journal where rec_no_room = "+DataClass.CommonRoom ,myConnection) ;
                        ds = new DataSet() ;
                        SelectList.Fill(ds,"listHistory") ;
                        int countadd =
ds.Tables["listHistory"].Rows.Count ;
                        for(int i = 0 ;i < countadd ; i++ )
                        {
                              drCus = dtCus.NewRow() ;
                              drCus[0]    =
ds.Tables["listHistory"].Rows[i][0].ToString() ;
                              drCus[1]    =
ds.Tables["listHistory"].Rows[i][1].ToString() ;
                              drCus[2]    =
ds.Tables["listHistory"].Rows[i][2].ToString() ;
                              drCus[3]    =
ds.Tables["listHistory"].Rows[i][3].ToString() ;
                              drCus[4]    =
ds.Tables["listHistory"].Rows[i][4].ToString() ;
                              drCus[5]    =
Convert.ToDateTime(ds.Tables["listHistory"].Rows[i][5]).ToLongDateStrin
g() ;

                              dtCus.Rows.Add(drCus) ;
                        }
                        ds.Tables.Remove("listHistory") ;

                        listHistory.DataSource = dtCus.DefaultView ;
                        listHistory.Show() ;
                  }
                  catch(SqlException)
                  {
                        MessageBox.Show("ไม่สามารถทำการ Connect Server ได้ กรุณา
ตรวจสอบการเชื่อมต่อกับระบบ") ;
                  }
            }

      }
}
```

```
################################ Main Manu ##########################
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;
using System.Data.SqlClient;
using System.IO.Ports;

namespace WindowsApplication1
{
      /// <summary>
      /// Summary description for Form1.
      /// </summary>
      public class MainMenu : System.Windows.Forms.Form
    {
            private System.Windows.Forms.Button button4;
            private System.Windows.Forms.Button button5;
            private System.Windows.Forms.Button button6;
            private System.Windows.Forms.Button button3;
            private System.Windows.Forms.Button btnRoom;
        private System.Windows.Forms.Button btnExit;
        private IContainer components;

            SqlConnection myConnection ;
            DataSet ds ;
            private DataTable dtCus = new DataTable() ;
            DataRow drCus ;
            private System.Windows.Forms.DataGrid listMember;
        private SerialPort serialPort1;

        SerialPort sp = new SerialPort();

            public struct ColumnMemberInfo
            {
                  public string columnname ;
                  public HorizontalAlignment          columnaligncol ;
                  public object columntype ;
                  public int columnwidth ;
                  public ColumnMemberInfo(string columnname,object
columntype ,HorizontalAlignment columnaligncol , int columnwidth)
                  {
                        this.columnname                =     columnname
;
                        this.columntype                =     columntype
;
                        this.columnaligncol            =
      columnaligncol ;
                        this.columnwidth         =     columnwidth ;
                  }
            }
            public static ColumnMemberInfo[] AddMemColumn = {

             new ColumnMemberInfo("เลขที่ห้อง",                    "int",
      HorizontalAlignment.Left,      50),
```

```
                new ColumnMemberInfo("ชื่อเจ้าของห้อง",                    "string",
        HorizontalAlignment.Left,       100),

                new ColumnMemberInfo("นามสกุล เจ้าของห้อง",          "string",
        HorizontalAlignment.Left,       140),

                new ColumnMemberInfo("ที่อยู่",                        "string",
        HorizontalAlignment.Left,       200),

                new ColumnMemberInfo("เบอร์โทรศัพท์",                  "string",
        HorizontalAlignment.Left,       95),

                } ;
                public MainMenu()
                {
                        //
                        // Required for Windows Form Designer support
                        //
                        InitializeComponent();
                        CreateTableCus()    ;

                        //
                        // TODO: Add any constructor code after
InitializeComponent call
                        //
                }

                /// <summary>
                /// Clean up any resources being used.
                /// </summary>
                protected override void Dispose( bool disposing )
                {
                        if( disposing )
                        {
                                if (components != null)
                                {
                                        components.Dispose();
                                }
                        }
                        base.Dispose( disposing );
                }

                #region Windows Form Designer generated code
                /// <summary>
                /// Required method for Designer support - do not modify
                /// the contents of this method with the code editor.
                /// </summary>
                private void InitializeComponent()
                {
                this.components = new System.ComponentModel.Container();
                this.btnRoom = new System.Windows.Forms.Button();
                this.button4 = new System.Windows.Forms.Button();
                this.button5 = new System.Windows.Forms.Button();
                this.button6 = new System.Windows.Forms.Button();
                this.button3 = new System.Windows.Forms.Button();
```

```
            this.listMember = new System.Windows.Forms.DataGrid();
            this.btnExit = new System.Windows.Forms.Button();
            this.serialPort1 = new
System.IO.Ports.SerialPort(this.components);

((System.ComponentModel.ISupportInitialize)(this.listMember)).BeginInit
();
            this.SuspendLayout();
            //
            // btnRoom
            //
            this.btnRoom.Font = new System.Drawing.Font("Microsoft Sans
Serif", 11.25F, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.btnRoom.Location = new System.Drawing.Point(8, 8);
            this.btnRoom.Name = "btnRoom";
            this.btnRoom.Size = new System.Drawing.Size(113, 56);
            this.btnRoom.TabIndex = 0;
            this.btnRoom.Text = "เลือกห้อง";
            this.btnRoom.Click += new
System.EventHandler(this.button1_Click);
            //
            // button4
            //
            this.button4.Font = new System.Drawing.Font("Microsoft Sans
Serif", 11.25F, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.button4.Location = new System.Drawing.Point(246, 8);
            this.button4.Name = "button4";
            this.button4.Size = new System.Drawing.Size(117, 56);
            this.button4.TabIndex = 3;
            this.button4.Text = "เพิ่มผู้ใช้";
            this.button4.Click += new
System.EventHandler(this.button4_Click);
            //
            // button5
            //
            this.button5.Font = new System.Drawing.Font("Microsoft Sans
Serif", 11.25F, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.button5.Location = new System.Drawing.Point(369, 8);
            this.button5.Name = "button5";
            this.button5.Size = new System.Drawing.Size(123, 56);
            this.button5.TabIndex = 4;
            this.button5.Text = "แสดง/แก้ไข ผู้ใช้";
            this.button5.Click += new
System.EventHandler(this.button5_Click);
            //
            // button6
            //
            this.button6.Font = new System.Drawing.Font("Microsoft Sans
Serif", 11.25F, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.button6.Location = new System.Drawing.Point(498, 8);
            this.button6.Name = "button6";
            this.button6.Size = new System.Drawing.Size(126, 56);
```

```
            this.button6.TabIndex = 5;
            this.button6.Text = "ประวัติค่าใช้จ่าย";
            this.button6.Click += new
System.EventHandler(this.button6_Click);
            //
            // button3
            //
            this.button3.Font = new System.Drawing.Font("Microsoft Sans
Serif", 11.25F, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.button3.Location = new System.Drawing.Point(127, 8);
            this.button3.Name = "button3";
            this.button3.Size = new System.Drawing.Size(113, 56);
            this.button3.TabIndex = 1;
            this.button3.Text = "คำนวณค่าน้ำ";
            this.button3.Click += new
System.EventHandler(this.button3_Click);
            //
            // listMember
            //
            this.listMember.DataMember = "";
            this.listMember.HeaderForeColor =
System.Drawing.SystemColors.ControlText;
            this.listMember.Location = new System.Drawing.Point(8, 72);
            this.listMember.Name = "listMember";
            this.listMember.Size = new System.Drawing.Size(624, 280);
            this.listMember.TabIndex = 7;
            //
            // btnExit
            //
            this.btnExit.Font = new System.Drawing.Font("Microsoft Sans
Serif", 11.25F, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.btnExit.Location = new System.Drawing.Point(264, 368);
            this.btnExit.Name = "btnExit";
            this.btnExit.Size = new System.Drawing.Size(104, 56);
            this.btnExit.TabIndex = 8;
            this.btnExit.Text = "ออกจากโปรแกรม";
            this.btnExit.Click += new
System.EventHandler(this.btnExit_Click);
            //
            // serialPort1
            //
            this.serialPort1.DataReceived += new
System.IO.Ports.SerialDataReceivedEventHandler(this.serialPort1_DataRec
eived);
            //
            // MainMenu
            //
            this.AutoScaleBaseSize = new System.Drawing.Size(5, 13);
            this.ClientSize = new System.Drawing.Size(636, 446);
            this.Controls.Add(this.btnExit);
            this.Controls.Add(this.listMember);
            this.Controls.Add(this.button3);
            this.Controls.Add(this.button6);
            this.Controls.Add(this.button5);
```

```
            this.Controls.Add(this.button4);
            this.Controls.Add(this.btnRoom);
            this.Name = "MainMenu";
            this.StartPosition =
System.Windows.Forms.FormStartPosition.CenterScreen;
            this.Text = "โปรแกรมคำนวณค่าน้ำ";
            this.Closed += new
System.EventHandler(this.MainMenu_Closed);
            this.Activated += new
System.EventHandler(this.MainMenu_Activated);
            this.KeyDown += new
System.Windows.Forms.KeyEventHandler(this.MainMenu_KeyDown);
            this.Load += new System.EventHandler(this.MainMenu_Load);

((System.ComponentModel.ISupportInitialize)(this.listMember)).EndInit()
;
            this.ResumeLayout(false);

            }
            #endregion

            /// <summary>
            /// The main entry point for the application.
            /// </summary>
            [STAThread]
            static void Main()
            {
                  Application.EnableVisualStyles();
                  Application.Run(new Config());
            }

            private void MainMenu_Load(object sender, System.EventArgs
e)
            {
                  myConnection = new
SqlConnection(DataClass.CommonConnection);
            }

            //Room
            private void button1_Click(object sender, System.EventArgs
e)
            {
                  Room room = new Room();
                  this.AddOwnedForm(room);
                  room.Show();
                  this.Hide();
            }
            //Add Mem
            private void button4_Click(object sender, System.EventArgs
e)
            {
                  DataClass.CommonRoom = "";
                  AddMember addmem = new AddMember();
                  this.AddOwnedForm(addmem);
                  addmem.Show();
                  this.Hide();
            }
```

```
		//View/Edit
		private void button5_Click(object sender, System.EventArgs
e)
		{
			if(DataClass.CommonRoom != "")
			{
				AddMember addmem = new AddMember();
				this.AddOwnedForm(addmem);
				addmem.Show();
				addmem.txRoom.Enabled = false ;
				addmem.txRoom.Text = DataClass.CommonRoom;
				this.Hide();
			}
			else
			{
				MessageBox.Show("กรุณาทำการเลือกห้องก่อนทำรายการแก้ไขนะครับ");

			}
		}

		//History
		private void button6_Click(object sender, System.EventArgs
e)
		{
		if (DataClass.CommonRoom == "")
		{
		    MessageBox.Show("กรุณาเลือกห้องที่ต้องการดูประวัติค่าใช้จ่ายด้วยครับ");
		}
		else
		{
		    SqlDataAdapter SelectList =      new
SqlDataAdapter("select * from journal where rec_no_room =
'"+DataClass.CommonRoom+"'",myConnection) ;
				ds = new DataSet() ;
				SelectList.Fill(ds,"listRoom") ;

		    if (ds.Tables["listRoom"].Rows.Count != 0)
		    {
			History history = new History();
			this.AddOwnedForm(history);
			history.Show();
			this.Hide();
		    }
		    else
		    {
			MessageBox.Show("ยังไม่มีรายการค่าใช้จ่ายของห้องนี้");
		    }
		}
		}

		//Calculate
		private void button3_Click(object sender, System.EventArgs
e)
		{
		//send DataClass.CommonRoom
```

```
            //sp.WriteLine(DataClass.CommonRoom);            //ส่งเลขห้องไปที่บอร์ด

            //receive unit;
            //DataClass.CommonMeterUnit = sp.ReadLine();     //รับค่ามิเตอร์จากบอร์ด


            DataClass.CommonMeterUnit = 25;//ค่าน้ำล่าสุด กำหนดเอง เมื่อเสร็จแล้วจะดึงค่าจาก
COM1

            Print printing = new Print();
            this.AddOwnedForm(printing);
            printing.Show();
            printing.btnPrint.Enabled = true;
            this.Hide();
            }

            private void btnExit_Click(object sender, System.EventArgs
e)
            {
                  Application.Exit() ;
            }

            private void MainMenu_Closed(object sender,
System.EventArgs e)
            {
                        Application.Exit();
            }

            private void MainMenu_Activated(object sender, EventArgs e)
            {
                  StatusCusList();
            }

            private void CreateTableCus()
            {
                  //ReportList//
                  dtCus.TableName = "AddGridrCus" ;
                  DataGridTableStyle cusstyle = new
DataGridTableStyle();
                  cusstyle.MappingName = "AddGridrCus";

                  for (int i =0; i < 5; i ++)
                  {
                        dtCus.Columns.Add(new
DataColumn(AddMemColumn[i].columnname)) ;

                        DataGridColumnStyle MemCol          =      new
DataGridTextBoxColumn();
                        MemCol.MappingName                         =
      AddMemColumn[i].columnname;
                        MemCol.HeaderText                   =
      AddMemColumn[i].columnname;
                        MemCol.Width                               =
      AddMemColumn[i].columnwidth ;
```

```
                    MemCol.Alignment                    =
        AddMemColumn[i].columnaligncol ;

                    cusstyle.GridColumnStyles.Add(MemCol);
                    listMember.TableStyles.Add(cusstyle);

                }
                listMember.DataSource = dtCus.DefaultView ;

                listMember.Show() ;
            }
            private void StatusCusList()
            {
                try
                {
                    dtCus.Clear() ;
                    SqlDataAdapter SelectList     =     new
SqlDataAdapter("select rec_no_room,name_member,lname_member,address,tel
from member where status =1",myConnection) ;
                    ds = new DataSet() ;
                    SelectList.Fill(ds,"listMember") ;
                    int countadd =
ds.Tables["listMember"].Rows.Count ;
                    for(int i = 0 ;i < countadd ; i++ )
                    {
                        drCus = dtCus.NewRow() ;
                        drCus[0]     =
ds.Tables["listMember"].Rows[i][0].ToString() ;
                        drCus[1]     =
ds.Tables["listMember"].Rows[i][1].ToString() ;
                        drCus[2]     =
ds.Tables["listMember"].Rows[i][2].ToString() ;
                        drCus[3]     =
ds.Tables["listMember"].Rows[i][3].ToString() ;
                        drCus[4]     =
ds.Tables["listMember"].Rows[i][4].ToString() ;

                        dtCus.Rows.Add(drCus) ;
                    }
                    ds.Tables.Remove("listMember") ;

                    listMember.DataSource = dtCus.DefaultView ;
                    listMember.Show() ;
                }
                catch(SqlException)
                {
                    MessageBox.Show("ไม่สามารถทำการ Connect Server ได้ กรุณา
ตรวจสอบการเชื่อมต่อกับระบบ") ;
                }
            }
        private void GetDataWithSE()
        {
            serialPort1.Open();
        }
```

```
		private void MainMenu_KeyDown(object sender,
System.Windows.Forms.KeyEventArgs e)
		{
			if(e.KeyValue == 41 ||e.KeyValue == 61)
			{
				DataClass.CommonRoom = "";
				AddMember addmem = new AddMember();
				this.AddOwnedForm(addmem);
				addmem.Show();
				this.Hide();
			}
			else if(e.KeyValue == 43 ||e.KeyValue ==
63)//Calculate
			{
				if(DataClass.CommonRoom != "")
				{
					//send room number
					//send DataClass.CommonRoom

					//receive meter unit
					//sample meter unit = 12345

					//Calulate
				}
				else
				{
					MessageBox.Show("กรุณาเลือกห้องที่ต้องการคำนวณก่อนครับ");
				}
			}
			else if(e.KeyValue == 45 ||e.KeyValue == 65)//Exit
			{
				Application.Exit();
			}
			else if(e.KeyValue == 48 ||e.KeyValue == 68)//History
			{
				History history = new History();
				this.AddOwnedForm(history);
				history.Show();
				this.Hide();
			}
			else if(e.KeyValue == 50 ||e.KeyValue == 70)//Print
			{
			}
			else if(e.KeyValue == 52 ||e.KeyValue == 72)//Room
			{
				Room room = new Room();
				this.AddOwnedForm(room);
				room.Show();
				this.Hide();
			}
			else if(e.KeyValue == 56 ||e.KeyValue == 76)//View
			{

				if(DataClass.CommonRoom != "")
				{
					AddMember addmem = new AddMember();
					this.AddOwnedForm(addmem);
```

```
                            addmem.Show();
                            addmem.txRoom.Enabled = false ;
                            addmem.txRoom.Text =
DataClass.CommonRoom;
                            this.Hide();
                        }
                        else
                        {
                            MessageBox.Show("กรุณาทำการเลือกห้องก่อนทำรายการแก้ไขนะ
ครับ");
                        }
                    }
                }

        private void serialPort1_DataReceived(object sender,
SerialDataReceivedEventArgs e)
        {

        }

    }
}
```

```
################################# Print #################################
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;
using System.Data.SqlClient;

namespace WindowsApplication1
{
      /// <summary>
      /// Summary description for Print.
      /// </summary>
      public class Print : System.Windows.Forms.Form
      {
        private TableLayoutPanel tableLayoutPanel1;
        private TableLayoutPanel tableLayoutPanel2;
        private TableLayoutPanel tableLayoutPanel3;
        private TableLayoutPanel tableLayoutPanel4;
        private Label lbName;
        private Label lbRoom;
        private Label label9;
        private Label label8;
        private Label label2;
        private Label lbUse;
        private Label lbPrice;
        private Label lbOldUnit;
        private Label lbNewUnit;
        private Label label3;
        private Label label7;
        private Label label6;
        private Label label5;
        private Label label4;
        private Label label1;
        private Label label16;
        private TableLayoutPanel tableLayoutPanel5;
        private Label label18;
        private Label label17;
        public Button btnPrint;
        private Button btnBack;
            /// <summary>
            /// Required designer variable.
            /// </summary>
            private System.ComponentModel.Container components = null;
        SqlConnection myConnection;
        DataSet dsUser;
        DataSet dsMax;
        DataSet ds;
        private Label lbDate;
        private Label lbRent;
        private Label lbRentWater;
        private TableLayoutPanel tableLayoutPanel6;
        private Label lbUnit;
        DataSet dsJrnl;

        //Declare Variable For Use
```

```
        string strRoom = DataClass.CommonRoom;
        decimal d_NewMeter = DataClass.CommonMeterUnit;
        decimal d_UnitPrice = DataClass.CommonUnit;
        string strUser;
        decimal d_OldMeter;
        decimal d_User;
        decimal d_Price;
        private System.Drawing.Printing.PrintDocument print1;
        private Label label10;
        decimal d_RentWater;

            public Print()
            {
                  //
                  // Required for Windows Form Designer support
                  //
                  InitializeComponent();

                  //
                  // TODO: Add any constructor code after
InitializeComponent call
                  //
            }

            /// <summary>
            /// Clean up any resources being used.
            /// </summary>
            protected override void Dispose( bool disposing )
            {
                  if( disposing )
                  {
                        if(components != null)
                        {
                              components.Dispose();
                        }
                  }
                  base.Dispose( disposing );
            }

            #region Windows Form Designer generated code
            /// <summary>
            /// Required method for Designer support - do not modify
            /// the contents of this method with the code editor.
            /// </summary>
            private void InitializeComponent()
            {
            this.tableLayoutPanel1 = new
System.Windows.Forms.TableLayoutPanel();
            this.label10 = new System.Windows.Forms.Label();
            this.lbName = new System.Windows.Forms.Label();
            this.label8 = new System.Windows.Forms.Label();
            this.label2 = new System.Windows.Forms.Label();
            this.tableLayoutPanel2 = new
System.Windows.Forms.TableLayoutPanel();
            this.tableLayoutPanel3 = new
System.Windows.Forms.TableLayoutPanel();
```

```
            this.label7 = new System.Windows.Forms.Label();
            this.label6 = new System.Windows.Forms.Label();
            this.label5 = new System.Windows.Forms.Label();
            this.label4 = new System.Windows.Forms.Label();
            this.tableLayoutPanel6 = new
System.Windows.Forms.TableLayoutPanel();
            this.label3 = new System.Windows.Forms.Label();
            this.lbUnit = new System.Windows.Forms.Label();
            this.label1 = new System.Windows.Forms.Label();
            this.label16 = new System.Windows.Forms.Label();
            this.label9 = new System.Windows.Forms.Label();
            this.lbRoom = new System.Windows.Forms.Label();
            this.tableLayoutPanel4 = new
System.Windows.Forms.TableLayoutPanel();
            this.lbUse = new System.Windows.Forms.Label();
            this.lbPrice = new System.Windows.Forms.Label();
            this.lbOldUnit = new System.Windows.Forms.Label();
            this.lbNewUnit = new System.Windows.Forms.Label();
            this.lbRent = new System.Windows.Forms.Label();
            this.lbRentWater = new System.Windows.Forms.Label();
            this.tableLayoutPanel5 = new
System.Windows.Forms.TableLayoutPanel();
            this.lbDate = new System.Windows.Forms.Label();
            this.label18 = new System.Windows.Forms.Label();
            this.label17 = new System.Windows.Forms.Label();
            this.btnPrint = new System.Windows.Forms.Button();
            this.btnBack = new System.Windows.Forms.Button();
            this.print1 = new System.Drawing.Printing.PrintDocument();
            this.tableLayoutPanel1.SuspendLayout();
            this.tableLayoutPanel2.SuspendLayout();
            this.tableLayoutPanel3.SuspendLayout();
            this.tableLayoutPanel6.SuspendLayout();
            this.tableLayoutPanel4.SuspendLayout();
            this.tableLayoutPanel5.SuspendLayout();
            this.SuspendLayout();
            //
            // tableLayoutPanel1
            //
            this.tableLayoutPanel1.CellBorderStyle =
System.Windows.Forms.TableLayoutPanelCellBorderStyle.OutsetDouble;
            this.tableLayoutPanel1.ColumnCount = 5;
            this.tableLayoutPanel1.ColumnStyles.Add(new
System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Percent,
24.57338F));
            this.tableLayoutPanel1.ColumnStyles.Add(new
System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Percent,
75.42662F));
            this.tableLayoutPanel1.ColumnStyles.Add(new
System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Absolute
, 279F));
            this.tableLayoutPanel1.ColumnStyles.Add(new
System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Absolute
, 56F));
            this.tableLayoutPanel1.ColumnStyles.Add(new
System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Absolute
, 86F));
```

```
            this.tableLayoutPanel1.ColumnStyles.Add(new
System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Absolute
, 61F));
            this.tableLayoutPanel1.ColumnStyles.Add(new
System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Absolute
, 66F));
            this.tableLayoutPanel1.ColumnStyles.Add(new
System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Absolute
, 108F));
            this.tableLayoutPanel1.Controls.Add(this.label10, 0, 3);
            this.tableLayoutPanel1.Controls.Add(this.lbName, 1, 1);
            this.tableLayoutPanel1.Controls.Add(this.label8, 3, 0);
            this.tableLayoutPanel1.Controls.Add(this.label2, 1, 0);
            this.tableLayoutPanel1.Controls.Add(this.tableLayoutPanel2,
2, 0);
            this.tableLayoutPanel1.Controls.Add(this.label1, 0, 0);
            this.tableLayoutPanel1.Controls.Add(this.label16, 0, 2);
            this.tableLayoutPanel1.Controls.Add(this.label9, 4, 0);
            this.tableLayoutPanel1.Controls.Add(this.lbRoom, 0, 1);
            this.tableLayoutPanel1.Controls.Add(this.tableLayoutPanel4,
2, 1);
            this.tableLayoutPanel1.Controls.Add(this.lbRent, 1, 2);
            this.tableLayoutPanel1.Controls.Add(this.lbRentWater, 2,
2);
            this.tableLayoutPanel1.Location = new
System.Drawing.Point(12, 51);
            this.tableLayoutPanel1.Name = "tableLayoutPanel1";
            this.tableLayoutPanel1.RowCount = 4;
            this.tableLayoutPanel1.RowStyles.Add(new
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Percent,
36.99634F));
            this.tableLayoutPanel1.RowStyles.Add(new
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Percent,
63.00366F));
            this.tableLayoutPanel1.RowStyles.Add(new
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Absolute,
20F));
            this.tableLayoutPanel1.RowStyles.Add(new
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Absolute,
20F));
            this.tableLayoutPanel1.RowStyles.Add(new
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Absolute,
20F));
            this.tableLayoutPanel1.RowStyles.Add(new
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Absolute,
20F));
            this.tableLayoutPanel1.Size = new System.Drawing.Size(690,
260);
            this.tableLayoutPanel1.TabIndex = 0;
            //
            // label10
            //
            this.label10.Location = new System.Drawing.Point(6, 236);
            this.label10.Name = "label10";
            this.label10.Size = new System.Drawing.Size(55, 21);
            this.label10.TabIndex = 12;
            this.label10.Text = "รวมทั้งสิ้น";
```

```
            //
            // lbName
            //
            this.lbName.Location = new System.Drawing.Point(70, 81);
            this.lbName.Name = "lbName";
            this.lbName.Size = new System.Drawing.Size(183, 39);
            this.lbName.TabIndex = 8;
            this.lbName.TextAlign =
System.Drawing.ContentAlignment.MiddleLeft;
            //
            // label8
            //
            this.label8.Location = new System.Drawing.Point(544, 3);
            this.label8.Name = "label8";
            this.label8.Size = new System.Drawing.Size(50, 39);
            this.label8.TabIndex = 5;
            this.label8.Text = "ค่าไฟ";
            this.label8.TextAlign =
System.Drawing.ContentAlignment.MiddleCenter;
            //
            // label2
            //
            this.label2.Location = new System.Drawing.Point(70, 3);
            this.label2.Name = "label2";
            this.label2.Size = new System.Drawing.Size(183, 39);
            this.label2.TabIndex = 4;
            this.label2.Text = "ชื่อผู้เช่า";
            this.label2.TextAlign =
System.Drawing.ContentAlignment.MiddleCenter;
            //
            // tableLayoutPanel2
            //
            this.tableLayoutPanel2.CellBorderStyle =
System.Windows.Forms.TableLayoutPanelCellBorderStyle.OutsetDouble;
            this.tableLayoutPanel2.ColumnCount = 1;
            this.tableLayoutPanel2.ColumnStyles.Add(new
System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Percent,
50F));
            this.tableLayoutPanel2.ColumnStyles.Add(new
System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Percent,
50F));
            this.tableLayoutPanel2.Controls.Add(this.tableLayoutPanel3,
0, 1);
            this.tableLayoutPanel2.Controls.Add(this.tableLayoutPanel6,
0, 0);
            this.tableLayoutPanel2.Location = new
System.Drawing.Point(262, 6);
            this.tableLayoutPanel2.Name = "tableLayoutPanel2";
            this.tableLayoutPanel2.RowCount = 2;
            this.tableLayoutPanel2.RowStyles.Add(new
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Percent,
50F));
            this.tableLayoutPanel2.RowStyles.Add(new
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Absolute,
43F));
```

```
            this.tableLayoutPanel2.Size = new System.Drawing.Size(273,
67);
            this.tableLayoutPanel2.TabIndex = 1;
            //
            // tableLayoutPanel3
            //
            this.tableLayoutPanel3.CellBorderStyle =
System.Windows.Forms.TableLayoutPanelCellBorderStyle.OutsetDouble;
            this.tableLayoutPanel3.ColumnCount = 4;
            this.tableLayoutPanel3.ColumnStyles.Add(new
System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Absolute
, 59F));
            this.tableLayoutPanel3.ColumnStyles.Add(new
System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Absolute
, 57F));
            this.tableLayoutPanel3.ColumnStyles.Add(new
System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Absolute
, 63F));
            this.tableLayoutPanel3.ColumnStyles.Add(new
System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Absolute
, 96F));
            this.tableLayoutPanel3.Controls.Add(this.label7, 0, 0);
            this.tableLayoutPanel3.Controls.Add(this.label6, 0, 0);
            this.tableLayoutPanel3.Controls.Add(this.label5, 0, 0);
            this.tableLayoutPanel3.Controls.Add(this.label4, 0, 0);
            this.tableLayoutPanel3.Location = new
System.Drawing.Point(6, 24);
            this.tableLayoutPanel3.Name = "tableLayoutPanel3";
            this.tableLayoutPanel3.RowCount = 1;
            this.tableLayoutPanel3.RowStyles.Add(new
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Percent,
100F));
            this.tableLayoutPanel3.Size = new System.Drawing.Size(261,
37);
            this.tableLayoutPanel3.TabIndex = 1;
            //
            // label7
            //
            this.label7.Location = new System.Drawing.Point(128, 3);
            this.label7.Name = "label7";
            this.label7.Size = new System.Drawing.Size(56, 31);
            this.label7.TabIndex = 7;
            this.label7.Text = "ใช้ไป (หน่วย)";
            this.label7.TextAlign =
System.Drawing.ContentAlignment.MiddleCenter;
            //
            // label6
            //
            this.label6.Location = new System.Drawing.Point(194, 3);
            this.label6.Name = "label6";
            this.label6.Size = new System.Drawing.Size(56, 31);
            this.label6.TabIndex = 6;
            this.label6.Text = "คิดเป็น (บาท)";
            this.label6.TextAlign =
System.Drawing.ContentAlignment.MiddleCenter;
            //
```

```
            // label5
            //
            this.label5.Location = new System.Drawing.Point(6, 3);
            this.label5.Name = "label5";
            this.label5.Size = new System.Drawing.Size(53, 31);
            this.label5.TabIndex = 5;
            this.label5.Text = "ครั้งที่แล้ว";
            this.label5.TextAlign =
System.Drawing.ContentAlignment.MiddleCenter;
            //
            // label4
            //
            this.label4.Location = new System.Drawing.Point(68, 3);
            this.label4.Name = "label4";
            this.label4.Size = new System.Drawing.Size(51, 31);
            this.label4.TabIndex = 4;
            this.label4.Text = "ครั้งล่าสุด";
            this.label4.TextAlign =
System.Drawing.ContentAlignment.MiddleCenter;
            //
            // tableLayoutPanel6
            //
            this.tableLayoutPanel6.ColumnCount = 2;
            this.tableLayoutPanel6.ColumnStyles.Add(new
System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Percent,
56.2753F));
            this.tableLayoutPanel6.ColumnStyles.Add(new
System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Percent,
43.7247F));
            this.tableLayoutPanel6.Controls.Add(this.label3, 0, 0);
            this.tableLayoutPanel6.Controls.Add(this.lbUnit, 1, 0);
            this.tableLayoutPanel6.Location = new
System.Drawing.Point(6, 6);
            this.tableLayoutPanel6.Name = "tableLayoutPanel6";
            this.tableLayoutPanel6.RowCount = 1;
            this.tableLayoutPanel6.RowStyles.Add(new
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Percent,
50F));
            this.tableLayoutPanel6.RowStyles.Add(new
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Percent,
50F));
            this.tableLayoutPanel6.Size = new System.Drawing.Size(247,
9);
            this.tableLayoutPanel6.TabIndex = 4;
            //
            // label3
            //
            this.label3.Location = new System.Drawing.Point(3, 0);
            this.label3.Name = "label3";
            this.label3.Size = new System.Drawing.Size(124, 9);
            this.label3.TabIndex = 4;
            this.label3.Text = "ค่าน้ำหน่วยละ";
            this.label3.TextAlign =
System.Drawing.ContentAlignment.MiddleRight;
            //
            // lbUnit
```

```
            //
            this.lbUnit.Location = new System.Drawing.Point(141, 0);
            this.lbUnit.Name = "lbUnit";
            this.lbUnit.Size = new System.Drawing.Size(83, 9);
            this.lbUnit.TabIndex = 5;
            this.lbUnit.TextAlign =
System.Drawing.ContentAlignment.MiddleLeft;
            //
            // label1
            //
            this.label1.Location = new System.Drawing.Point(6, 3);
            this.label1.Name = "label1";
            this.label1.Size = new System.Drawing.Size(49, 73);
            this.label1.TabIndex = 3;
            this.label1.Text = "ห้อง";
            this.label1.TextAlign =
System.Drawing.ContentAlignment.MiddleCenter;
            //
            // label16
            //
            this.label16.Location = new System.Drawing.Point(6, 213);
            this.label16.Name = "label16";
            this.label16.Size = new System.Drawing.Size(55, 20);
            this.label16.TabIndex = 9;
            this.label16.Text = "ค่าห้อง";
            //
            // label9
            //
            this.label9.Location = new System.Drawing.Point(603, 3);
            this.label9.Name = "label9";
            this.label9.Size = new System.Drawing.Size(34, 39);
            this.label9.TabIndex = 6;
            this.label9.Text = "รวม";
            this.label9.TextAlign =
System.Drawing.ContentAlignment.MiddleCenter;
            //
            // lbRoom
            //
            this.lbRoom.Location = new System.Drawing.Point(6, 81);
            this.lbRoom.Name = "lbRoom";
            this.lbRoom.Size = new System.Drawing.Size(55, 39);
            this.lbRoom.TabIndex = 7;
            this.lbRoom.TextAlign =
System.Drawing.ContentAlignment.MiddleCenter;
            //
            // tableLayoutPanel4
            //
            this.tableLayoutPanel4.CellBorderStyle =
System.Windows.Forms.TableLayoutPanelCellBorderStyle.OutsetDouble;
            this.tableLayoutPanel4.ColumnCount = 4;
            this.tableLayoutPanel4.ColumnStyles.Add(new
System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Percent,
52.71318F));
            this.tableLayoutPanel4.ColumnStyles.Add(new
System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Percent,
47.28682F));
```

```
            this.tableLayoutPanel4.ColumnStyles.Add(new
System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Absolute
, 62F));
            this.tableLayoutPanel4.ColumnStyles.Add(new
System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Absolute
, 70F));
            this.tableLayoutPanel4.Controls.Add(this.lbUse, 0, 0);
            this.tableLayoutPanel4.Controls.Add(this.lbPrice, 0, 0);
            this.tableLayoutPanel4.Controls.Add(this.lbOldUnit, 0, 0);
            this.tableLayoutPanel4.Controls.Add(this.lbNewUnit, 0, 0);
            this.tableLayoutPanel4.Location = new
System.Drawing.Point(262, 84);
            this.tableLayoutPanel4.Name = "tableLayoutPanel4";
            this.tableLayoutPanel4.RowCount = 1;
            this.tableLayoutPanel4.RowStyles.Add(new
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Percent,
50F));
            this.tableLayoutPanel4.RowStyles.Add(new
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Percent,
50F));
            this.tableLayoutPanel4.Size = new System.Drawing.Size(267,
118);
            this.tableLayoutPanel4.TabIndex = 2;
            //
            // lbUse
            //
            this.lbUse.Location = new System.Drawing.Point(131, 3);
            this.lbUse.Name = "lbUse";
            this.lbUse.Size = new System.Drawing.Size(56, 39);
            this.lbUse.TabIndex = 7;
            this.lbUse.TextAlign =
System.Drawing.ContentAlignment.MiddleRight;
            //
            // lbPrice
            //
            this.lbPrice.Location = new System.Drawing.Point(196, 3);
            this.lbPrice.Name = "lbPrice";
            this.lbPrice.Size = new System.Drawing.Size(54, 39);
            this.lbPrice.TabIndex = 6;
            this.lbPrice.TextAlign =
System.Drawing.ContentAlignment.MiddleRight;
            //
            // lbOldUnit
            //
            this.lbOldUnit.Location = new System.Drawing.Point(6, 3);
            this.lbOldUnit.Name = "lbOldUnit";
            this.lbOldUnit.Size = new System.Drawing.Size(45, 39);
            this.lbOldUnit.TabIndex = 5;
            this.lbOldUnit.TextAlign =
System.Drawing.ContentAlignment.MiddleRight;
            //
            // lbNewUnit
            //
            this.lbNewUnit.Location = new System.Drawing.Point(72, 3);
            this.lbNewUnit.Name = "lbNewUnit";
            this.lbNewUnit.Size = new System.Drawing.Size(40, 39);
            this.lbNewUnit.TabIndex = 4;
```

```
            this.lbNewUnit.TextAlign =
System.Drawing.ContentAlignment.MiddleRight;
            //
            // lbRent
            //
            this.lbRent.Location = new System.Drawing.Point(70, 213);
            this.lbRent.Name = "lbRent";
            this.lbRent.Size = new System.Drawing.Size(100, 20);
            this.lbRent.TabIndex = 10;
            this.lbRent.TextAlign =
System.Drawing.ContentAlignment.MiddleLeft;
            //
            // lbRentWater
            //
            this.lbRentWater.Location = new System.Drawing.Point(262,
213);
            this.lbRentWater.Name = "lbRentWater";
            this.lbRentWater.Size = new System.Drawing.Size(267, 20);
            this.lbRentWater.TabIndex = 11;
            this.lbRentWater.TextAlign =
System.Drawing.ContentAlignment.MiddleRight;
            //
            // tableLayoutPanel5
            //
            this.tableLayoutPanel5.ColumnCount = 3;
            this.tableLayoutPanel5.ColumnStyles.Add(new
System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Percent,
86.94639F));
            this.tableLayoutPanel5.ColumnStyles.Add(new
System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Percent,
13.05361F));
            this.tableLayoutPanel5.ColumnStyles.Add(new
System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Absolute
, 275F));
            this.tableLayoutPanel5.Controls.Add(this.lbDate, 2, 0);
            this.tableLayoutPanel5.Controls.Add(this.label18, 1, 0);
            this.tableLayoutPanel5.Controls.Add(this.label17, 0, 0);
            this.tableLayoutPanel5.Location = new
System.Drawing.Point(12, 358);
            this.tableLayoutPanel5.Name = "tableLayoutPanel5";
            this.tableLayoutPanel5.RowCount = 1;
            this.tableLayoutPanel5.RowStyles.Add(new
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Percent,
50F));
            this.tableLayoutPanel5.Size = new System.Drawing.Size(690,
62);
            this.tableLayoutPanel5.TabIndex = 1;
            //
            // lbDate
            //
            this.lbDate.Location = new System.Drawing.Point(417, 0);
            this.lbDate.Name = "lbDate";
            this.lbDate.Size = new System.Drawing.Size(204, 39);
            this.lbDate.TabIndex = 7;
            this.lbDate.Text = "  ";
            this.lbDate.TextAlign =
System.Drawing.ContentAlignment.MiddleLeft;
```

```
            //
            // label18
            //
            this.label18.Location = new System.Drawing.Point(363, 0);
            this.label18.Name = "label18";
            this.label18.Size = new System.Drawing.Size(45, 39);
            this.label18.TabIndex = 5;
            this.label18.Text = "วันที่";
            this.label18.TextAlign =
System.Drawing.ContentAlignment.MiddleCenter;
            //
            // label17
            //
            this.label17.Location = new System.Drawing.Point(3, 0);
            this.label17.Name = "label17";
            this.label17.Size = new System.Drawing.Size(67, 39);
            this.label17.TabIndex = 4;
            this.label17.Text = "ผู้รับเงิน";
            this.label17.TextAlign =
System.Drawing.ContentAlignment.MiddleCenter;
            //
            // btnPrint
            //
            this.btnPrint.Location = new System.Drawing.Point(226,
440);
            this.btnPrint.Name = "btnPrint";
            this.btnPrint.Size = new System.Drawing.Size(75, 23);
            this.btnPrint.TabIndex = 2;
            this.btnPrint.Text = "พิมพ์";
            this.btnPrint.UseVisualStyleBackColor = true;
            this.btnPrint.Click += new
System.EventHandler(this.btnPrint_Click);
            //
            // btnBack
            //
            this.btnBack.Location = new System.Drawing.Point(433, 440);
            this.btnBack.Name = "btnBack";
            this.btnBack.Size = new System.Drawing.Size(75, 23);
            this.btnBack.TabIndex = 3;
            this.btnBack.Text = "กลับหน้าแรก";
            this.btnBack.UseVisualStyleBackColor = true;
            this.btnBack.Click += new
System.EventHandler(this.btnBack_Click);
            //
            // Print
            //
            this.AutoScaleBaseSize = new System.Drawing.Size(5, 13);
            this.ClientSize = new System.Drawing.Size(728, 510);
            this.Controls.Add(this.btnBack);
            this.Controls.Add(this.btnPrint);
            this.Controls.Add(this.tableLayoutPanel5);
            this.Controls.Add(this.tableLayoutPanel1);
            this.Name = "Print";
            this.Text = "พิมพ์";
            this.Load += new System.EventHandler(this.Print_Load);
            this.tableLayoutPanel1.ResumeLayout(false);
```

```
            this.tableLayoutPanel2.ResumeLayout(false);
            this.tableLayoutPanel3.ResumeLayout(false);
            this.tableLayoutPanel6.ResumeLayout(false);
            this.tableLayoutPanel4.ResumeLayout(false);
            this.tableLayoutPanel5.ResumeLayout(false);
            this.ResumeLayout(false);

            }
            #endregion

        private void btnBack_Click(object sender, EventArgs e)
        {
            this.Owner.Show();
            this.Hide();
        }
        private void Print_Load(object sender, EventArgs e)
        {
            myConnection = new
SqlConnection(DataClass.CommonConnection);

            //Get User
            SqlDataAdapter SelectUser = new SqlDataAdapter("select
name_member from member where status =1 and rec_no_room =" + strRoom,
myConnection);
            dsUser = new DataSet();
            SelectUser.Fill(dsUser, "User");

            //Get Last Data of Room
            SqlDataAdapter SelectMax = new SqlDataAdapter("select
max(rec_no_jrnl) from journal where rec_date like '%28%' and
rec_no_room = " + strRoom, myConnection);
            dsMax = new DataSet();
            SelectMax.Fill(dsMax, "LastData");

            //Get Last UnitMeter Old
            SqlDataAdapter SelectJrnl = new SqlDataAdapter("select
meter_unit_new from journal where status = 1 and rec_no_jrnl = " +
dsMax.Tables["LastData"].Rows[0][0], myConnection);
            dsJrnl = new DataSet();
            SelectJrnl.Fill(dsJrnl, "LastJrnl");

           strUser = dsUser.Tables["User"].Rows[0][0].ToString();
           d_OldMeter =
Convert.ToDecimal(dsJrnl.Tables["LastJrnl"].Rows[0][0]);

            d_User = (d_NewMeter - d_OldMeter);
            d_Price = (d_UnitPrice * d_User);
            d_RentWater = (DataClass.CommonRent+d_Price);

            //ใส่ค่าที่ได้ลงที่ Labelแต่ละตัว
            lbRoom.Text = strRoom;
            lbName.Text = strUser;
            lbOldUnit.Text = d_OldMeter.ToString();
            lbNewUnit.Text = d_NewMeter.ToString();
            lbUse.Text = d_User.ToString();
            lbPrice.Text = d_Price.ToString();
            lbDate.Text = DateTime.Now.ToLongDateString();
```

```
            lbRent.Text = DataClass.CommonRent.ToString();
            lbRentWater.Text = d_RentWater.ToString();
            lbUnit.Text = DataClass.CommonUnit.ToString() + " บาท";
        }
        private void btnPrint_Click(object sender, EventArgs e)
        {
            btnPrint.Enabled = false;
            InsertJrnl();
            PrePrint();

            this.Owner.Show();
            this.Hide();
        }
        private void InsertJrnl()
        {
            int iRecord;
            SqlDataAdapter SelectCus = new SqlDataAdapter("select
Max(rec_no_jrnl) from journal", myConnection);
            ds = new DataSet();
            SelectCus.Fill(ds, "Generate");

            if (ds.Tables["Generate"].Rows[0][0] !=
System.DBNull.Value)                    // Check status ID
            {
                iRecord =
Convert.ToInt32(ds.Tables["Generate"].Rows[0][0]);
                iRecord = iRecord + 1;
            }
            else
            {
                iRecord = 1;
            }

            string insertCmd = "insert into
Journal(rec_no_jrnl,rec_no_room,meter_unit_old,meter_unit_new,water_use
,price,status,rec_date,user_name,rent,unit) " +

"values(@rec_jrnl,@rec_room,@unit_old,@unit_new,@use,@price,@status,@da
te,@user_name,@rent,@unit)";

            SqlCommand myCommand = new SqlCommand(insertCmd,
myConnection);

            myCommand.Parameters.Add(new SqlParameter("@rec_jrnl",
SqlDbType.Int, 4));
            myCommand.Parameters["@rec_jrnl"].Value = iRecord;

            myCommand.Parameters.Add(new SqlParameter("@rec_room",
SqlDbType.VarChar, 10));
            myCommand.Parameters["@rec_room"].Value = lbRoom.Text;

            myCommand.Parameters.Add(new SqlParameter("@unit_old",
SqlDbType.Int, 4));
            myCommand.Parameters["@unit_old"].Value =
Convert.ToInt32(lbOldUnit.Text);
```

```
                myCommand.Parameters.Add(new SqlParameter("@unit_new",
SqlDbType.Int, 4));
                myCommand.Parameters["@unit_new"].Value =
Convert.ToInt32(lbNewUnit.Text);

                myCommand.Parameters.Add(new SqlParameter("@use",
SqlDbType.Decimal, 9));
                myCommand.Parameters["@use"].Value =
Convert.ToDecimal(lbUse.Text);

                myCommand.Parameters.Add(new SqlParameter("@price",
SqlDbType.Decimal, 9));
                myCommand.Parameters["@price"].Value =
Convert.ToDecimal(lbPrice.Text);

                myCommand.Parameters.Add(new SqlParameter("@status",
SqlDbType.Int, 4));
                myCommand.Parameters["@status"].Value = 1;

                myCommand.Parameters.Add(new SqlParameter("@date",
SqlDbType.DateTime, 20));
                myCommand.Parameters["@date"].Value = lbDate.Text;

                myCommand.Parameters.Add(new SqlParameter("@user_name",
SqlDbType.VarChar, 50));
                myCommand.Parameters["@user_name"].Value = strUser;

                myCommand.Parameters.Add(new SqlParameter("@rent",
SqlDbType.Decimal, 9));
                myCommand.Parameters["@rent"].Value = lbRent.Text;

                myCommand.Parameters.Add(new SqlParameter("@unit",
SqlDbType.Decimal, 9));
                myCommand.Parameters["@unit"].Value = DataClass.CommonUnit;

                myCommand.Connection.Open();
                try
                {
                    myCommand.ExecuteNonQuery();
                }
                catch (SqlException)
                {
                    MessageBox.Show("Cann't Print  Error At Insert Jrnl");
                }
                myCommand.Connection.Close();
            }
            private void PrePrint()
            {
                //กำหนดตัวแปร strDate เพื่อใช้ในการเลือกค่า
                string strMonth = DateTime.Now.Month.ToString();
                if (strMonth.Length == 1)
                    strMonth = "0" + strMonth;
                string strDate = DateTime.Now.Year.ToString() + strMonth +
DateTime.Now.Day.ToString();
```

```
            SqlDataAdapter SelectCus = new SqlDataAdapter("select
max(rec_no_jrnl) from journal where rec_no_room = '"+strRoom+"'",
myConnection);
            ds = new DataSet();
            SelectCus.Fill(ds, "Generate");

            int iRec_jrnl =
Convert.ToInt32(ds.Tables["Generate"].Rows[0][0]);

            //select Journal for Print
            SqlDataAdapter test = new SqlDataAdapter();

            SqlCommand commandhead = new SqlCommand();
            commandhead.CommandType = CommandType.Text;
            commandhead.Connection = myConnection;
            commandhead.CommandText = "select * from journal where
rec_no_jrnl = " + iRec_jrnl + " and rec_no_room = '" + strRoom + "' and
rec_date='" + strDate + "'";

            ////commandhead/////
            test.SelectCommand = commandhead;//SelectCommand ใช้เพื่อเลือกว่าจะรัน
คำสั่งไหน

            PrintSet bill = new PrintSet();

            try
            {
                myConnection.Open();
                test.Fill(bill, "PrintBill");
                myConnection.Close();
            }
            catch (Exception e)
            {
                MessageBox.Show(e.Message);
            }


            //Print ////
            PrintBill print = new PrintBill();          //File Crystal
Report
            print.SetDataSource(bill);  //กำหนดให้เอาข้อมูลจาก DataSet ชื่อ bill
            print.PrintToPrinter(1, false, 0, 0);     //Print ออกเครื่องพิมพ์หลัก
        }
    }
}
```

```
################################ Room ################################
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;
using System.Data.SqlClient;

namespace WindowsApplication1
{
	/// <summary>
	/// Summary description for Room.
	/// </summary>
	public class Room : System.Windows.Forms.Form
	{
		public System.Windows.Forms.Label label1;
		private System.Windows.Forms.TextBox txRoom;
		private System.Windows.Forms.Button btnOK;
		private System.Windows.Forms.Button btnCancle;
		/// <summary>
		/// Required designer variable.
		/// </summary>
		private System.ComponentModel.Container components = null;
		SqlConnection myConnection ;
		//DataSet ds ;

		public Room()
		{
			//
			// Required for Windows Form Designer support
			//
			InitializeComponent();

			//
			// TODO: Add any constructor code after
InitializeComponent call
			//
		}

		/// <summary>
		/// Clean up any resources being used.
		/// </summary>
		protected override void Dispose( bool disposing )
		{
			if( disposing )
			{
				if(components != null)
				{
					components.Dispose();
				}
			}
			base.Dispose( disposing );
		}

		#region Windows Form Designer generated code
		/// <summary>
```

```
            /// Required method for Designer support - do not modify
            /// the contents of this method with the code editor.
            /// </summary>
            private void InitializeComponent()
            {
            this.label1 = new System.Windows.Forms.Label();
            this.txRoom = new System.Windows.Forms.TextBox();
            this.btnOK = new System.Windows.Forms.Button();
            this.btnCancle = new System.Windows.Forms.Button();
            this.SuspendLayout();
            //
            // label1
            //
            this.label1.Font = new System.Drawing.Font("Microsoft Sans
Serif", 9F, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.label1.Location = new System.Drawing.Point(0, 16);
            this.label1.Name = "label1";
            this.label1.Size = new System.Drawing.Size(120, 23);
            this.label1.TabIndex = 0;
            this.label1.Text = "กรุณาใส่หมายเลขห้อง";
            //
            // txRoom
            //
            this.txRoom.Location = new System.Drawing.Point(128, 16);
            this.txRoom.Name = "txRoom";
            this.txRoom.Size = new System.Drawing.Size(100, 20);
            this.txRoom.TabIndex = 1;
            this.txRoom.KeyDown += new
System.Windows.Forms.KeyEventHandler(this.txRoom_KeyDown);
            //
            // btnOK
            //
            this.btnOK.Font = new System.Drawing.Font("Microsoft Sans
Serif", 9F, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.btnOK.Location = new System.Drawing.Point(40, 48);
            this.btnOK.Name = "btnOK";
            this.btnOK.Size = new System.Drawing.Size(75, 23);
            this.btnOK.TabIndex = 2;
            this.btnOK.Text = "O.K.";
            this.btnOK.Click += new
System.EventHandler(this.btnOK_Click);
            //
            // btnCancle
            //
            this.btnCancle.Font = new System.Drawing.Font("Microsoft
Sans Serif", 9F, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.btnCancle.Location = new System.Drawing.Point(136,
48);
            this.btnCancle.Name = "btnCancle";
            this.btnCancle.Size = new System.Drawing.Size(75, 23);
            this.btnCancle.TabIndex = 3;
            this.btnCancle.Text = "Cancle";
            this.btnCancle.Click += new
System.EventHandler(this.btnCancle_Click);
```

```
			//
			// Room
			//
			this.AutoScaleBaseSize = new System.Drawing.Size(5, 13);
			this.ClientSize = new System.Drawing.Size(240, 78);
			this.Controls.Add(this.btnCancle);
			this.Controls.Add(this.btnOK);
			this.Controls.Add(this.txRoom);
			this.Controls.Add(this.label1);
			this.Name = "Room";
			this.StartPosition =
System.Windows.Forms.FormStartPosition.CenterScreen;
			this.Text = "กำหนดห้องที่ต้องการ";
			this.Closed += new System.EventHandler(this.Room_Closed);
			this.Activated += new
System.EventHandler(this.Room_Activated);
			this.Load += new System.EventHandler(this.Room_Load);
			this.ResumeLayout(false);
			this.PerformLayout();

		}
		#endregion

		private void btnCancle_Click(object sender,
System.EventArgs e)
		{
			this.Owner.Show();
			this.Hide();
		}

		private void btnOK_Click(object sender, System.EventArgs e)
		{
			GetRoom();
		}

		private void Room_Closed(object sender, System.EventArgs e)
		{
				Application.Exit();
		}

		private void Room_Load(object sender, System.EventArgs e)
		{
				myConnection = new
SqlConnection(DataClass.CommonConnection);
		}

		private void txRoom_KeyDown(object sender,
System.Windows.Forms.KeyEventArgs e)
		{
			if(e.KeyValue == 13)	//Check การกด Enter
			{
				GetRoom();
			}
		}
		private void GetRoom()
		{
```

```
                        SqlDataAdapter myCommand = new SqlDataAdapter("select
rec_no_room  from Member where status =1 and rec_no_room
="+txRoom.Text, myConnection);
                        DataSet ds = new DataSet();
                        myCommand.Fill(ds, "Room");

                        if(ds.Tables["Room"].Rows.Count != 0)
                        {
                                DataClass.CommonRoom = txRoom.Text;
                                MessageBox.Show("สามารถทำรายการต่างๆของห้อง  "+txRoom.Text
+" ได้แล้วครับ");

                                this.Owner.Show();
                                this.Hide();
                        }
                        else
                        {
                                MessageBox.Show("ไม่มีห้องนี้อยู่ในระบบ");
                                txRoom.Clear();
                                txRoom.Focus();
                        }
                }

                private void Room_Activated(object sender, System.EventArgs
e)
                {

                }
        }
}
```

โปรแกรมส่วนแสดงผลทาง LCD โมดูล

```
#include <reg51.h>
sbit  e     =     P3^7;
sbit  rs    =     P3^5;
sbit pulse=P3^0;
code unsigned char text1[]="00 UNIT ";
code unsigned char text2[]="WATER = ";
void delay (int tick)
{
      int i,j;
      for (i=0;i<tick;i++)
      for   (j=0;j<250;j++);
}
void lcd_text(unsigned char text)
{
      rs = 1;
      e = 1;
      P1=text;
      delay(10);
      e = 0;
      delay(10);
}
void lcd_init()
{
      delay(500);
      lcd_command(0X38);
      lcd_command(0X0C);
      lcd_command(0X01);
}
void putchar(unsigned char outdata)
{
    SBUF = outdata;
      while(~TI);
      TI = 0;
}
void delay_s(unsigned char s )
{   unsigned int t,k;
    TH0 =0X00;
    t = s*3920; k = 0;
    TR0=1;   //start counter
    while(k < t)   // delay 5 ms  1 machine = 12/11.0592MHz = 1.08 us
      {  if(TF0)  // TH0=255  1.08us*255=255 us
         { k++; TF0=0; } //  5/255us =  19600
      }
      TR0=0; // stop counter
}
void delay_ms(unsigned char s )
{   unsigned int t,k;
    TH0 =0X00;
    t = s*4; k = 0;
    TR0=1;   //start counter
    while(k < t)   // delay 5 ms  1 machine = 12/11.0592MHz = 1.08 us
```

```
    {   if(TF0)   // TH0=255  1.08us*255=255  us
        { k++; TF0=0; } //   5/255us =   19600
    }
    TR0=0; // stop counter
}
unsigned char getchar(void)
{     unsigned char dat;
    unsigned int  sum;
    bit k = 0;
      dat = 0;
      while(~RI)
      { sum++;
        delay_ms(1);
        if( sum > 3000)
        {
            k = 1;
            RI=1;
        }
      }
      RI=0;
      sum = 0;
      if(k==0)
      dat = SBUF;
      return(dat);
}
void smal(void)
{   char m,a,b;
    if(pulse==1)
    {  delay(5);
       if(pulse==1)
      {
       //sum=sum+1;if(sum>99) sum=0;
            a=(sum/10)+0X30;
          b=(sum%10)+0X30;
            lcd_command(0XC0);
            for(m=0;m<8;m++)
              { if(m==0)
               lcd_text(a);
                    else if(m==1)
                     lcd_text(b);
                     else
              lcd_text(text1[m]);
              }
              sum=sum+1;if(sum>99) sum=0;
              while(pulse);
       }}
}
void main(void)
{     unsigned char m,a,b;// small
        lcd_init();
        lcd_command(0X80);
        for(m=0;m<8;m++)
        lcd_text(text2[m]);
    a=0X30;
        b= 0X30;
```

```
            lcd_command(0xC0);
            for(m=0;m<8;m++)
              { if(m==0)
               lcd_text(a);
                   else if(m==1)
                    lcd_text(b);
                    else
              lcd_text(text1[m]);
         }
    delay(1000);
          while(1)
    {
     smal();


    }
}
```

โปรแกรมส่วน Master

```
#include <reg51.h>
sbit rr=P1^0;  //small
sbit tt=P1^1;
char sum1=0;
char sum2=0;
char sum3=0;
void senddata(char dd)
{       char x,y;
       x=(dd/10)+0x30;
         y=(dd%10)+0x30;
        TI = 0;
            SBUF = x;
            while(~TI);
            TI = 0;
            SBUF = y;
            while(~TI);
            TI = 0;
}
void tttt() interrupt 1
{  char e;
   e++;
   TF0=0;
    TH0=0x4b;
      TL0=0xfd;

      if(e>=40) {senddata(sum1);senddata(sum2);senddata(sum3);e=0;}
}
void delay (int tick)
{
      int i,j;
      for (i=0;i<tick;i++)
      for   (j=0;j<250;j++);
}
void pcc()
{
  if(rr==0 && tt==0)
    {
         delay(10);
       if(rr==0 && tt==0)
             {
                  sum1=sum1+1;

             }
             while(rr==0 && tt==0);

      }

      if(rr==0 && tt==1)
    {
       delay(10);
        if(rr==0 && tt==1)
            {
               sum2=sum2+1;
             }
         while(rr==0 && tt==1);
      }
```

```
        if(rr==1 && tt==0)
     {
          delay(10);
          if(rr==1 && tt==0)
               {
                sum3=sum3+1;
       }
           while(rr==1 && tt==0);
       }

  }
  void main(void)
  {
   unsigned char dat = 0;
    P3=0xff;
       TMOD = 0x21;

       SCON = 0x50;
       TH1 = 0xFD;
       TL1 = 0xFD;
       RI = 0;
       TH0=0x4b;
       TL0=0xfd;
       EA=1;ET0=1;TR0=0;
       TI = 0;
       TR1 = 1;
     rr=1;tt=1;

     while(1)
       {
          pcc();

              if(RI==1)
            {
               RI=0;
              dat = SBUF;
              if(dat==0x31)
              {
                  senddata(sum1);
               }
            if(dat==0x32)
              {
                  senddata(sum2);
               }
               //if(dat==0x33)
              if(dat==0x33)
              {
                  senddata(sum3);
               }
              }
       }
}
```

โปรแกรมส่วน Slave

```
#include <reg51.h>
sbit scan1=P2^0;
sbit scan2=P2^1;
sbit scan3=P2^2;
sbit send1=P2^3;
sbit send2=P2^4;
sbit send3=P2^5;
sbit rr=P3^0;
bit tt=P3^1;
char sum1=0;
char sum2=0;
char sum3=0;
char a1,a2,a3;
void delay (int tick)
{
      int i,j;
      for (i=0;i<tick;i++)
      for   (j=0;j<250;j++);
}
void bigg()
{
   if(scan1==1) {send1=1; sum1=sum1+1; rr=0;tt=0;delay(2);}
   else {send1=0;}

    if(scan2==1) {send2=1; sum2=sum2+1;rr=0;tt=1;delay(2);}
     else {send2=0;}

    if(scan3==1) {send3=1; sum3=sum3+1;rr=1;tt=0;delay(2);}
   else {send3=0;}
   //else {send1=0;send2=0;send3=0;rr=1;tt=1;}
}

void main(void)
{
      P3=0xff;

    send1=0;send2=0;send3=0;

             while(1)
      {

         bigg();
        }
}
```

Features

- Compatible with MCS®51 Products
- 2K/4K Bytes of In-System Programmable (ISP) Flash Program Memory
  - Serial Interface for Program Downloading
  - Endurance: 10,000 Write/Erase Cycles
- 2.7V to 5.5V Operating Range
- Fully Static Operation: 0 Hz to 24 MHz
- Two-level Program Memory Lock
- 256 x 8-bit Internal RAM
- 15 Programmable I/O Lines
- Two 16-bit Timer/Counters
- Six Interrupt Sources
- Programmable Serial UART Channel
- Direct LED Drive Outputs
- On-chip Analog Comparator with Selectable Interrupt
- 8-bit PWM (Pulse-width Modulation)
- Low Power Idle and Power-down Modes
- Brownout Reset
- Enhanced UART Serial Port with Framing Error Detection and Automatic Address Recognition
- Internal Power-on Reset
- Interrupt Recovery from Power-down Mode
- Programmable and Fuseable x2 Clock Option
- Four-level Enhanced Interrupt Controller
- Power-off Flag
- Flexible Programming (Byte and Page Modes)
  - Page Mode: 32 Bytes/Page
- User Serviceable Signature Page (32 Bytes)

8-bit Microcontroller with 2K/4K Bytes Flash

AT89S2051
AT89S4051

Preliminary

1. Description

The AT89S2051/S4051 is a low-voltage, high-performance CMOS 8-bit microcontroller with 2K/4K bytes of In-System Programmable (ISP) Flash program memory. The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard MCS-51 instruction set. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89S2051/S4051 is a powerful microcontroller which provides a highly-flexible and cost-effective solution to many embedded control applications. Moreover, the AT89S2051/S4051 is designed to be function compatible with the AT89C2051/C4051 devices, respectively.

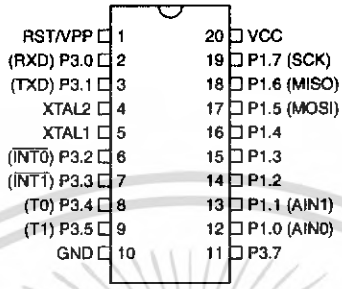
The AT89S2051/S4051 provides the following standard features: 2K/4K bytes of Flash, 256 bytes of RAM, 15 I/O lines, two 16-bit timer/counters, a six-vector, four-level interrupt architecture, a full duplex enhanced serial port, a precision analog comparator, on-chip and clock circuitry. Hardware support for PWM with 8-bit resolution and 8-bit prescaler is available by reconfiguring the two on-chip timer/counters. In addition, the AT89S2051/S4051 is designed with static logic for operation down to zero frequency and supports two software-selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port and interrupt system to continue functioning. The power-down mode saves the RAM contents but freezes the disabling all other chip functions until the next external interrupt or hardware reset.

3390C–MICRO–7/05

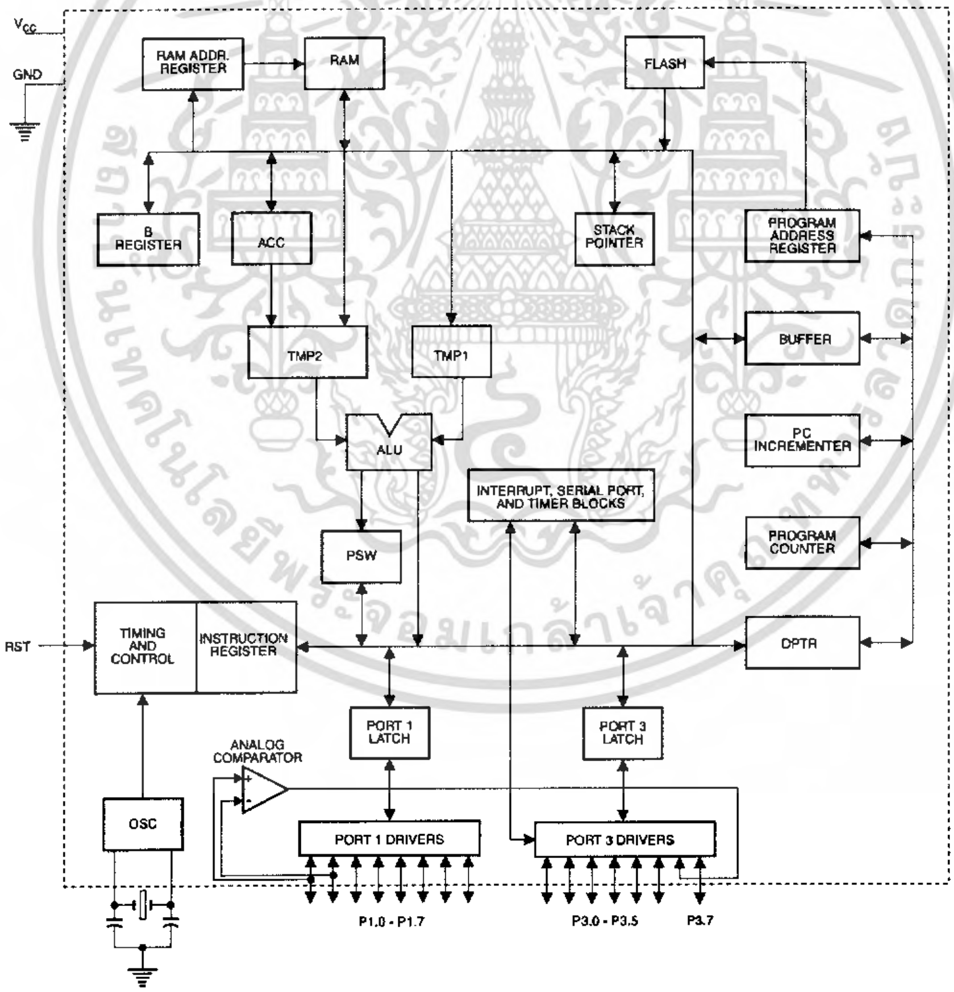
The on-board Flash program memory is accessible through the ISP serial interface. Holding RST active forces the device into a serial programming interface and allows the program memory to be written to or read from, unless one or more lock bits have been activated.

2. Pin Configuration

2.1 20-lead PDIP/SOIC



3. Block Diagram



4. Pin Description

4.1 VCC

Supply voltage.

4.2 GND

Ground.

4.3 Port 1

Port 1 is an 8-bit bi-directional I/O port. Port pins P1.2 to P1.7 provide internal pull-ups. P1.0 and P1.1 require external pull-ups. P1.0 and P1.1 also serve as the positive input (AIN0) and the negative input (AIN1), respectively, of the on-chip precision analog comparator. The Port 1 output buffers can sink 20 mA and can drive LED displays directly. When 1s are written to Port 1 pins, they can be used as inputs. When pins P1.2 to P1.7 are used as inputs and are externally pulled low, they will source current (I_{IL}) because of the internal pull-ups.

Port 1 also receives code data during Flash programming and verification.

Port Pin	Alternate Functions
P1.5	MOSI (Master data output, slave data input pin for ISP channel)
P1.6	MISO (Master data input, slave data output pin for ISP channel)
P1.7	SCK (Master clock output, slave clock input pin for ISP channel)

4.4 Port 3

Port 3 pins P3.0 to P3.5, P3.7 are seven bi-directional I/O pins with internal pull-ups. P3.6 is hard-wired as an input to the output of the on-chip comparator and is not accessible as a general-purpose I/O pin. The Port 3 output buffers can sink 20 mA. When 1s are written to Port 3 pins they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}) because of the pull-ups.

Port 3 also serves the functions of various special features of the AT89S2051/S4051 as listed below:

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{\text{INT0}}$ (external interrupt 0)
P3.3	$\overline{\text{INT1}}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)/ PWM output

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

4.5 RST

Reset input. Holding the RST pin high for two machine cycles while the is running resets the device.

Each machine cycle takes 6 or clock cycles.

4.6 XTAL1

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

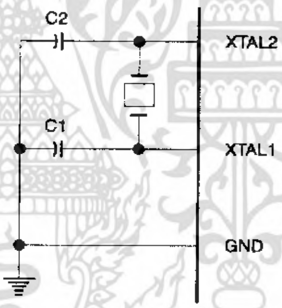
4.7 XTAL2

Output from the inverting amplifier.

5. Characteristics

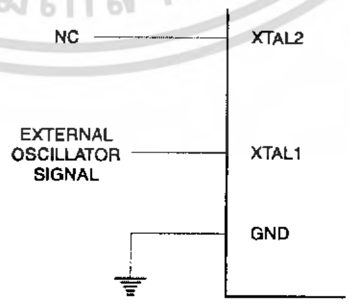
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip , as shown in Figure 5-1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 5-2. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Figure 5-1. Connections



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

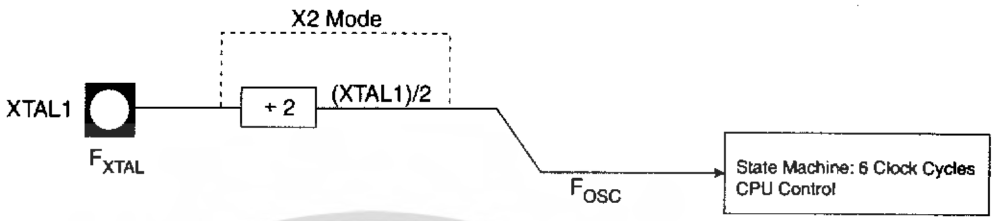
Figure 5-2. External Clock Drive Configuration



6. X2 Mode Description

The clock for the entire circuit and peripherals is normally divided by 2 before being used by the CPU core and peripherals. This allows any cyclic ratio (duty cycle) to be accepted on XTAL1 input. In X2 mode this divider is bypassed. Figure 6-1 shows the clock generation block diagram.

Figure 6-1. Clock Generation Block Diagram



7. Special Function Registers

A map of the on-chip memory area called the Special Function Register (SFR) space is shown in Table 7-1.

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

User software should not write 1s to these unlisted locations, since they may be used in future products to invoke new features. In that case, the reset or inactive values of the new bits will always be 0.

Table 7-1. AT89S2051/S4051 SFR Map and Reset Values

0F8H									0FFH
0F0H	B 00000000								0F7H
0E8H									0EFH
0E0H	ACC 00000000								0E7H
0D8H									0DFH
0D0H	PSW 00000000								0D7H
0C8H									0CFH
0C0H									0C7H
0B8H	IP X0X00000	SADEN 00000000							0BFH
0B0H	P3 11111111							IPH X0X00000	0B7H
0A8H	IE 00X00000	SADDR 00000000							0AFH
0A0H									0A7H
98H	SCON 00000000	SBUF XXXXXXXX							9FH
90H	P1 11111111							ACSR XXX00000	97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000		CLKREG XXXXXX0X	8FH
80H		SP 00000111	DPL 00000000	DPH 00000000				PCON 000X0000	87H

8. Restrictions on Certain Instructions

The AT89S2051/S4051 is an economical and cost-effective member of Atmel's family of microcontrollers. It contains 2K/4K bytes of Flash program memory. It is fully compatible with the MCS-51 architecture, and can be programmed using the MCS-51 instruction set. However, there are a few considerations one must keep in mind when utilizing certain instructions to program this device.

All the instructions related to jumping or branching should be restricted such that the destination address falls within the physical program memory space of the device, which is 2K/4K for the AT89S2051/S4051. This should be the responsibility of the software programmer. For example, LJMP 7E0H would be a valid instruction for the AT89S2051 (with 2K of memory), whereas LJMP 900H would not.

8.1 Branching Instructions

LCALL, LJMP, ACALL, AJMP, SJMP, JMP @A+DPTR. These unconditional branching instructions will execute correctly as long as the programmer keeps in mind that the destination branching address must fall within the physical boundaries of the program memory size (locations 00H to 7FFH/FFFH for the AT89S2051/S4051). Violating the physical space limits may cause unknown program behavior.

CJNE [...], DJNZ [...], JB, JNB, JC, JNC, JBC, JZ, JNZ. With these conditional branching instructions, the same rule above applies. Again, violating the memory boundaries may cause erratic execution.

For applications involving interrupts, the normal interrupt service routine address locations of the 80C51 family architecture have been preserved.

8.2 MOVX-related Instructions, Data Memory

The AT89S2051/S4051 contains 256 bytes of internal data memory. External DATA memory access is not supported in this device, nor is external PROGRAM memory execution. Therefore, no MOVX [...] instructions should be included in the program.

A typical 80C51 assembler will still assemble instructions, even if they are written in violation of the restrictions mentioned above. It is the responsibility of the user to know the physical features and limitations of the device being used and adjust the instructions used accordingly.

9. Program Memory Lock Bits

On the chip are two lock bits which can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in Table 9-1:

Table 9-1. Lock Bit Protection Modes(1)

Program Lock Bits			
	LB1	LB2	Protection Type
1	U	U	No program lock features.
2	P	U	Further programming of the Flash is disabled.
3	P	P	Same as mode 2, also verify is disabled.

Note: 1. The Lock Bits can only be erased with the Chip Erase operation.

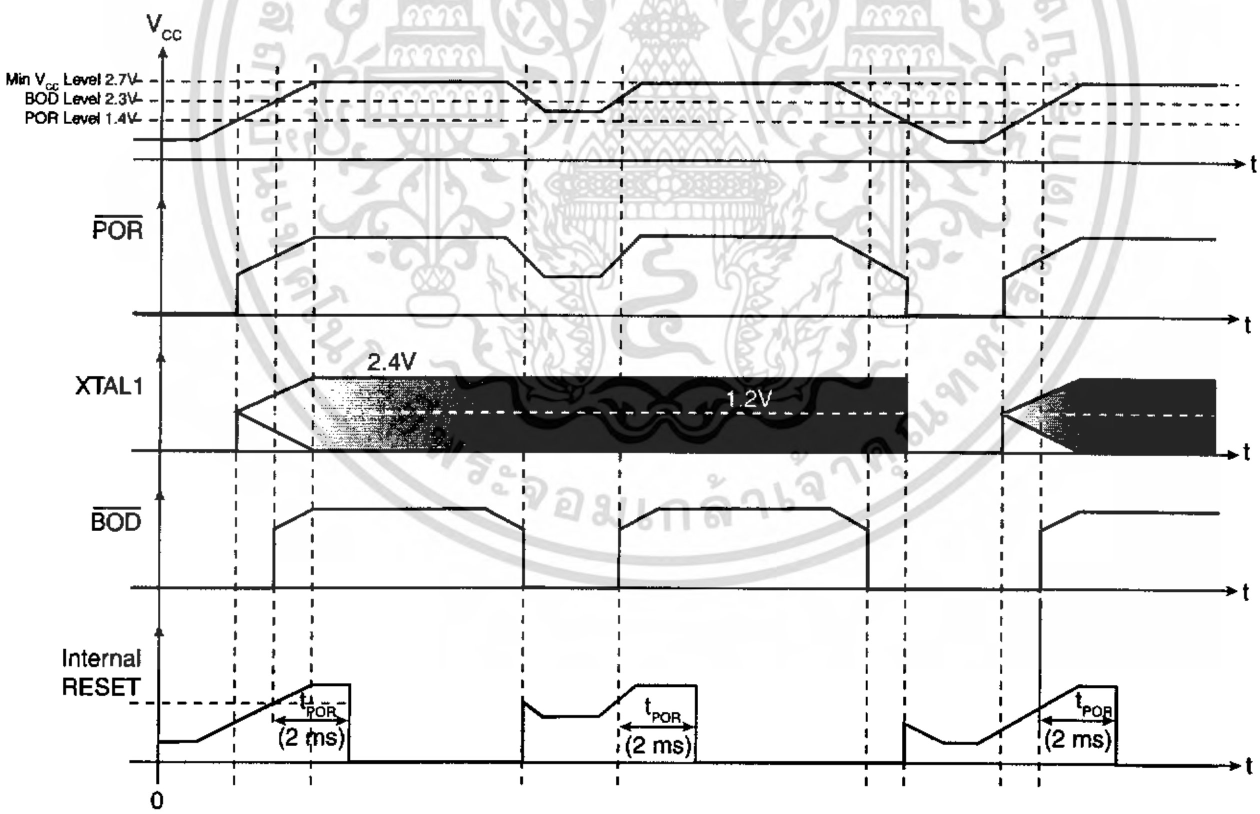
10. Reset

During reset, all I/O Registers are set to their initial values, the port pins are weakly pulled to V_{CC}, and the program starts execution from the Reset Vector, 0000H. The AT89S2051/S4051 has three sources of reset: power-on reset, brown-out reset, and external reset.

10.1 Power-On Reset

A Power-On Reset (POR) is generated by an on-chip detection circuit. The detection level is nominally 1.4V. The POR is activated whenever V_{CC} is below the detection level. The POR circuit can be used to trigger the start-up reset or to detect a supply voltage failure in devices without a brown-out detector. The POR circuit ensures that the device is reset from power-on. When V_{CC} reaches the Power-on Reset threshold voltage, the Pierce Oscillator is enabled (if the XTAL Oscillator Bypass fuse is OFF). Only after V_{CC} has also reached the BOD (brown-out detection) level (see Section 10.2 "Brown-out Reset"), the BOD delay counter starts measuring a 2-ms delay after which the Internal Reset is deasserted and the microcontroller starts executing. The built-in 2-ms delay allows the V_{CC} voltage to reach the minimum 2.7V level before executing, thus guaranteeing the maximum operating clock frequency. The POR signal is activated again, without any delay, when V_{CC} falls below the POR threshold level. A Power-On Reset (i.e. a cold reset) will set the POF flag in PCON. Refer to Figure 10-1 for details on the POR/BOD behavior.

Figure 10-1. Power-up and Brown-out Detection Sequence



10.2 Brown-out Reset

The AT89S2051/S4051 has an on-chip Brown-out Detection (BOD) circuit for monitoring the V_{CC} level during operation by comparing it to a fixed trigger level. The trigger level for the BOD is nominally 2.2V. The purpose of the BOD is to ensure that if V_{CC} fails or dips while executing at speed, the system will gracefully enter reset without the possibility of errors induced by incorrect execution. When V_{CC} decreases to a value below the trigger level, the Brown-out Reset is immediately activated. When V_{CC} increases above the trigger level, the BOD delay counter starts the microcontroller after the timeout period has expired in approximately 2 ms.

10.3 External Reset

The RST pin functions as an active-high reset input. The pin must be held high for at least two machine cycles to trigger the internal reset. RST also serves as the In-System Programming (ISP) enable input. ISP mode is enabled when the external reset pin is held high and the ISP Enable fuse is set.

11. Clock Register

Table 11-1. CLKREG – Clock Register

CLKREG = 8FH — Reset Value = XXXX XX0XB

Not Bit Addressable

	–	–	–	–	–	–	PWDEX	X2
Bit	7	6	5	4	3	2	1	0

Symbol	Function
PWDEX	Power-down Exit Mode. When PWDEX = 1, wake up from Power-down is externally controlled. When PWDEX = 0, wake up from Power-down is internally timed.
X2	When X2 = 0, the frequency (at XTAL1 pin) is internally divided by 2 before it is used as the device system frequency. When X2 = 1, the divide by 2 is no longer used and the XTAL1 frequency becomes the device system frequency. This enables the user to use a 6 MHz crystal instead of a 12 MHz crystal in order to reduce EMI. The X2 bit is initialized on power-up with the value of the X2 user fuse and may be changed at runtime by software.

12. Power Saving Modes

The AT89S2051/S4051 supports two power-reducing modes: Idle and Power-down. These modes are accessed through the PCON register.

12.1 Idle Mode

Setting the IDL bit in PCON enters idle mode. Idle mode halts the internal CPU clock. The CPU state is preserved in its entirety, including the RAM, stack pointer, program counter, program status word, and accumulator. The Port pins hold the logical states they had at the time that Idle was activated. Idle mode leaves the peripherals running in order to allow them to wake up the CPU when an interrupt is generated. Timer 0, Timer 1, and the UART will continue to function during Idle mode. The analog comparator is disabled during Idle. Any enabled interrupt source or reset may terminate Idle mode. When exiting Idle mode with an interrupt, the interrupt will immediately be serviced, and following RETI, the next instruction to be executed will be the one following the instruction that put the device into Idle.

P1.0 and P1.1 should be set to "0" if no external pull-ups are used, or set to "1" if external pull-ups are used.

12.2 Power-down Mode

Setting the PD bit in PCON enters Power-down mode. Power-down mode stops the and powers down the Flash memory in order to minimize power consumption. Only the power-on circuitry will continue to draw power during Power-down. During Power-down the power supply voltage may be reduced to the RAM keep-alive voltage. The RAM contents will be retained; however, the SFR contents are not guaranteed once V_{CC} has been reduced. Power-down may be exited by external reset, power-on reset, or certain interrupts.

The user should not attempt to enter (or re-enter) the power-down mode for a minimum of 4 µs until after one of the following conditions has occurred: Start of code execution (after any type of reset), or Exit from power-down mode.

12.3 Interrupt Recovery from Power-down

Two external interrupts may be configured to terminate Power-down mode. External interrupts $\overline{\text{INT0}}$ (P3.2) and $\overline{\text{INT1}}$ (P3.3) may be used to exit Power-down. To wake up by external interrupt $\overline{\text{INT0}}$ or $\overline{\text{INT1}}$, the interrupt must be enabled and configured for level-sensitive operation.

When terminating Power-down by an interrupt, two different wake up modes are available. When PWDEX in CLKREG.2 is zero, the wake up period is internally timed. At the falling edge on the interrupt pin, Power-down is exited, the is restarted, and an internal timer begins counting. The internal clock will not be allowed to propagate and the CPU will not resume execution until after the timer has counted for nominally 2 ms. After the timeout period the interrupt service routine will begin. To prevent the interrupt from re-triggering, the ISR should disable the interrupt before returning. The interrupt pin should be held low until the device has timed out and begun executing.

When PWDEX = 1 the wakeup period is controlled externally by the interrupt. Again, at the falling edge on the interrupt pin, Power-down is exited and the is restarted. However, the internal clock will not propagate and CPU will not resume execution until the **rising edge** of the interrupt pin. After the rising edge on the pin, the interrupt service routine will begin. The interrupt should be held low long enough for the to stabilize.

12.4 Reset Recovery from Power-down

Wakeup from Power-down through an external reset is similar to the interrupt with PWDEX = 0. At the rising edge of RST, Power-down is exited, the is restarted, and an internal timer begins counting. The internal clock will not be allowed to propagate to the CPU until after the timer has counted for nominally 2 ms. The RST pin must be held high for longer than the timeout period to ensure that the device is reset properly. The device will begin executing once RST is brought low.

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

P1.0 and P1.1 should be set to "0" if no external pull-ups are used, or set to "1" if external pull-ups are used.

Table 12-1. PCON – Power Control Register

PCON = 87H — Reset Value = 000X 0000B

Not Bit Addressable

	SMOD1	SMOD0	PWMEN	POF	GF1	GF0	PD	IDL
Bit	7	6	5	4	3	2	1	0

Symbol	Function
SMOD1	Double Baud Rate bit. Doubles the baud rate of the UART in modes 1, 2, or 3.
SMOD0	Frame Error Select. When SMOD0 = 0, SCON.7 is SM0. When SMOD0 = 1, SCON.7 is FE. Note that FE will be set after a frame error regardless of the state of SMOD0.
PWMEN	Pulse Width Modulation Enable. When PWMEN = 1, Timer 0 and Timer 1 are configured as an 8-bit PWM counter with 8-bit auto-reload prescaler. The PWM outputs on T1 (P3.5).
POF	Power Off Flag. POF is set to "1" during power up (i.e. cold reset). It can be set or reset under software control and is not affected by RST or BOD (i.e. warm resets).
GF1, GF0	General-purpose Flags
PD	Power Down bit. Setting this bit activates power down operation.
IDL	Idle Mode bit. Setting this bit activates idle mode operation

13. Interrupts

The AT89S2051/S4051 provides 6 interrupt sources: two external interrupts, two timer interrupts, a serial port interrupt, and an analog comparator interrupt. These interrupts and the system reset each have a separate program vector at the start of the program memory space. Each interrupt source can be individually enabled or disabled by setting or clearing a bit in the interrupt enable register IE. The IE register also contains a global disable bit, EA, which disables all interrupts.

Each interrupt source can be individually programmed to one of four priority levels by setting or clearing bits in the interrupt priority registers IP and IPH. An interrupt service routine in progress can be interrupted by a higher priority interrupt, but not by another interrupt of the same or lower priority. The highest priority interrupt cannot be interrupted by any other interrupt source. If two requests of different priority levels are pending at the end of an instruction, the request of higher priority level is serviced. If requests of the same priority level are pending at the end of an instruction, an internal polling sequence determines which request is serviced. The polling sequence is based on the vector address; an interrupt with a lower vector address has higher priority than an interrupt with a higher vector address. Note that the polling sequence is only used to resolve pending requests of the same priority level.

The External Interrupts $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ can each be either level-activated or transition-activated, depending on bits IT0 and IT1 in Register TCON. The flags that actually generate these interrupts are the IE0 and IE1 bits in TCON. When the service routine is vectored to, hardware clears the flag that generated an external interrupt only if the interrupt was transition-activated. If the interrupt was level activated, then the external requesting source (rather than the on-chip hardware) controls the request flag.

The Timer 0 and Timer 1 Interrupts are generated by TF0 and TF1, which are set by a rollover in their respective Timer/Counter registers (except for Timer 0 in Mode 3). When a timer interrupt is generated, the on-chip hardware clears the flag that generated it when the service routine is vectored to.

The Serial Port Interrupt is generated by the logical OR of RI and TI in SCON. Neither of these flags is cleared by hardware when the service routine is vectored to. In fact, the service routine normally must determine whether RI or TI generated the interrupt, and the bit must be cleared in software.

The CF bit in ACSR generates the Comparator Interrupt. The flag is not cleared by hardware when the service routine is vectored to and must be cleared by software.

Most of the bits that generate interrupts can be set or cleared by software, with the same result as though they had been set or cleared by hardware. That is, interrupts can be generated and pending interrupts can be canceled in software.

Interrupt	Source	Vector Address
System Reset	RST or POR or BOD	0000H
External Interrupt 0	IE0	0003H
Timer 0 Overflow	TF0	000BH
External Interrupt 1	IE1	0013H
Timer 1 Overflow	TF1	001BH
Serial Port	RI or TI	0023H
Analog Comparator	CF	0033H

19-4323; Rev 10; 8/01

MAXIM

+5V-Powered, Multichannel RS-232 Drivers/Receivers

MAX220–MAX249

General Description

The MAX220–MAX249 family of line drivers/receivers is intended for all EIA/TIA-232E and V.28/V.24 communications interfaces, particularly applications where ±12V is not available.

These parts are especially useful in battery-powered systems, since their low-power shutdown mode reduces power dissipation to less than 5µW. The MAX225, MAX233, MAX235, and MAX245/MAX246/MAX247 use no external components and are recommended for applications where printed circuit board space is critical.

Applications

Portable Computers

Low-Power Modems

Interface Translation

Battery-Powered RS-232 Systems

Multidrop RS-232 Networks

Features

Superior to Bipolar

- ♦ **Operate from Single +5V Power Supply (+5V and +12V—MAX231/MAX239)**
- ♦ **Low-Power Receive Mode in Shutdown (MAX223/MAX242)**
- ♦ **Meet All EIA/TIA-232E and V.28 Specifications**
- ♦ **Multiple Drivers and Receivers**
- ♦ **3-State Driver and Receiver Outputs**
- ♦ **Open-Line Detection (MAX243)**

Ordering Information

PART	TEMP. RANGE	PIN-PACKAGE
MAX220CPE	0°C to +70°C	16 Plastic DIP
MAX220CSE	0°C to +70°C	16 Narrow SO
MAX220CWE	0°C to +70°C	16 Wide SO
MAX220C/D	0°C to +70°C	Dice*
MAX220EPE	-40°C to +85°C	16 Plastic DIP
MAX220ESE	-40°C to +85°C	16 Narrow SO
MAX220EWE	-40°C to +85°C	16 Wide SO
MAX220EJE	-40°C to +85°C	16 CERDIP
MAX220MJE	-55°C to +125°C	16 CERDIP

Ordering Information continued at end of data sheet.

**Contact factory for dice specifications.*

Selection Table

Part Number	Power Supply (V)	No. of RS-232 Drivers/Rx	No. of Ext. Caps	Nominal Cap. Value (µF)	SHDN & Three-State	Rx Active in SHDN	Data Rate (kbps)	Features
MAX220	+5	2/2	4	0.1	No	—	120	Ultra-low-power, industry-standard pinout
MAX222	+5	2/2	4	0.1	Yes	—	200	Low-power shutdown
MAX223 (MAX213)	+5	4/5	4	1.0 (0.1)	Yes	✔	120	MAX241 and receivers active in shutdown
MAX225	+5	5/5	0	—	Yes	✔	120	Available in SO
MAX230 (MAX200)	+5	5/0	4	1.0 (0.1)	Yes	—	120	5 drivers with shutdown
MAX231 (MAX201)	+5 and +7.5 to +13.2	2/2	2	1.0 (0.1)	No	—	120	Standard +5/+12V or battery supplies; same functions as MAX232
MAX232 (MAX202)	+5	2/2	4	1.0 (0.1)	No	—	120 (64)	Industry standard
MAX232A	+5	2/2	4	0.1	No	—	200	Higher slew rate, small caps
MAX233 (MAX203)	+5	2/2	0	—	No	—	120	No external caps
MAX233A	+5	2/2	0	—	No	—	200	No external caps, high slew rate
MAX234 (MAX204)	+5	4/0	4	1.0 (0.1)	No	—	120	Replaces 1488
MAX235 (MAX205)	+5	5/5	0	—	Yes	—	120	No external caps
MAX236 (MAX206)	+5	4/3	4	1.0 (0.1)	Yes	—	120	Shutdown, three state
MAX237 (MAX207)	+5	5/3	4	1.0 (0.1)	No	—	120	Complements IBM PC serial port
MAX238 (MAX208)	+5	4/4	4	1.0 (0.1)	No	—	120	Replaces 1488 and 1489
MAX239 (MAX209)	+5 and +7.5 to +13.2	3/5	2	1.0 (0.1)	No	—	120	Standard +5/+12V or battery supplies; single-package solution for IBM PC serial port
MAX240	+5	5/5	4	1.0	Yes	—	120	DIP or flatpack package
MAX241 (MAX211)	+5	4/5	4	1.0 (0.1)	Yes	—	120	Complete IBM PC serial port
MAX242	+5	2/2	4	0.1	Yes	✔	200	Separate shutdown and enable
MAX243	+5	2/2	4	0.1	No	—	200	Open-line detection simplifies cabling
MAX244	+5	8/10	4	1.0	No	—	120	High slew rate
MAX245	+5	8/10	0	—	Yes	✔	120	High slew rate, int. caps, two shutdown modes
MAX246	+5	8/10	0	—	Yes	✔	120	High slew rate, int. caps, three shutdown modes
MAX247	+5	8/9	0	—	Yes	✔	120	High slew rate, int. caps, nine operating modes
MAX248	+5	8/8	4	1.0	Yes	✔	120	High slew rate, selective half-chip enables
MAX249	+5	6/10	4	1.0	Yes	✔	120	Available in quad flatpack package

For pricing, delivery, and ordering information, please contact Maxim/Dallas Direct! at 1-888-629-4642, or visit Maxim's website at www.maxim-ic.com.

ABSOLUTE MAXIMUM RATINGS—MAX220/222/232A/233A/242/243

Supply Voltage (V_{CC}) ....-0.3V to +6V
Input Voltages
T_{IN} ....-0.3V to (V_{CC} - 0.3V)
R_{IN} (Except MAX220) ....±30V
R_{IN} (MAX220) ....±25V
T_{OUT} (Except MAX220) (Note 1) ....±15V
T_{OUT} (MAX220) ....±13.2V
Output Voltages
T_{OUT} ....±15V
R_{OUT} ....-0.3V to (V_{CC} + 0.3V)
Driver/Receiver Output Short Circuited to GND ....Continuous
Continuous Power Dissipation (T_A = +70°C)
16-Pin Plastic DIP (derate 10.53mW/°C above +70°C)....842mW
18-Pin Plastic DIP (derate 11.11mW/°C above +70°C)....889mW
20-Pin Plastic DIP (derate 8.00mW/°C above +70°C) ..440mW
16-Pin Narrow SO (derate 8.70mW/°C above +70°C)...696mW
16-Pin Wide SO (derate 9.52mW/°C above +70°C)......762mW
18-Pin Wide SO (derate 9.52mW/°C above +70°C)......762mW
20-Pin Wide SO (derate 10.00mW/°C above +70°C)....800mW
20-Pin SSOP (derate 8.00mW/°C above +70°C) 640mW
16-Pin CERDIP (derate 10.00mW/°C above +70°C).....800mW
18-Pin CERDIP (derate 10.53mW/°C above +70°C).....842mW
Operating Temperature Ranges
MAX2_ _AC_ _, MAX2_ _C_ _ ....0°C to +70°C
MAX2_ _AE_ _, MAX2_ _E_ _ ....-40°C to +85°C
MAX2_ _AM_ _, MAX2_ _M_ _ ....-55°C to +125°C
Storage Temperature Range ....-65°C to +160°C
Lead Temperature (soldering, 10sec) ....+300°C

Note 1: Input voltage measured with T_{OUT} in high-impedance state, $\overline{SHDN}$ or V_{CC} = 0V.
Note 2: For the MAX220, V+ and V- can have a maximum magnitude of 7V, but their absolute difference cannot exceed 13V.

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ELECTRICAL CHARACTERISTICS—MAX220/222/232A/233A/242/243

(V_{CC} = +5V ±10%, C1–C4 = 0.1µF, MAX220, C1 = 0.047µF, C2–C4 = 0.33µF, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
RS-232 TRANSMITTERS					
Output Voltage Swing	All transmitter outputs loaded with 3kΩ to GND	±5	±8		V
Input Logic Threshold Low			1.4	0.8	V
Input Logic Threshold High	All devices except MAX220	2	1.4		V
	MAX220: V_{CC} = 5.0V	2.4			
Logic Pull-Up/Input Current	All except MAX220, normal operation		5	40	µA
	$\overline{SHDN}$ = 0V, MAX222/242, shutdown, MAX220		±0.01	±1	
Output Leakage Current	V_{CC} = 5.5V, $\overline{SHDN}$ = 0V, V_{OUT} = ±15V, MAX222/242		±0.01	±10	µA
	V_{CC} = $\overline{SHDN}$ = 0V, V_{OUT} = ±15V		±0.01	±10	
Data Rate			200	116	kb/s
Transmitter Output Resistance	V_{CC} = V+ = V- = 0V, V_{OUT} = ±2V	300	10M		Ω
Output Short-Circuit Current	V_{OUT} = 0V	±7	±22		mA
RS-232 RECEIVERS					
RS-232 Input Voltage Operating Range				±30	V
RS-232 Input Threshold Low	V_{CC} = 5V, All except MAX243 $R2_{IN}$	0.8	1.3		V
	V_{CC} = 5V, MAX243 $R2_{IN}$ (Note 2)	-3			
RS-232 Input Threshold High	V_{CC} = 5V, All except MAX243 $R2_{IN}$		1.8	2.4	V
	V_{CC} = 5V, MAX243 $R2_{IN}$ (Note 2)		-0.5	-0.1	
RS-232 Input Hysteresis	All except MAX243, V_{CC} = 5V, no hysteresis in shdn.	0.2	0.5	1	V
	MAX243		1		
RS-232 Input Resistance		3	5	7	kΩ
TTL/CMOS Output Voltage Low	I_{OUT} = 3.2mA		0.2	0.4	V
TTL/CMOS Output Voltage High	I_{OUT} = -1.0mA	3.5	V_{CC} - 0.2		V
TTL/CMOS Output Short-Circuit Current	Sourcing V_{OUT} = GND	-2	-10		mA
	Shrinking V_{OUT} = V_{CC}	10	30		

ELECTRICAL CHARACTERISTICS—MAX220/222/232A/233A/242/243 (continued)

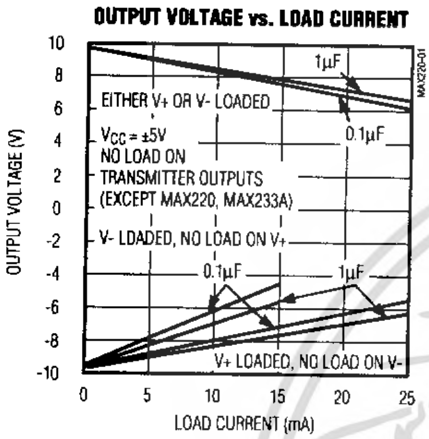
(V_{CC} = +5V ±10%, C1–C4 = 0.1μF, MAX220, C1 = 0.047μF, C2–C4 = 0.33μF, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

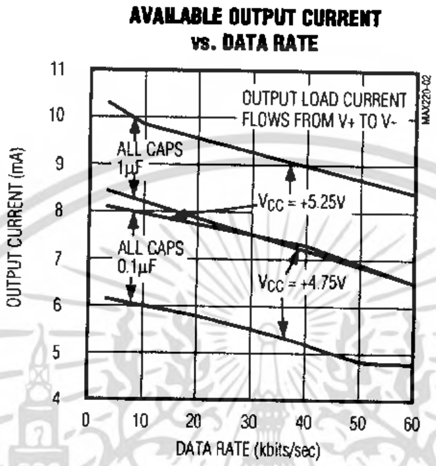
PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
TTL/CMOS Output Leakage Current	$\overline{SHDN}$ = V_{CC} or $\overline{EN}$ = V_{CC} ($\overline{SHDN}$ = 0V for MAX222), 0V ≤ V_{OUT} ≤ V_{CC}			±0.05	±10	μA
$\overline{EN}$ Input Threshold Low	MAX242			1.4	0.8	V
$\overline{EN}$ Input Threshold High	MAX242		2.0	1.4		V
Operating Supply Voltage			4.5		5.5	V
V_{CC} Supply Current ($\overline{SHDN}$ = V_{CC}), Figures 5, 6, 11, 19	No load	MAX220		0.5	2	mA
		MAX222/232A/233A/242/243		4	10	
	3kΩ load both inputs	MAX220		12		
		MAX222/232A/233A/242/243		15		
Shutdown Supply Current	MAX222/242	T_A = +25°C		0.1	10	μA
		T_A = 0°C to +70°C		2	50	
		T_A = -40°C to +85°C		2	50	
		T_A = -55°C to +125°C		35	100	
$\overline{SHDN}$ Input Leakage Current	MAX222/242				±1	μA
$\overline{SHDN}$ Threshold Low	MAX222/242			1.4	0.8	V
$\overline{SHDN}$ Threshold High	MAX222/242		2.0	1.4		V
Transition Slew Rate	C_L = 50pF to 2500pF, R_L = 3kΩ to 7kΩ, V_{CC} = 5V, T_A = +25°C, measured from +3V to -3V or -3V to +3V	MAX222/232A/233A/242/243	6	12	30	V/μs
		MAX220	1.5	3	30	
Transmitter Propagation Delay TLL to RS-232 (normal operation), Figure 1	t_{PHLT}	MAX222/232A/233A/242/243		1.3	3.5	μs
		MAX220		4	10	
	t_{PLHT}	MAX222/232A/233A/242/243		1.5	3.5	
		MAX220		5	10	
Receiver Propagation Delay RS-232 to TLL (normal operation), Figure 2	t_{PHLR}	MAX222/232A/233A/242/243		0.5	1	μs
		MAX220		0.6	3	
	t_{PLHR}	MAX222/232A/233A/242/243		0.6	1	
		MAX220		0.8	3	
Receiver Propagation Delay RS-232 to TLL (shutdown), Figure 2	t_{PHLS}	MAX242		0.5	10	μs
	t_{PLHS}	MAX242		2.5	10	
Receiver-Output Enable Time, Figure 3	t_{ER}	MAX242		125	500	ns
Receiver-Output Disable Time, Figure 3	t_{DR}	MAX242		160	500	ns
Transmitter-Output Enable Time ($\overline{SHDN}$ goes high), Figure 4	t_{ET}	MAX222/242, 0.1μF caps (includes charge-pump start-up)		250		μs
Transmitter-Output Disable Time ($\overline{SHDN}$ goes low), Figure 4	t_{DT}	MAX222/242, 0.1μF caps		600		ns
Transmitter + to - Propagation Delay Difference (normal operation)	t_{PHLT} - t_{PLHT}	MAX222/232A/233A/242/243		300		ns
		MAX220		2000		
Receiver + to - Propagation Delay Difference (normal operation)	t_{PHLR} - t_{PLHR}	MAX222/232A/233A/242/243		100		ns
		MAX220		225		

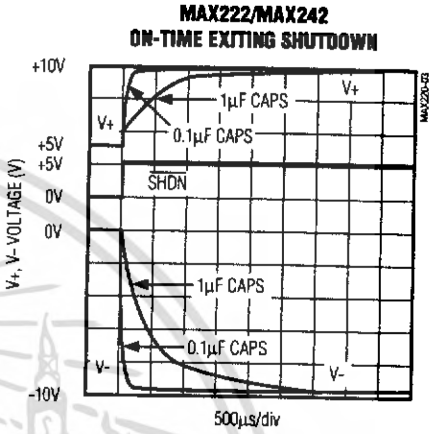
Note 3: MAX243 $R2_{OUT}$ is guaranteed to be low when $R2_{IN}$ is ≥ 0V or is floating.

Typical Operating Characteristics

MAX220/MAX222/MAX232A/MAX233A/MAX242/MAX243







ABSOLUTE MAXIMUM RATINGS—MAX223/MAX230–MAX241

V_{CC} .... -0.3V to +6V
V+ .... (V_{CC} - 0.3V) to +14V
V- .... +0.3V to -14V
Input Voltages
T_{IN} .... -0.3V to (V_{CC} + 0.3V)
R_{IN} .... ±30V
Output Voltages
T_{OUT} .... (V+ + 0.3V) to (V- - 0.3V)
R_{OUT} .... -0.3V to (V_{CC} + 0.3V)
Short-Circuit Duration, T_{OUT} .... Continuous
Continuous Power Dissipation (T_A = +70°C)
14-Pin Plastic DIP (derate 10.00mW/°C above +70°C).... 800mW
16-Pin Plastic DIP (derate 10.53mW/°C above +70°C).... 842mW
20-Pin Plastic DIP (derate 11.11mW/°C above +70°C).... 889mW
24-Pin Narrow Plastic DIP
(derate 13.33mW/°C above +70°C) 1.07W
24-Pin Plastic DIP (derate 9.09mW/°C above +70°C)...... 500mW
16-Pin Wide SO (derate 9.52mW/°C above +70°C)......... 762mW
20-Pin Wide SO (derate 10 00mW/°C above +70°C)....... 800mW
24-Pin Wide SO (derate 11.76mW/°C above +70°C)....... 941mW
28-Pin Wide SO (derate 12.50mW/°C above +70°C) 1W
44-Pin Plastic FP (derate 11.11mW/°C above +70°C) 889mW
14-Pin CERDIP (derate 9.09mW/°C above +70°C) 727mW
16-Pin CERDIP (derate 10.00mW/°C above +70°C) 800mW
20-Pin CERDIP (derate 11.11mW/°C above +70°C) 889mW
24-Pin Narrow CERDIP
(derate 12.50mW/°C above +70°C) 1W
24-Pin Sidebraze (derate 20.0mW/°C above +70°C).......... 1.6W
28-Pin SSOP (derate 9.52mW/°C above +70°C)............ 762mW
Operating Temperature Ranges
MAX2 _ _ C _ _ .... 0°C to +70°C
MAX2 _ _ E _ _ .... -40°C to +85°C
MAX2 _ _ M _ _ .... -55°C to +125°C
Storage Temperature Range .... -65°C to +160°C
Lead Temperature (soldering, 10sec) .... +300°C

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ELECTRICAL CHARACTERISTICS—MAX223/MAX230–MAX241

(MAX223/230/232/234/236/237/238/240/241, V_{CC} = +5V ±10; MAX233/MAX235, V_{CC} = 5V ±5%, C1–C4 = 1.0μF; MAX231/MAX239, V_{CC} = 5V ±10%; V+ = 7.5V to 13.2V; T_A = T_{MIN} to T_{MAX}; unless otherwise noted.)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
Output Voltage Swing	All transmitter outputs loaded with 3kΩ to ground		±5.0	±7.3		V
V_{CC} Power-Supply Current	No load, T_A = +25°C	MAX232/233		5	10	mA
		MAX223/230/234–238/240/241		7	15	
		MAX231/239		0.4	1	
V+ Power-Supply Current		MAX231		1.8	5	mA
		MAX239		5	15	
Shutdown Supply Current	T_A = +25°C	MAX223		15	50	μA
		MAX230/235/236/240/241		1	10	
Input Logic Threshold Low	T_{IN}; EN, $\overline{SHDN}$ (MAX233); $\overline{EN}$, SHDN (MAX230/235–241)				0.8	V
Input Logic Threshold High	T_{IN}		2.0			V
	EN, $\overline{SHDN}$ (MAX223); $\overline{EN}$, SHDN (MAX230/235/236/240/241)		2.4			
Logic Pull-Up Current	T_{IN} = 0V			1.5	200	μA
Receiver Input Voltage Operating Range			-30		30	V

+5V-Powered, Multichannel RS-232 Drivers/Receivers

ELECTRICAL CHARACTERISTICS—MAX223/MAX230–MAX241 (continued)

(MAX223/230/232/234/236/237/238/240/241, V_{CC} = +5V ±10; MAX233/MAX235, V_{CC} = 5V ±5%, C1–C4 = 1.0μF; MAX231/MAX239, V_{CC} = 5V ±10%; V+ = 7.5V to 13.2V; T_A = T_{MIN} to T_{MAX}; unless otherwise noted.)

PARAMETER	CONDITIONS			MIN	TYP	MAX	UNITS
RS-232 Input Threshold Low	T_A = +25°C, V_{CC} = 5V	Normal operation $\overline{SHDN}$ = 5V (MAX223) SHDN = 0V (MAX235/236/240/241)		0.8	1.2		V
		Shutdown (MAX223) $\overline{SHDN}$ = 0V, EN = 5V ($R4_{IN}$, $R5_{IN}$)		0.6	1.5		
RS-232 Input Threshold High	T_A = +25°C, V_{CC} = 5V	Normal operation $\overline{SHDN}$ = 5V (MAX223) SHDN = 0V (MAX235/236/240/241)			1.7	2.4	V
		Shutdown (MAX223) $\overline{SHDN}$ = 0V, EN = 5V ($R4_{IN}$, $R5_{IN}$)			1.5	2.4	
RS-232 Input Hysteresis	V_{CC} = 5V, no hysteresis in shutdown			0.2	0.5	1.0	V
RS-232 Input Resistance	T_A = +25°C, V_{CC} = 5V			3	5	7	kΩ
TTL/CMOS Output Voltage Low	I_{OUT} = 1.6mA (MAX231/232/233, I_{OUT} = 3.2mA)					0.4	V
TTL/CMOS Output Voltage High	I_{OUT} = -1mA			3.5	V_{CC} - 0.4		V
TTL/CMOS Output Leakage Current	0V ≤ R_{OUT} ≤ V_{CC}; EN = 0V (MAX223); $\overline{EN}$ = V_{CC} (MAX235–241)				0.05	±10	μA
Receiver Output Enable Time	Normal operation	MAX223			600		ns
		MAX235/236/239/240/241			400		
Receiver Output Disable Time	Normal operation	MAX223			900		ns
		MAX235/236/239/240/241			250		
Propagation Delay	RS-232 IN to TTL/CMOS OUT, C_L = 150pF	Normal operation			0.5	10	μs
		$\overline{SHDN}$ = 0V (MAX223)	t_{PHLS}		4	40	
			t_{PLHS}		6	40	
Transition Region Slew Rate	MAX223/MAX230/MAX234–241, T_A = +25°C, V_{CC} = 5V, R_L = 3kΩ to 7kΩ, C_L = 50pF to 2500pF, measured from +3V to -3V or -3V to +3V			3	5.1	30	V/μs
	MAX231/MAX232/MAX233, T_A = +25°C, V_{CC} = 5V, R_L = 3kΩ to 7kΩ, C_L = 50pF to 2500pF, measured from +3V to -3V or -3V to +3V				4	30	
Transmitter Output Resistance	V_{CC} = V+ = V- = 0V, V_{OUT} = ±2V			300			Ω
Transmitter Output Short-Circuit Current					±10		mA