

## รายงานผลการวิจัย

ระบบอุปกรณ์การตอบสนองระหว่างผู้เรียนและผู้สอน

โดย  
ผู้ช่วยศาสตราจารย์บุญเย็น ภูระหงษ์ หัวหน้าโครงการ  
นายกรวิรุช อิศวคุปตานนท์ ผู้ร่วมโครงการ  
นางสาวขวัญชนก ศรีमान ผู้ร่วมโครงการ

RCH

QA

76-9

.158

เลขหมู่..... 216245

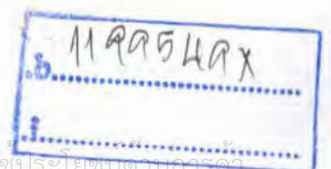
เลขทะเบียน..... 84558

วัน,เดือน,ปี..... 13 ต.ค. 2551

โครงการวิจัยที่ได้รับทุนสนับสนุนจากเงินรายได้คณะวิศวกรรมศาสตร์

ประจำปี 2550

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทคัดย่อ

โครงการนี้เป็นกรออกแบบระบบการรับและการตอบสนองระหว่างผู้เรียนและผู้สอน โดยแบ่งการทำงานออกเป็น 2 ส่วน คือ ส่วนของฮาร์ดแวร์ ซึ่งไมโครคอนโทรลเลอร์จะเป็นตัวรับข้อมูลจากผู้เรียน แล้วส่งไปยังเครื่องคอมพิวเตอร์ของผู้สอน โดยผ่านทางพอร์ตอนุกรม (RS-232) ในส่วนของซอฟต์แวร์ ใช้ภาษา C# ในการรับข้อมูลและจัดทำหน้าต่างแสดงผล (User Interface) และใช้ Microsoft SQL Server 2005 เป็นตัวจัดเก็บข้อมูล



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ

	หน้า
บทคัดย่อ	ก
สารบัญ	
สารบัญรูป	
บทที่ 1 บทนำ	1
1.1 แนวคิดเริ่มต้นในการทำโครงงาน	1
1.2 วัตถุประสงค์	2
1.3 ขอบเขตของโครงงาน	2
1.4 สถาปัตยกรรมของระบบ	3
1.5 ขั้นตอนการดำเนินโครงงาน	4
1.6 ผลที่คาดว่าจะได้รับ	4
บทที่ 2 ทฤษฎีและหลักการที่ใช้ในโครงงาน	6
2.1 การรับส่งข้อมูลแบบอนุกรม	6
2.1.1 ความรู้เบื้องต้นเกี่ยวกับมาตรฐาน RS-232	6
2.2 ไมโครคอนโทรลเลอร์ตระกูล AVR	7
2.2.1 ATmega128	7
2.2.1.1 คุณสมบัติของไมโครคอนโทรลเลอร์ตระกูลAVRเบอร์ ATmega128	7
2.2.1.2 รายละเอียดต่างๆของไอซี ATmega128	9
2.2.2 ATmega16	11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ(ต่อ)

หน้า

2.2.2.1 คุณสมบัติของไมโครคอนโทรลเลอร์ตระกูลAVRเบอร์ATmega16	11
2.2.2.2 รายละเอียดต่างๆของไอซี ATmega16	13
2.2.3 การสื่อสารผ่านพอร์ตอนุกรม UART	16
2.3 หลักการของระบบฐานข้อมูล	17
2.3.1 ความหมายของระบบฐานข้อมูล	17
2.3.2 ความสำคัญของระบบฐานข้อมูล	18
2.3.3 การบริหารฐานข้อมูล	18
2.3.4 ระบบการจัดการฐานข้อมูล (data base management system, DBMS)	20
บทที่ 3 การออกแบบโครงงาน	21
3.1 การออกแบบฐานข้อมูล	21
3.2 การออกแบบส่วนติดต่อกับผู้ใช้	22
3.3 การออกแบบวงจร	22
3.3.1 การออกแบบตัวแม่	22
3.3.2 การออกแบบตัวลูก	24
บทที่ 4 ผลการทดลอง	26
บทที่ 5 สรุปผลและวิจารณ์ผลการทดลอง	32
5.1 สรุปผล	32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ(ต่อ)

	หน้า
5.2 วิจารณ์ผลการทดลอง	32
5.3 แนวทางการพัฒนาในอนาคต	32

บรรณานุกรม

ภาคผนวก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป

	หน้า
ภาพที่ 1.1 สถาปัตยกรรมระบบ	3
ภาพที่ 2.1 การใช้งานพอร์ตอนุกรม RS-232	6
ภาพที่ 2.2 Pin Configurations ATmega128	8
ภาพที่ 2.3 Block Diagram ของ ไอซีเพอร์ ATmega128	10
ภาพที่ 2.4 Pin Configurations ATmega16	12
ภาพที่ 2.5 Block Diagram ของ ไอซีเพอร์ ATmega16	14
ภาพที่ 2.6 สถาปัตยกรรมแบบ RISC ของ ATmega16	15
ภาพที่ 3.1 วงจรตัวแม่(Master)	18
ภาพที่ 3.2 วงจรตัวลูก(Slave)1	19
ภาพที่ 3.3 วงจรตัวลูก(Slave)2	20
ภาพที่ 4.1 Database Diagrame	21
ภาพที่ 4.2 หน้าต่างเลือกวิชาเรียน	22
ภาพที่ 4.3 หน้าต่าง Machine Register	22
ภาพที่ 4.4 หน้าต่าง Question Setting	23
ภาพที่ 4.5 หน้าต่าง Start Exam	23
ภาพที่ 4.6 หน้าต่าง Result	24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป(ต่อ)

	หน้า
ภาพที่ 4.7 ตัวลูก(1)	24
ภาพที่ 4.8 ตัวลูก(2)	25
ภาพที่ 4.9 ตัวแม่	25
ภาพที่ 4.10 รวมอุปกรณ์	26



# บทที่ 1

## บทนำ

### 1.1 สรุปที่มาของปัญหาหรือแนวคิดเริ่มต้นในการทำโครงการ

กระบวนการเรียนการสอนทุกรายวิชา จำเป็นอย่างยิ่งต้องมีการวัดและประเมินผลผู้เรียนเพื่อวัดความก้าวหน้าทางการเรียน (แบบทดสอบหลังจบหน่วยเรียน/บทเรียน) และวัดผลสัมฤทธิ์ทางการเรียน (การสอบปลายภาค) ประโยชน์ของการวัดความก้าวหน้าทางการเรียนนั้น ก็เพื่อตัวอาจารย์ผู้สอนเอง มีโอกาสตรวจปรับการสอนของตนเองให้เข้ากับกลุ่มนักศึกษาและวุฒิภาวะของนักศึกษา อีกทั้งยังเป็นแรงกระตุ้นให้นักศึกษาทราบผลการเรียนรู้ของตนเองเพื่อจะนำไปสู่การตั้งใจเรียนในการเรียนต่อไป ส่วนการหาทางการเรียนนั้นจะเป็นการวัดความรู้ของนักศึกษาว่ามีความเข้าใจในเนื้อหาที่เรียนและมีการจดจำเนื้อหาได้มากน้อยเพียงใด แต่ด้วยปัจจัยบางประการทำให้การเรียนการสอนส่วนใหญ่ ผลสัมฤทธิ์ขาดการวัดความก้าวหน้าทางการเรียน และมุ่งเน้นวัดผลสัมฤทธิ์ทางการเรียนเท่านั้น และหากมีการวัดความก้าวหน้าทางการเรียน ก็ขาดกระบวนการตรวจสอบในห้องเรียน เนื่องจากอาจารย์นำข้อสอบไปตรวจนอกเวลาเรียน ทำให้นักศึกษาไม่สามารถทราบได้ว่า ตนเองทำข้อสอบได้มากน้อยเพียงใด และที่นำไปนั้นจะผิดหรือถูก ซึ่งปัญหาดังกล่าวหากแก้ไขได้ก็จะสามารถทำให้ตัวนักศึกษาเองมีความกระตือรือร้น และสามารถสอบถามอาจารย์ผู้สอนได้ทันต่อเนื้อหาที่ตนไม่เข้าใจ และทำให้เกิดเจตคติที่ดีต่อการเรียนการสอนในวิชา ส่วนอาจารย์ผู้สอนก็สามารถแยกนักศึกษาออกเป็นกลุ่มเก่ง กลุ่มปานกลางและกลุ่มอ่อนได้อย่างชัดเจนในการวัดและประเมินผลการเรียนแต่ละครั้งจะเสียเวลาไปกับ การออกข้อสอบ พิมพ์ ถ่ายสำเนาตามจำนวนผู้เรียน ซึ่งเป็นการสิ้นเปลืองทรัพยากรเป็นจำนวนมาก และหลังจากทดสอบแล้วก็ต้องมีการตรวจข้อสอบ จัดเก็บคะแนน สุดท้ายจะเป็นการแจ้งคะแนนให้ผู้เรียนทราบ ดังนั้นทางกลุ่มผู้จัดทำโครงการ จึงเสนอทำโครงการเรื่อง “ระบบการรับและการตอบสนองระหว่างผู้เรียนและผู้สอน” เพื่อลดระยะเวลาในส่วนนี้ทั้งหมด ระบบการรับและการตอบสนองระหว่างผู้เรียนและผู้สอนนี้จะทำให้ทั้งการเรียนการสอน การทดสอบ ตรวจสอบคำตอบ และแจ้งผลคะแนนสามารถทำได้ในชั่วโมงที่ทำการเรียนการสอนนั้นๆ และเมื่อทราบผลคะแนน ก็จะทำให้ผู้สอนสามารถวิเคราะห์ และประเมินผลว่าการเรียนการสอนครั้งนั้นๆ มีผู้เรียนเข้าใจในเนื้อหา มากน้อยเพียงใด และสามารถอธิบายซ้ำในเนื้อหาที่ผู้เรียนไม่เข้าใจในการเรียนการสอนครั้งนั้น ได้เลย

## 1.2 วัตถุประสงค์

- 1.2.1 เพื่อสร้างระบบเครื่องมือในการตอบสนองระหว่างผู้เรียนและผู้สอน
- 1.2.2 เพื่อใช้ในการเก็บข้อมูล เวลาเรียน คะแนนที่ทำได้ของผู้เรียนเป็นรายบุคคล ในการเรียนการสอนแต่ละครั้ง
- 1.2.3 เพื่อช่วยผู้เรียนในเรื่องการปรับตัวหลังจากการประเมินตนเอง และช่วยผู้สอนในการตรวจปรับแนวทางเพื่อให้เหมาะสมต่อการสอน

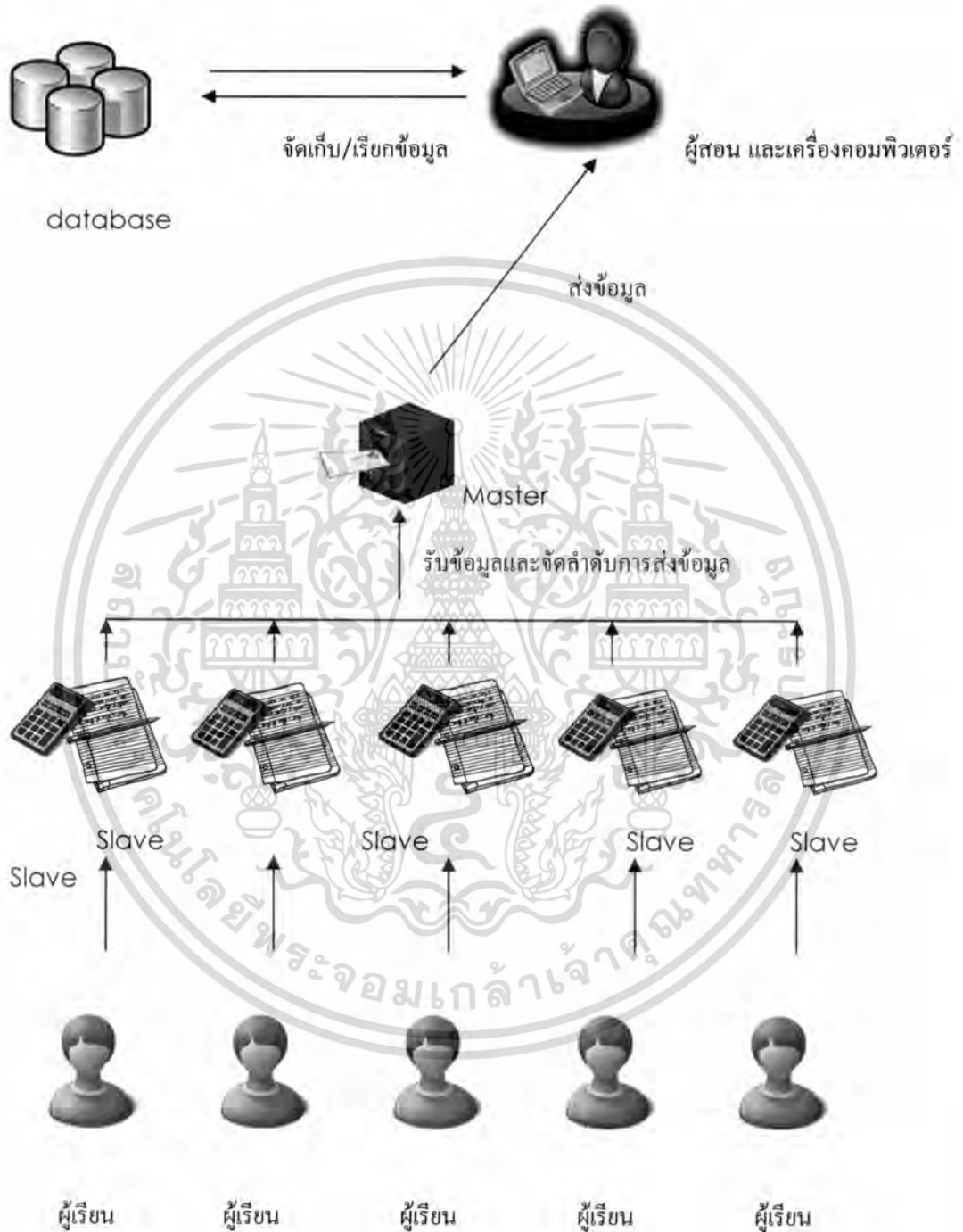
## 1.3 ขอบเขตของโครงการ

โครงการนี้เป็นโครงการที่ทำทั้ง ซอฟต์แวร์และฮาร์ดแวร์ โดยฮาร์ดแวร์จะแบ่งเป็นสองส่วน ส่วนแรกคือ ตัวแม่(Master) จะทำการสื่อสารระหว่างเครื่องคอมพิวเตอร์ของผู้สอนและเครื่องลูกทุกตัว โดยจะทำการจัดลำดับข้อมูลจากเครื่องลูกแล้วส่งไปยังเครื่องคอมพิวเตอร์ ส่วนที่สอง คือ ตัวลูก(Slave) จะเป็นตัวที่แจกให้กับผู้เรียนทุกคนได้ใช้ในการตอบสนองในระหว่างที่มีการเรียนการสอน ซอฟต์แวร์ จะเป็นหน้าต่างสำหรับผู้สอนโดยเฉพาะ จึงต้องมีการตรวจสอบสิทธิ์ของผู้ใช้ ผู้สอนจะได้รับชื่อผู้ใช้(username) และรหัสผ่าน(password) ในหน้าต่างผู้สอนจะสามารถตั้งคำถาม คำตอบ ตรวจสอบข้อสอบ และตรวจสอบคะแนนของผู้เรียนได้

โครงการนี้จะสามารถใช้ได้กับทุกรายวิชา โดยจะเก็บข้อมูลตามใบรายชื่อของทางสถาบันระบบการรับและการตอบสนองระหว่างผู้เรียนและผู้สอนนี้ จะถูกนำไปทดลองใช้ในวิชา

MICROPROCESSORS AND MICROCOMPUTERS

## 1.4 สถาปัตยกรรมของระบบ



ภาพที่ 1.1 สถาปัตยกรรมระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.5 ขั้นตอนการดำเนินโครงการ

### 1.5.1 Design Project Structure

ออกแบบโครงสร้างและระบบพื้นฐานของโครงการ เช่น เลือกไมโครคอนโทรลเลอร์ที่เหมาะสมกับงานของโครงการ ลักษณะความต้องการของฐานข้อมูล ความเหมาะสมของภาษาที่จะนำมาใช้พัฒนาโครงการ เป็นต้น

### 1.5.2 Focus & Enhance Knowledge

เลือกศึกษาและหาความรู้เพิ่มเติมจากข้อมูลในหัวข้อที่เกี่ยวข้อง และเลือกประยุกต์ความรู้เหล่านั้นให้เหมาะสมกับระบบ

### 1.5.3 Design Project & Database System

ออกแบบระบบเพื่อรองรับความต้องการของโครงการ เช่น วงจรของตัวแม่และตัวลูก ระบบฐานข้อมูล เป็นต้น

### 1.5.4 Design Interface

ออกแบบโครงสร้าง และอินเตอร์เฟซ(interface) โดยต้องออกแบบให้เกิดความสะดวกแก่ผู้ใช้งานที่สุด

### 1.5.5 Work On Entire Article

ดำเนินการตามหัวข้อที่วางไว้

### 1.5.6 System Testing

ทดสอบระบบและทำการปรับปรุงแก้ไข

## 1.6 ผลที่คาดว่าจะได้รับ

1.6.1 ได้ชุดระบบการรับและการตอบสนองระหว่างผู้เรียนและผู้สอน

1.6.2 ช่วยทำให้มีการบันทึกข้อมูลของการเรียนการสอนของผู้เรียนในรายวิชานั้นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 1.6.3 ผู้เรียนทราบผลการเรียนของตนเองเป็นปัจจุบันอันจะส่งผลให้ผู้เรียนเกิดความตั้งใจเรียนมากขึ้น และผู้สอน มีการตรวจปรับแนวทางการสอนอย่างเป็นปัจจุบัน เพื่อตรงวัตถุประสงค์การสอนและตรงตามวุฒิภาวะของผู้เรียน
- 1.6.4 ทำให้เกิดความเข้าใจและพัฒนาความรู้ในเรื่องของภาษา C, C# ส่วนของการใช้งานโปรแกรม Microsoft Visual Studio 2008, Adobe Photoshop CS2 และการใช้งานระบบฐานข้อมูลของ Microsoft SQL Server 2005



## บทที่ 2

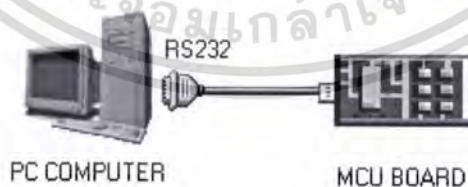
### ทฤษฎีและหลักการที่ใช้ในโครงการ

#### 2.1 การรับส่งข้อมูลแบบอนุกรม

การรับส่งข้อมูลแบบอนุกรม ซึ่งมีข้อดีคือ ใช้สายสัญญาณน้อยและส่งได้เป็นระยะทางไกล สำหรับโครงการนี้ใช้การติดต่อกับไมโครคอนโทรลเลอร์ รับส่งข้อมูลแบบอนุกรมกับคอมพิวเตอร์ เพื่อทำการเปลี่ยนระดับสัญญาณไฟฟ้าได้มาตรฐานในการรับส่งข้อมูลผ่านทางพอร์ตอนุกรม (Serial Port) ของเครื่องไมโครคอมพิวเตอร์ ซึ่งสัญญาณข่าวสารที่ได้จะถูกประมวลผลโดยโปรแกรม Visual Basic เพื่อแสดงผลออกทางหน้าจอคอมพิวเตอร์ ([www.JOBPUB.com](http://www.JOBPUB.com))

##### 2.1.1 ความรู้เบื้องต้นเกี่ยวกับมาตรฐาน RS-232

โดยปกติเครื่องไมโครคอมพิวเตอร์จะมีพอร์ตที่เป็นอนุกรมชื่อว่า RS-232 อยู่ในตัวมันเองอยู่แล้ว ซึ่งพอร์ต RS-232 นี้ทำหน้าที่รับและส่งข้อมูลในแบบอนุกรมเรียกว่า Universal Asynchronous Adapter เหตุที่มีชื่อเรียกว่า RS-232 ก็เนื่องมาจาก สมาคมผู้ผลิตอุปกรณ์อิเล็กทรอนิกส์ ของอเมริกา หรือ EIA (RS-232 : Recommended Standard Number 232, EIA: Electronic Industry Association) ได้กำหนดมาตรฐานของอุปกรณ์สื่อสารแบบอนุกรมเอาไว้ภายใต้ชื่อว่า RS-232 (ความจริงแล้ว มาตรฐานของรับส่งข้อมูลแบบอนุกรมมีหลายมาตรฐาน แต่ที่นิยมกันมากที่สุดสำหรับเครื่องไมโครคอมพิวเตอร์



ภาพที่ 2.1 การใช้งานพอร์ตอนุกรม RS-232

## 2.2 ไมโครคอนโทรลเลอร์ตระกูล AVR

ไมโครคอนโทรลเลอร์ AVR เป็นหนึ่งในไมโครคอนโทรลเลอร์ ที่ผลิตโดยบริษัท ATMEL AVR จัดเป็นไมโครคอนโทรลเลอร์ตระกูลใหม่จาก ATMEL มีสถาปัตยกรรมแบบ RISC (Advanced RISC architecture) คือหนึ่งคำสั่งทำงานใช้สัญญาณนาฬิกาเพียงหนึ่งลูก เป็นไมโครคอนโทรลเลอร์ที่มีประสิทธิภาพและความสามารถสูง แบ่งออกเป็นหลายอนุกรม ในแต่ละอนุกรมยังแบ่งออกเป็นหลายเบอร์ เพื่อรองรับความต้องการที่แตกต่างของผู้ใช้งาน

### 2.2.1 ไมโครคอนโทรลเลอร์ตระกูล AVR เบอร์ ATmega128

#### 2.2.1.1 คุณสมบัติของไมโครคอนโทรลเลอร์ตระกูล AVR เบอร์ ATmega128

- สมรรถภาพสูง, กินไฟน้อย, เป็นไมโครคอนโทรลเลอร์ชนิด 8 บิต
- สถาปัตยกรรมภายในเป็นแบบ RISC ขั้นสูง ( RISC คือ ทำให้การประมวลผลมีความเร็ว 1 คำสั่ง / 1 clock หรือ CPU สามารถประมวลคำสั่งได้ 1 MIPS / MHz)
- มีคำสั่งในการควบคุมการทำงานไมโครคอนโทรลเลอร์จำนวน 133 คำสั่ง
- มีกลุ่มรีจิสเตอร์ทำงานทั่วไป ขนาด 8 บิต จำนวน 32 ตัว และรีจิสเตอร์ควบคุม
- CPU สามารถประมวลคำสั่งได้สูงถึง 16 MIPS ส่งผ่านที่ 16 MHz
- มีหน่วยความจำชนิดแฟลชขนาด 128 กิโลไบต์ที่สามารถเขียนข้อมูลซ้ำได้ 10,000 ครั้ง
- มีหน่วยความจำชนิดEEPROM ขนาด 4 กิโลไบต์ สามารถเขียนข้อมูลซ้ำได้ 100,000 ครั้ง
- มีหน่วยความจำชนิด SRAM ขนาด 4 กิโลไบต์
- มีหน่วยความจำภายนอกสูงถึง 64 กิโลไบต์
- โมดูลสร้างสัญญาณ PWM (Pulse Width Modulator) มีจำนวน 4 ช่อง
- มีโมดูลแปลงสัญญาณอะนาลอกเป็นดิจิตอล(ADC) ขนาด 10 บิต มากถึง 8 ช่อง
- การสื่อสารข้อมูลมีทั้งแบบ UART ( Universal Asynchronous Receiver Transmitters ) หรือแบบ RS-232,SPI ( Serial Peripheral Interface ) และแบบ I<sup>2</sup>C เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-แรงดันไฟเลี้ยง

-2.7 ถึง 5.5 โวลต์ ในรุ่น ATmega 128L

-4.5 ถึง 5.5 โวลต์ ในรุ่น ATmega 128

-ความถี่ที่ใช้ในการทำงาน

-0 ถึง 8 MHz ในรุ่น ATmega 128L

-0 ถึง 16 MHz ในรุ่น ATmega 128



ภาพที่ 2.2 Pin Configurations ATmega128

### 2.2.1.2 รายละเอียดขาต่างๆของไอซี ATmega128

VCC

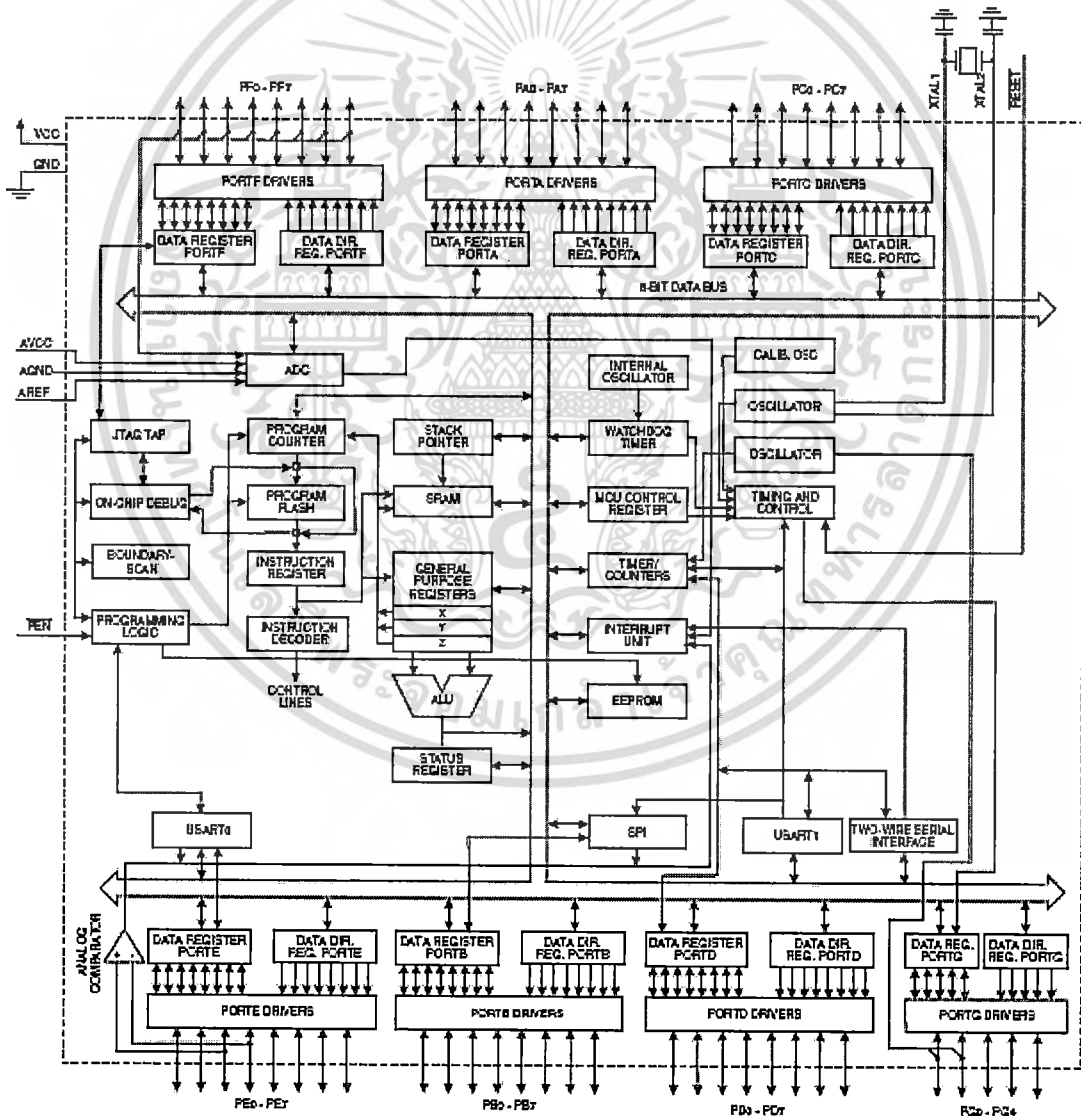
ขาย้ายไฟให้กับไอซี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GND	ขากราวด์
Port A (PA7..PA0)	เป็นพอร์ตสองทิศทางขนาด 8 บิต สามารถกำหนดให้แต่ละขาของพอร์ตสามารถพูลอัป (pull-up) ภายในแยกจากกัน และสามารถกำหนดงานใช้เป็นพอร์ตอินพุตสัญญาณอะนาล็อก (A/D converter) ได้
Port B (PB7..PB0)	เป็นพอร์ตสองทิศทางขนาด 8 บิต สามารถกำหนดให้แต่ละขาของพอร์ตสามารถพูลอัป(pull-up)ภายในแยกจากกัน และเป็นขาพอร์ตหน้าที่พิเศษอีกด้วย เช่น ขาสำหรับการโปรแกรมชิพ ขาอินพุตสัญญาณนาฬิกาภายนอก เป็นต้น
Port C (PC7..PC0)	เป็นพอร์ตสองทิศทางขนาด 8 บิต สามารถกำหนดให้แต่ละขาของพอร์ตสามารถพูลอัป (pull-up)ภายในแยกจากกัน และเป็นขาพอร์ตหน้าที่พิเศษ เช่น ขาเชื่อมต่อกับคีย์บอร์ดและโปรแกรมด้วยการเชื่อมต่อแบบ JTAG เป็นต้น
Port D (PD7..PD0)	เป็นพอร์ตสองทิศทางขนาด 8 บิต สามารถกำหนดให้แต่ละขาของพอร์ตสามารถพูลอัป(pull-up)ภายในแยกจากกัน และเป็นขาพอร์ตหน้าที่พิเศษ เช่น ขาเชื่อมต่อพอร์ตอนุกรม ขาอินเตอร์รัปต์เนื่องจากสัญญาณภายนอก เป็นต้น
Port E (PE7..PE0)	เป็นพอร์ตสองทิศทางขนาด 8 บิต สามารถกำหนดให้แต่ละขาของพอร์ตสามารถพูลอัป(pull-up)ภายในแยกจากกัน
Port F (PF7..PF0)	เป็นพอร์ตที่มีขาเป็นอินพุตรับค่าเป็นอะนาล็อกแล้วเปลี่ยนเป็นดิจิตอล
Port G (PG7..PG0)	เป็นพอร์ตสองทิศทางขนาด 5 บิต สามารถกำหนดให้แต่ละขาของพอร์ตสามารถพูลอัป(pull-up)ภายในแยกจากกัน
RESET	ขารีเซ็ตจะเกิดการรีเซ็ตก็ต่อเมื่อมีลอจิก0 เข้ามาเป็นเวลาไม่ต่ำกว่า 50 ns แม้ว่า จะไม่มีสัญญาณนาฬิกา ก็ตาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- XTAL1 เป็นขาอินพุตสำหรับสัญญาณนาฬิกาภายนอก
- XTAL2 เป็นขาเอาต์พุตของสัญญาณนาฬิกา
- AVCC เป็นขาที่ป้องกันความต่างศักย์ให้กับ Port F และเปลี่ยนอะนาลอกเป็นดิจิตอลและต้องต่อกับขา VCC เสมอ แม้จะไม่ใช่ ADC ก็ตาม
- AREF เป็นขาที่ใช้อ้างอิงสำหรับ โมดูลแปลงสัญญาณอะนาลอกเป็นดิจิตอล
- PEN เป็นขาที่ติดต่อโปรแกรมจากพอร์ตอนุกรม



ภาพที่ 2.3 Block Diagram ของไอซีเบอร์ ATmega128

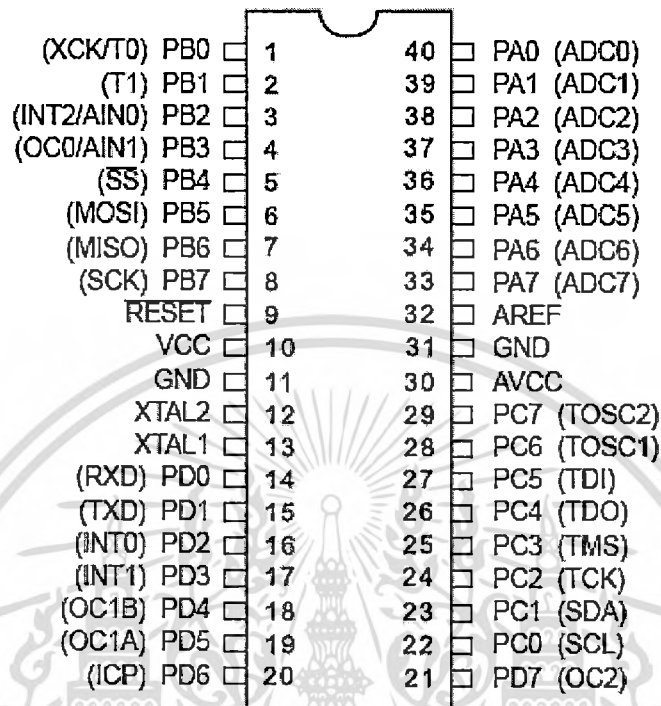
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.2.2 ไมโครคอนโทรลเลอร์ตระกูล AVR เบอร์ ATmega16

### 2.2.2.1 คุณสมบัติของไมโครคอนโทรลเลอร์ตระกูล AVR เบอร์ ATmega16

- สมรรถภาพสูง, กินไฟน้อย, เป็นไมโครคอนโทรลเลอร์ชนิด 8 บิต
- สถาปัตยกรรมภายในเป็นแบบ RISC ขั้นสูง ( RISC คือ ทำให้การประมวลผลมีความเร็ว 1 คำสั่ง / 1 clock หรือ CPU สามารถประมวลคำสั่งได้ 1 MIPS / MHz)
- มีคำสั่งในการควบคุมการทำงานไมโครคอนโทรลเลอร์จำนวน 131 คำสั่ง
- มีกลุ่มรีจิสเตอร์ทำงานทั่วไป ขนาด 8 บิต จำนวน 32 ตัว
- CPU สามารถประมวลคำสั่งได้สูงถึง 16 MIPS ที่ความถี่ 16 MHz
- มีหน่วยความจำชนิดแฟลชขนาด 16 กิโลไบต์ที่สามารถเขียนข้อมูลซ้ำได้ 10,000 ครั้ง
- มีหน่วยความจำชนิดEEPROM ขนาด 512 ไบต์ สามารถเขียนข้อมูลซ้ำได้ 100,000 ครั้ง
- มีหน่วยความจำชนิด SRAM ขนาด 1 กิโลไบต์
- มีตัวนับ/จับเวลา ขนาด 8 บิตและ 16 บิต พร้อมตัวหาร(Prescaler)
- โมดูลสร้างสัญญาณ PWM (Pulse Width Modulator) มีจำนวน 4 ช่อง
- มีโมดูลแปลงสัญญาณอะนาลอกเป็นดิจิตอล(ADC) ขนาด 10 บิต มากถึง 8 ช่อง
- การสื่อสารข้อมูลมีทั้งแบบ UART ( Universal Asynchronous Receiver Transmitters ) หรือแบบ RS-232,SPI ( Serial Peripheral Interface ) และแบบ I<sup>2</sup>C เป็นต้น
- แรงดันไฟเลี้ยง
  - 2.7 ถึง 5.5 โวลต์ ในรุ่น ATmega 16L
  - 4.5 ถึง 5.5 โวลต์ ในรุ่น ATmega 16
- ความถี่ที่ใช้ในการทำงาน
  - 0 ถึง 8 MHz ในรุ่น ATmega 16L
  - 0 ถึง 16 MHz ในรุ่น ATmega 16

### PDIP

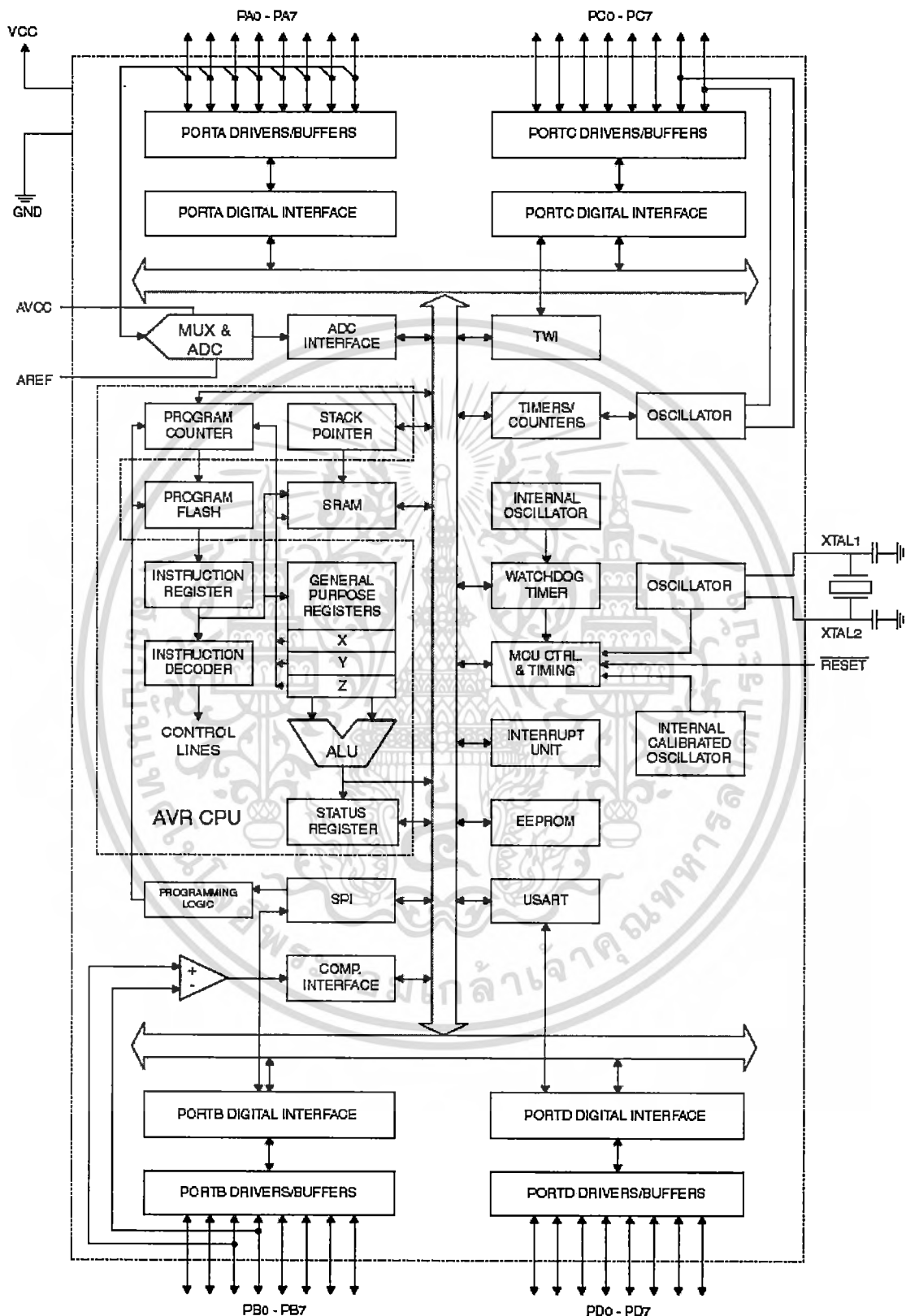


ภาพที่ 2.4 Pin Configurations ATmega16

#### 2.2.2.2 รายละเอียดขาต่างๆของไอซี ATmega16

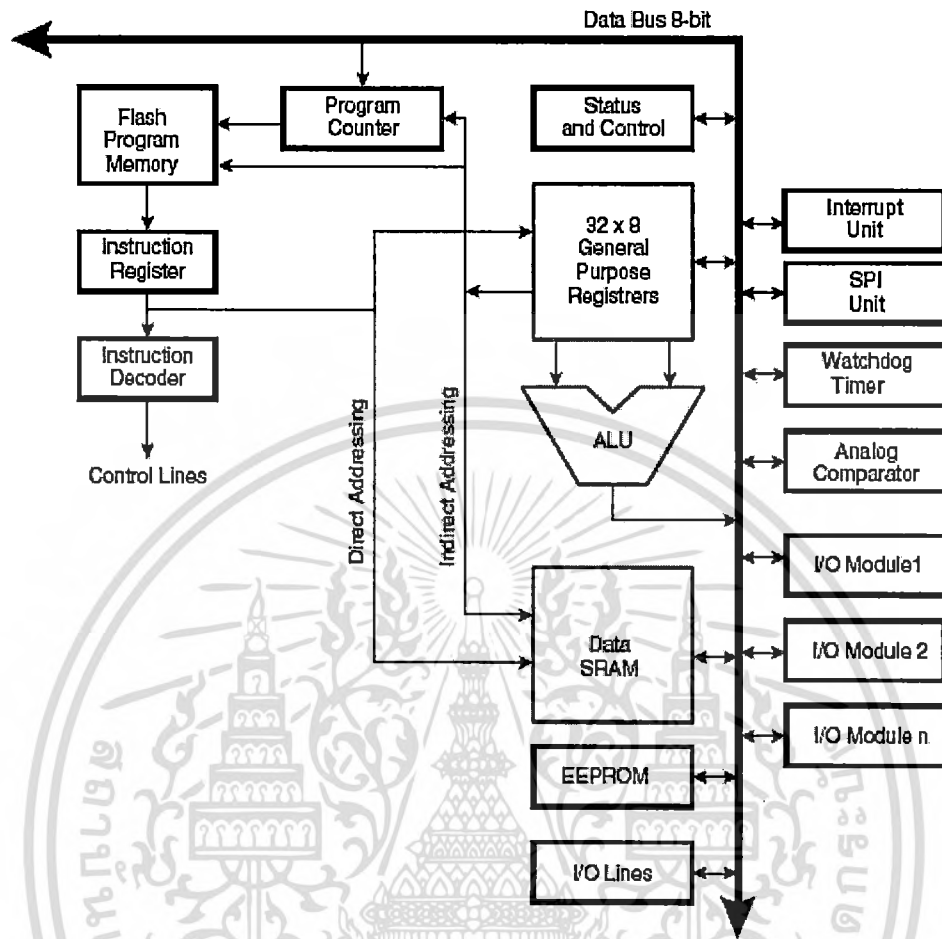
- VCC ขาแรงดันไฟตรง
- GND ขากราวด์
- Port A (PA7..PA0) เป็นพอร์ตสองทิศทางขนาด 8 บิต สามารถกำหนดให้แต่ขาของพอร์ตสามารถพูลอัป (pull-up) ภายในแยกจากกัน และสามารถกำหนดงานใช้เป็นพอร์ตอินพุตสัญญาณอะนาลอก (A/D converter) ได้
- Port B (PB7..PB0) เป็นพอร์ตสองทิศทางขนาด 8 บิต สามารถกำหนดให้แต่ขาของพอร์ตสามารถพูลอัป(pull-up)ภายในแยกจากกัน และเป็นขาพอร์ตหน้าที่พิเศษอีกด้วย เช่น ขาสำหรับการโปรแกรมชิพ ขาป้อนสัญญาณนาฬิกาภายนอก เป็นต้น

Port C (PC7..PC0)	เป็นพอร์ตสองทิศทางขนาด 8 บิต สามารถกำหนดให้แต่ละขาของพอร์ตสามารถพูลอัป (pull-up) ภายในแยกจากกัน และเป็นขาพอร์ตหน้าที่พิเศษ เช่น ขาเชื่อมต่อกับคีย์บอร์ดและโปรแกรมด้วยการเชื่อมต่อแบบ JTAG เป็นต้น
Port D (PD7..PD0)	เป็นพอร์ตสองทิศทางขนาด 8 บิต สามารถกำหนดให้แต่ละขาของพอร์ตสามารถพูลอัป (pull-up) ภายในแยกจากกัน และเป็นขาพอร์ตหน้าที่พิเศษ เช่น ขาเชื่อมต่อพอร์ตอนุกรม ขาอินเทอร์รัปต์เนื่องจากสัญญาณภายนอก เป็นต้น
RESET	ขารีเซ็ตจะเกิดการรีเซ็ตก็ต่อเมื่อมีลอจิก 0 เข้ามาเป็นเวลาไม่ต่ำกว่า 50 ns แม้ว่าจะไม่มีสัญญาณนาฬิกาก็ตาม
XTAL1	เป็นขาอินพุตสำหรับสัญญาณนาฬิกาภายนอก (ต่อคริสตัลออกสซิลเลเตอร์)
XTAL2	เป็นขาเอาต์พุตของสัญญาณนาฬิกา (ต่อคริสตัลออกสซิลเลเตอร์)
AVCC	ขาแรงดันสำหรับพอร์ต A และ โมดูลแปลงสัญญาณอะนาลอกเป็นดิจิตอล
AREF	เป็นขาที่ใช้อ้างอิงสำหรับ โมดูลแปลงสัญญาณอะนาลอกเป็นดิจิตอล



ภาพที่ 2.5 Block Diagram ของไมโครคอนโทรลเลอร์ ATmega16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.6 สถาปัตยกรรมแบบ RISC ของ ATmega16

### 2.2.3 การสื่อสารผ่านพอร์ตอนุกรม UART

การใช้งาน UART จะมีเฉพาะภายใน CPU model 1 มีลักษณะเป็นแบบ full-duplex ไมโครคอนโทรลเลอร์ ATmega128/16 ให้มีฟังก์ชัน โดยมีคุณสมบัติดังนี้

- สามารถเปลี่ยนแปลงบอรรถะของการสื่อสารได้หลายบอรรถะ
- สามารถสื่อสารได้ในอัตราบอรรถะที่สูงในขณะที่ความ XTAL ถี่ต่ำ
- สื่อสารข้อมูลได้ทั้ง 8 บิต และ 9 บิต
- มีส่วนของการกำจัดสัญญาณรบกวน
- ตรวจสอบความผิดพลาดของการสื่อสารข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ตรวจจับความผิดพลาดของบิตเริ่มต้น
- จัดให้มีการแยกอินเทอร์รัพท์ของการสื่อสาร
- มีบัฟเฟอร์ในการเก็บข้อมูล

การส่งข้อมูลจะถูกเริ่มต้นโดยการเขียนข้อมูลที่ต้องการส่งไปยังรีจิสเตอร์ UDR (UART I/O Data register) ข้อมูลจาก UDR จะถูกส่งเข้าไปยังชิพรีจิสเตอร์ เพื่อทำการส่งข้อมูลออกไปแบบอนุกรม ซึ่งชิพรีจิสเตอร์นี้จะอยู่ในสถานะว่างก็ต่อเมื่อส่งบิตสุดท้ายของข้อมูลออกไปนั่นก็คือ stop bit ซึ่งจะสามารถโหลดข้อมูลใหม่เข้ามาได้ทันที

หลังจากที่ข้อมูลถูกส่งจาก UDR ผ่านชิพรีจิสเตอร์ออกไป บิต UDRE(UART Data Register Empty) ที่อยู่ภายในรีจิสเตอร์สถานะ USR(UART Status Register) จะถูกเซต นั่นหมายความว่า UART พร้อมจะรับข้อมูลตัวถัดไป

## 2.3 ฐานข้อมูลและระบบจัดการฐานข้อมูล

ในปัจจุบันการจัดโครงสร้างข้อมูลให้เป็นแบบฐานข้อมูลกำลังเป็นที่นิยม เกือบทุกหน่วยงานที่มีการใช้ระบบสารสนเทศจะจัดทำข้อมูลให้เป็นแบบฐานข้อมูล เนื่องจากปริมาณข้อมูลมีมากถ้าจัดข้อมูลเป็นแบบแฟ้มข้อมูลจะทำให้มีแฟ้มข้อมูลเป็นจำนวนมาก ซึ่งจะทำให้เกิดข้อมูลที่ซ้ำซ้อนกันได้ ข้อมูลที่ซ้ำซ้อนนี้จะก่อให้เกิดปัญหามากมาย

### 2.3.1 ความหมายของระบบฐานข้อมูล

ฐานข้อมูล (database) หมายถึง กลุ่มของข้อมูลที่ถูกเก็บรวบรวมไว้ โดยมีความสัมพันธ์ซึ่งกันและกัน โดยไม่ได้บังคับว่าข้อมูลทั้งหมดนี้จะต้องเก็บไว้ในแฟ้มข้อมูลเดียวกันหรือแยกเก็บหลาย ๆ แฟ้มข้อมูล นั่นก็คือการเก็บข้อมูลในฐานข้อมูลนั้นเราอาจจะเก็บทั้งฐานข้อมูล โดยใช้แฟ้มข้อมูลเพียงแฟ้มข้อมูลเดียวกัน หรือจะเก็บไว้ในหลาย ๆ แฟ้มข้อมูล ที่สำคัญคือจะต้องสร้างความสัมพันธ์และเรียกใช้ความสัมพันธ์นั้นได้ มีการกำจัดความซ้ำซ้อนของข้อมูลออกและเก็บแฟ้มข้อมูลเหล่านี้ไว้ที่ศูนย์กลาง เพื่อที่จะนำข้อมูลเหล่านี้มาใช้ร่วมกัน ควบคุมดูแลรักษาเมื่อผู้

ต้องการใช้งานและผู้มีสิทธิ์จะใช้ข้อมูลนั้นสามารถดึงข้อมูลที่ต้องการออกไปใช้ได้ ข้อมูลบางส่วนอาจใช้ร่วมกับผู้อื่นได้ แต่บางส่วนของข้อมูลเหล่านั้นจึงจะสามารถใช้ได้ โดยทั่วไปองค์กรต่าง ๆ จะสร้างฐานข้อมูลไว้เพื่อเก็บข้อมูลต่าง ๆ ของตัวองค์กร โดยเฉพาะอย่างยิ่งข้อมูลในเชิงธุรกิจ

### 2.3.2 ความสำคัญของระบบฐานข้อมูล

- ลดการเก็บข้อมูลที่ซ้ำซ้อน
- รักษาความถูกต้องของข้อมูล
- การป้องกันและรักษาความปลอดภัยให้กับข้อมูลทำได้อย่างสะดวก
- สามารถใช้ข้อมูลร่วมกันได้
- มีความเป็นอิสระของข้อมูล
- สามารถขยายงานได้ง่าย
- ทำให้ข้อมูลบูรณะกลับสู่สภาพปกติได้เร็วและมีมาตรฐาน

### 2.3.3 การบริหารฐานข้อมูล

ในระบบฐานข้อมูลนอกจากจะมีระบบการจัดการฐานข้อมูล ซึ่งเป็นซอฟต์แวร์ที่สร้างขึ้นเพื่อจัดการกับข้อมูลให้เป็นระบบ จะได้นำไปเก็บรักษา เรียกใช้ หรือนำมาปรับปรุงให้ทันสมัยได้ง่ายแล้ว ในระบบฐานข้อมูลยังต้องประกอบด้วยบุคคลที่มีหน้าที่ควบคุมดูแลระบบฐานข้อมูล คือ ผู้บริหารฐานข้อมูล

เหตุผลประการหนึ่งของการจัดทำระบบจัดการฐานข้อมูล คือ การมีศูนย์กลางควบคุมทั้งข้อมูลและ โปรแกรมที่เข้าถึงข้อมูลเหล่านั้น บุคคลที่มีอำนาจหน้าที่ดูแลการควบคุมนี้ เรียกว่า ผู้บริหารฐานข้อมูล หรือ DBA (data base administrator) คือ ผู้มีหน้าที่ควบคุมการบริหารงานของฐานข้อมูลทั้งหมด

### 2.3.4 ระบบการจัดการฐานข้อมูล (data base management system, DBMS)

หน้าที่ของระบบการจัดการฐานข้อมูล

2.3.4.1 ระบบจัดการฐานข้อมูลเป็นซอฟต์แวร์ที่ทำหน้าที่ดังต่อไปนี้ **ดูแลการใช้งาน**  
ให้กับผู้ใช้

ในการติดต่อกับตัวจัดการระบบแฟ้มข้อมูลได้ ในระบบฐานข้อมูลนี้ข้อมูลจะมีขนาดใหญ่  
ซึ่งจะถูกจัดเก็บไว้ในหน่วยความจำสำรอง เมื่อผู้ใช้ต้องการจะใช้ฐานข้อมูล ระบบการจัดการ  
ฐานข้อมูลจะทำหน้าที่ติดต่อกับระบบแฟ้มข้อมูลซึ่งเสมือนเป็นผู้จัดการแฟ้มข้อมูล (file manager)  
นำข้อมูลจากหน่วยความจำสำรองเข้าสู่หน่วยความจำหลักเฉพาะส่วนที่ต้องการใช้งาน และทำ  
หน้าที่ประสานกับตัวจัดการระบบแฟ้มข้อมูลในการจัดเก็บ เรียกใช้ และแก้ไขข้อมูล

2.3.4.2 ควบคุมระบบความปลอดภัยของข้อมูล โดยป้องกันไม่ให้ผู้ที่ไม่ได้รับอนุญาตเข้า  
มาเรียกใช้หรือแก้ไขข้อมูลในส่วนป้องกันเอาไว้ พร้อมทั้งสร้างฟังก์ชันในการจัดทำข้อมูลสำรอง  
เมื่อเกิดความขัดข้องของระบบแฟ้มข้อมูล หรือเครื่องคอมพิวเตอร์เกิดการเสียหายนั้นฟังก์ชันนี้  
สามารถทำการฟื้นฟูสภาพของระบบข้อมูลกลับเข้าสู่สภาพที่ถูกต้องสมบูรณ์ได้

2.3.4.3 ควบคุมการใช้ข้อมูลในสภาพที่มีผู้ใช้พร้อม ๆ กันหลายคน โดยจัดการเมื่อมี  
ข้อผิดพลาดของข้อมูลเกิดขึ้น

## บทที่ 3

### การออกแบบโครงงาน

#### 3.1 การออกแบบฐานข้อมูล

เนื่องจากฐานข้อมูลที่จะสร้างต้องสามารถรองรับผู้เรียนและผู้สอนในทุกรายวิชา ฐานข้อมูลจึงมีขนาดใหญ่ ดังนั้นส่วนของฐานข้อมูลจึงใช้โปรแกรม Microsoft SQL Server 2005 โดยมี 10 ตาราง ดังนี้

ตาราง Student

Student_ID	Student_Name	Student_Section
------------	--------------	-----------------

ตาราง ClassQuestion

ClassQuestionId	ClassRoomId	QuestionId	QuestionNumber
-----------------	-------------	------------	----------------

ตาราง Question

QuestionId	QuestionTitle
------------	---------------

ตาราง Answer

QuestionId	ChoiceId	AnswerTitle	Correct
------------	----------	-------------	---------

ตาราง Classroom

ClassRoomId	ClassRoomDate	SubjectId
-------------	---------------	-----------

ตาราง StudentAnswer

ClassRoomId	MachineId	ClassQuestionId	ChoiceId
-------------	-----------	-----------------	----------

ตาราง Choice

ChoiceId
----------

ตาราง StudentClass

ClassRoomId	MachineId	StudentId
-------------	-----------	-----------

ตาราง Machine

MachineID
-----------

ตาราง Subject

SubjectId	SubjectName
-----------	-------------

### 3.2 การออกแบบส่วนติดต่อกับผู้ใช้ (User Interface)

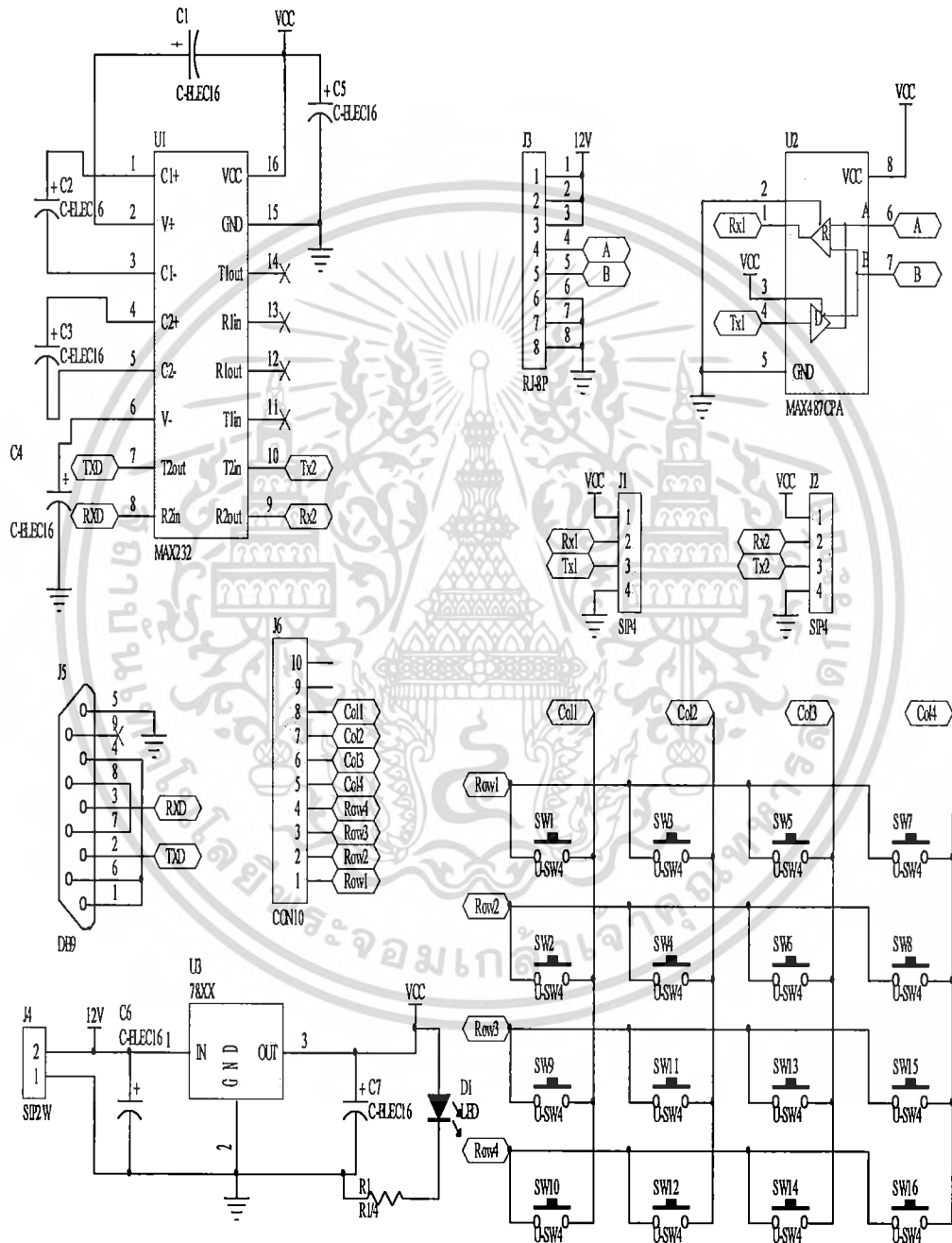
การออกแบบซอฟต์แวร์ในส่วนของหน้าต่างติดต่อผู้ใช้ จะใช้โปรแกรม Microsoft Visual Studio 2005 with C# โดยได้ออกแบบให้ง่ายต่อการใช้ แม้จะเป็นการเข้ามาใช้งานครั้งแรกผู้ใช้ก็สามารถเข้าใจและใช้งานได้เลย โปรแกรมจะทำหน้าที่แสดงข้อมูล และจัดเก็บข้อมูลที่ได้มาจากหน้าต่างติดต่อผู้ใช้และการประมวลผลในส่วนของไมโครคอนโทรลเลอร์ ที่รับผลผ่านทางพอร์ต RS-232 ของเครื่องคอมพิวเตอร์ ในรูปของฐานข้อมูล

### 3.3 การออกแบบวงจร

#### 3.3.1 การออกแบบตัวแม่ (Master)

ใช้ไมโครคอนโทรลเลอร์ตระกูล AVR เบอร์ ATmega128 ในส่วนของการประมวลผล ใช้ไอซีเร็กกูเลต เบอร์ 7805 เพื่อแปลงแรงดันให้เหลือแรงดันคงที่ 5 โวลต์ และป้อนให้ส่วนต่างๆภายในวงจร ส่วนของการติดต่อผ่านทางพอร์ต RS-232 ข้อมูลจะถูกส่งออกผ่านทาง

พอร์ต TXD ของไมโครคอนโทรลเลอร์ ส่วนการรับคำสั่งจะถูกนำเข้ามาโดยผ่านพอร์ต RXD จะมีสวิตช์ใช้คีย์ข้อมูลต่างๆเพื่อติดต่อไปยังเครื่องลูก

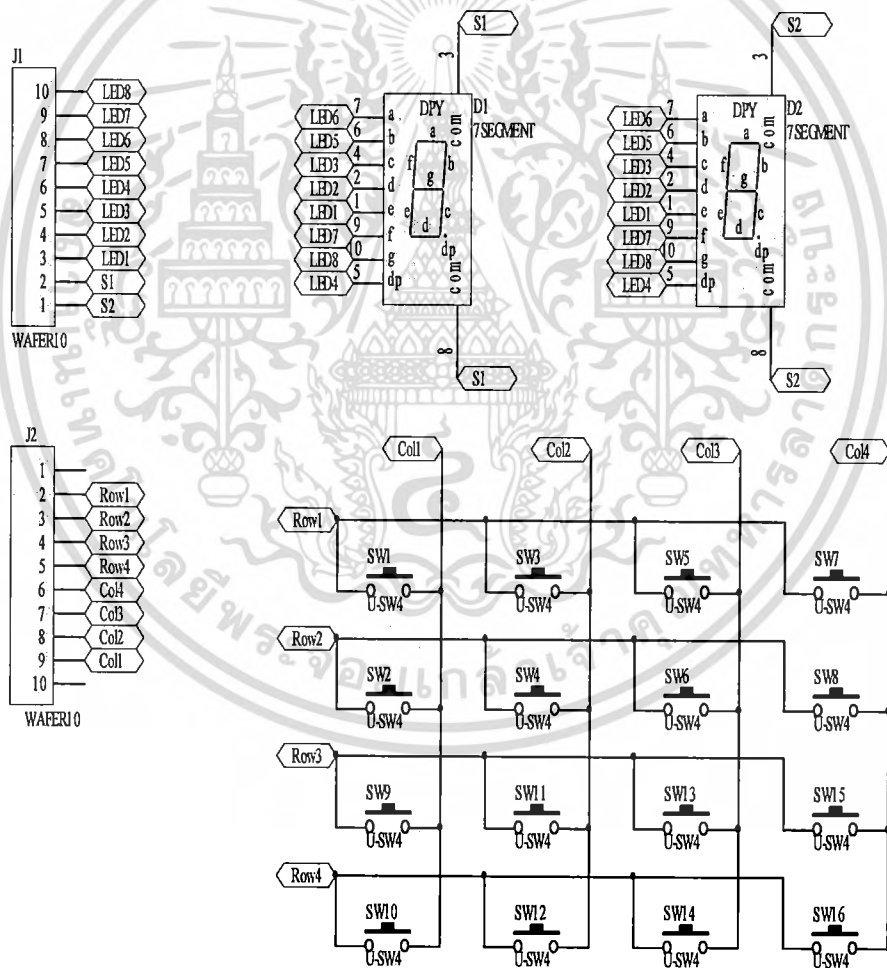


ภาพที่ 3.1 วงจรตัวแม่(Master)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

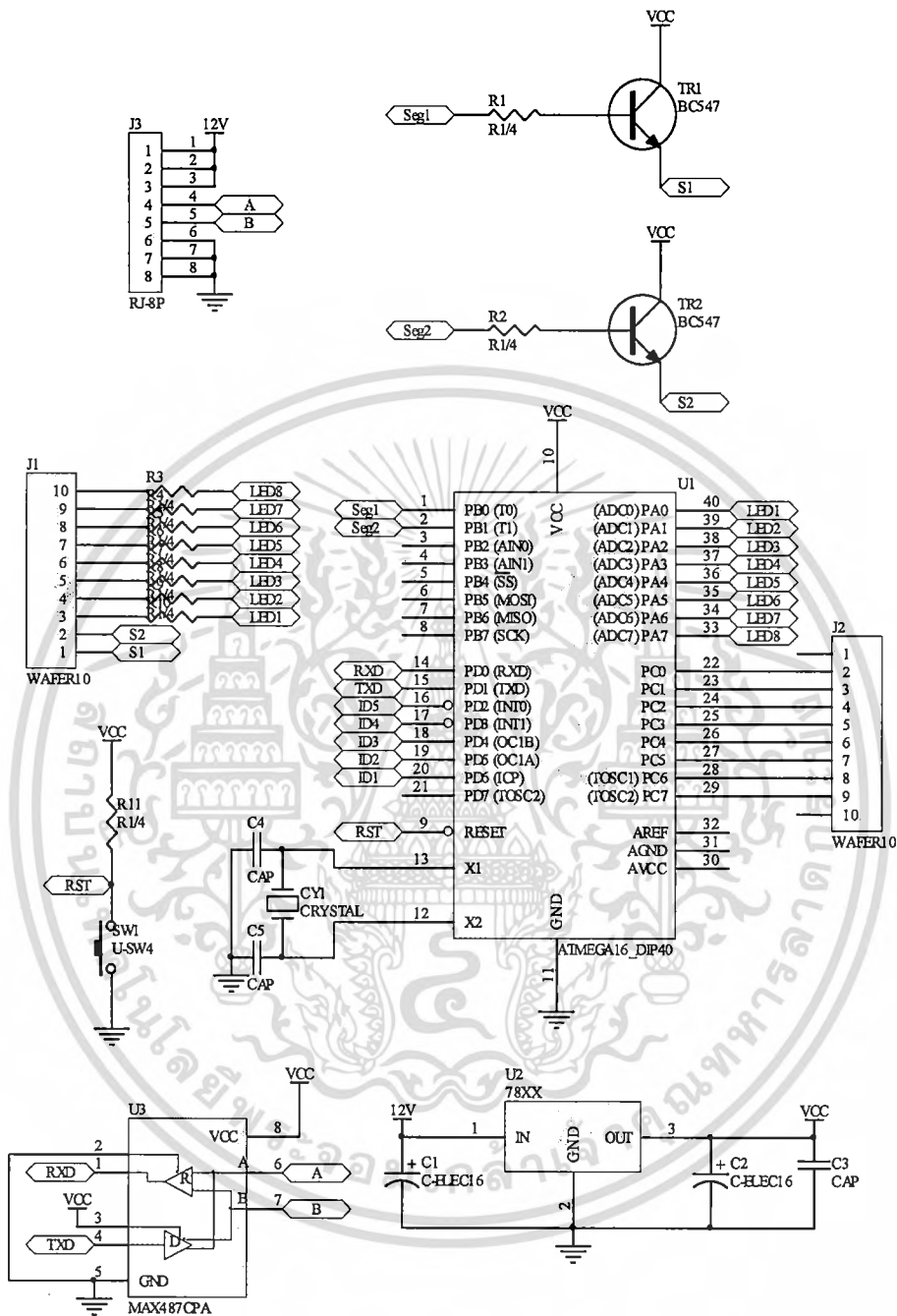
### 3.3.2 การออกแบบตัวลูก (Slave)

ใช้ไมโครคอนโทรลเลอร์ตระกูล AVR เบอร์ ATmega16 ในส่วนของการประมวลผล ใช้ไอซีเรีกูเลต เบอร์ 7805 เพื่อแปลงแรงดันให้เหลือแรงดันคงที่ 5 โวลต์ และป้อนให้ส่วนต่างๆภายในวงจร มีสวิตช์ใช้ก็ยข้อมูลต่างๆเพื่อติดต่อไปยังเครื่องแม่ มี seven segment ใช้แสดงผล



ภาพที่ 3.2 วงจรตัวลูก(Slave)1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



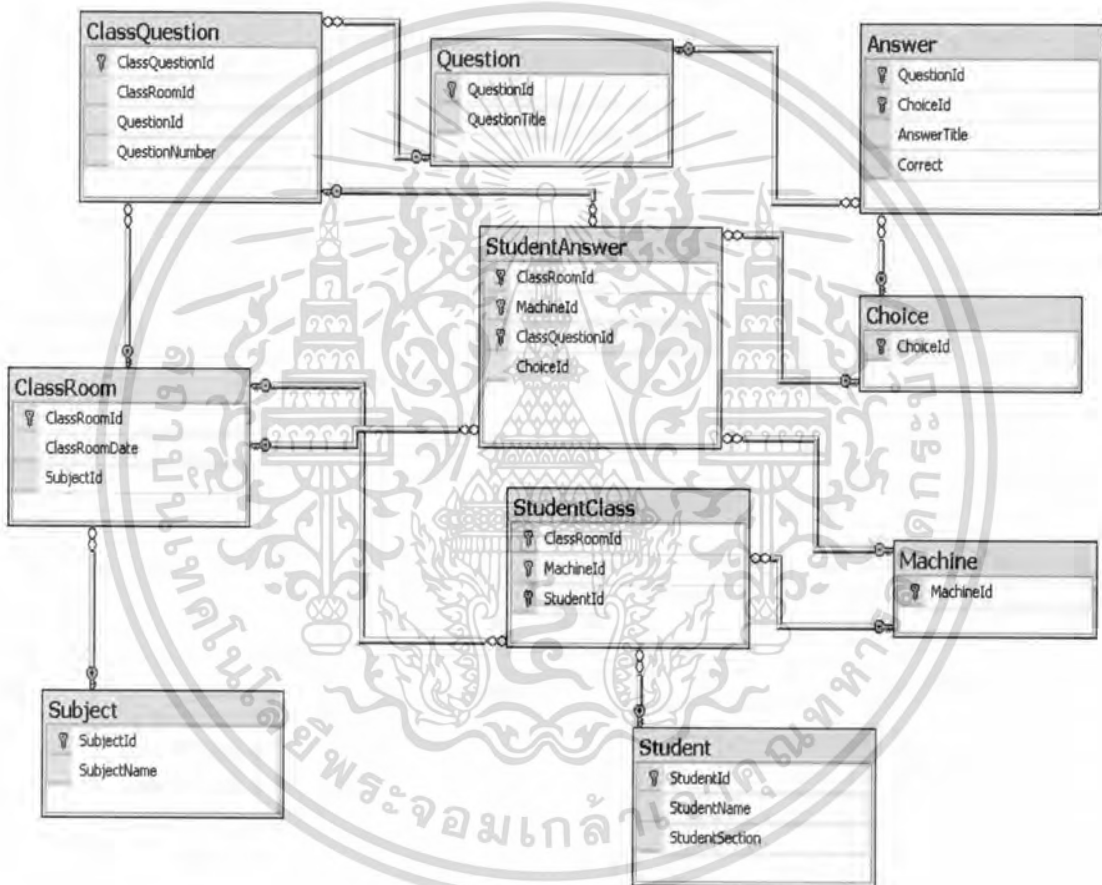
ภาพที่ 3.3 วงจรตัวลูก(Slave)2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### ผลการทดลอง

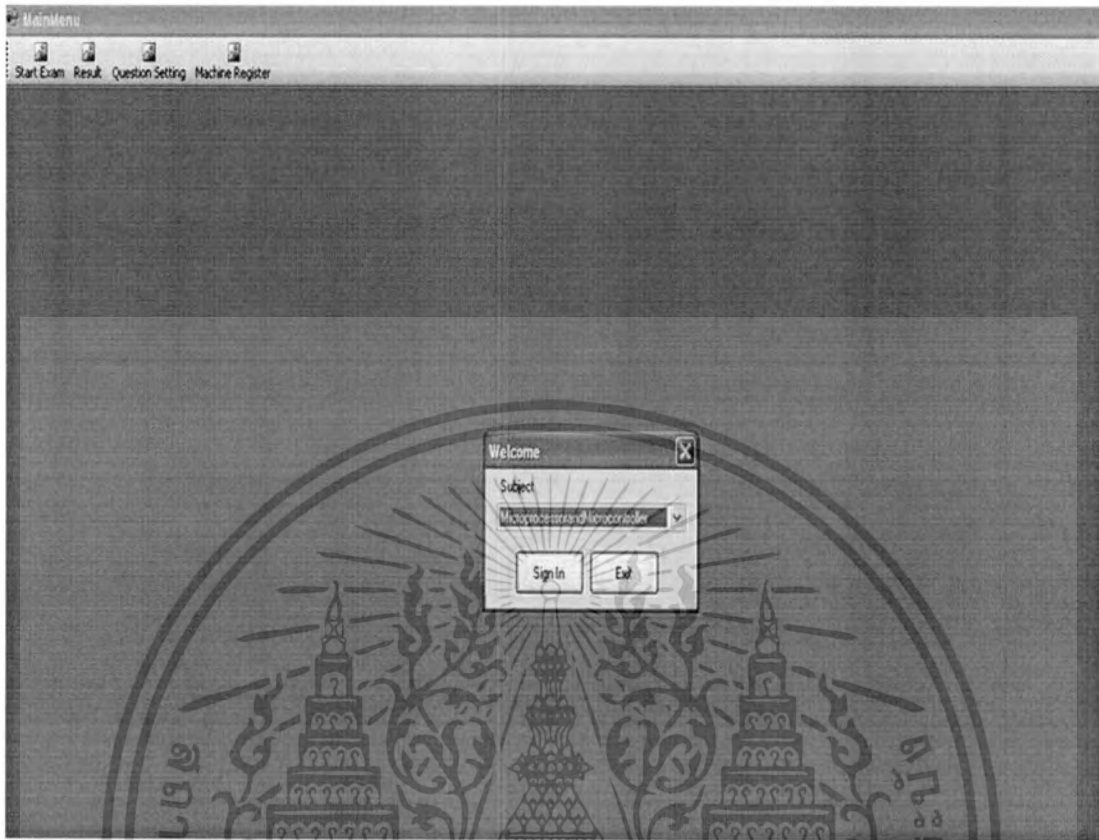
จากการออกแบบโครงการในบทที่ 3 ในบทนี้จะเป็นการนำมาปฏิบัติจริง โดยจัดสร้างตามที่ได้ ออกแบบไว้โดยมีการอ้างอิงจากทฤษฎีที่ได้อธิบายไว้ในบทที่ 2



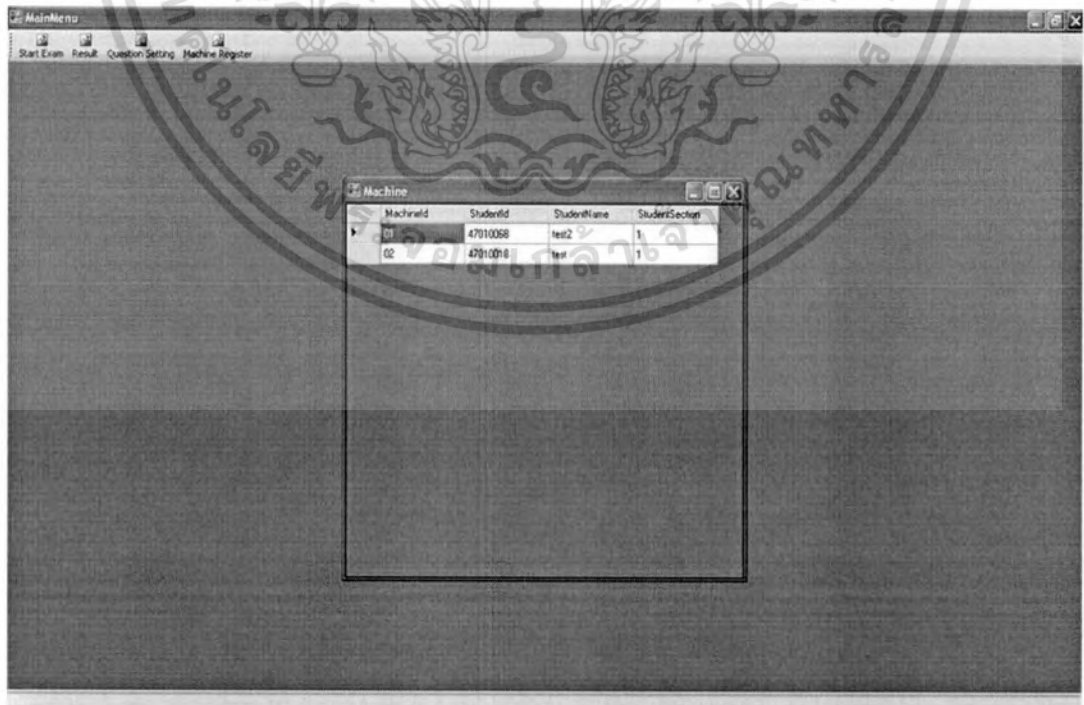
ภาพที่ 4.1 Database Diagram

จากการออกแบบหน้าต่างติดต่อผู้ใช้ โดยใช้โปรแกรม Microsoft Visual Studio 2008 โดยเขียนบน .NET framework 3.5 ด้วยภาษา C# จะได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

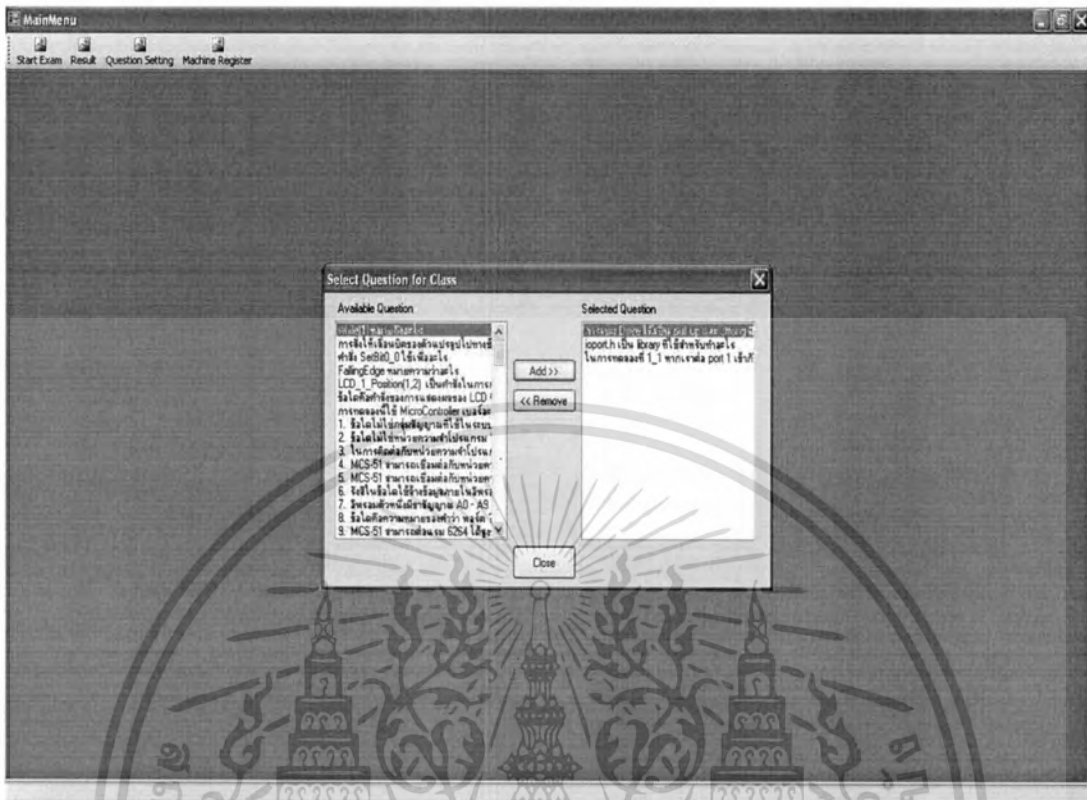


ภาพที่ 4.2 หน้าต่างเลือกวิชาที่สอน

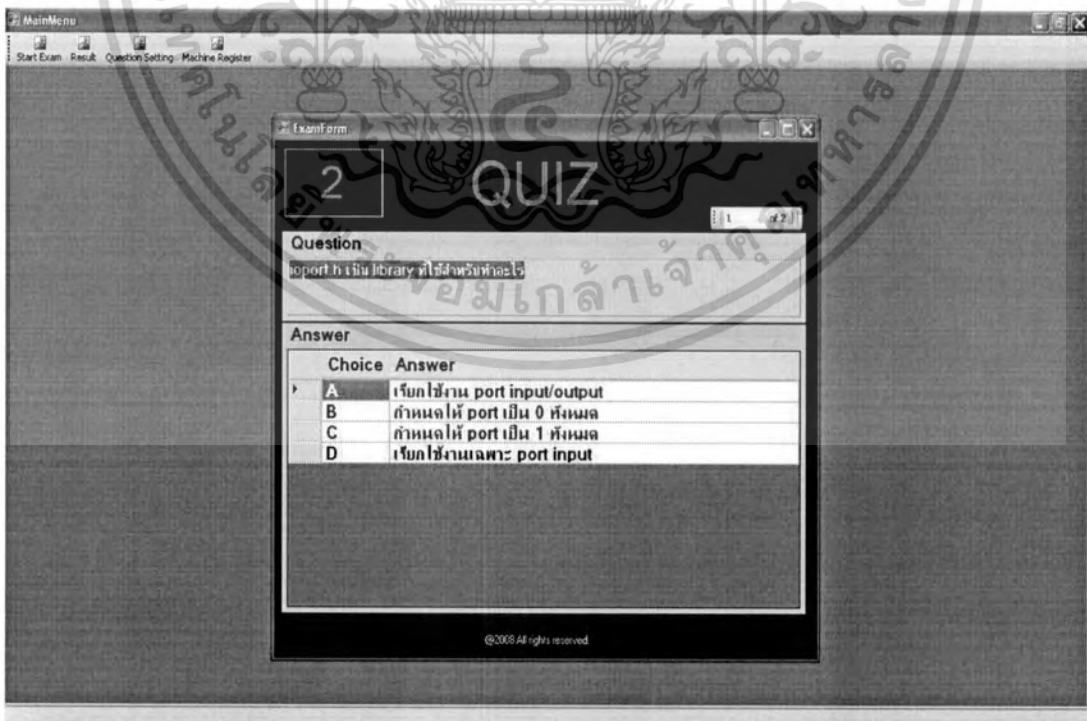


ภาพที่ 4.3 หน้าต่าง Machine Register

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

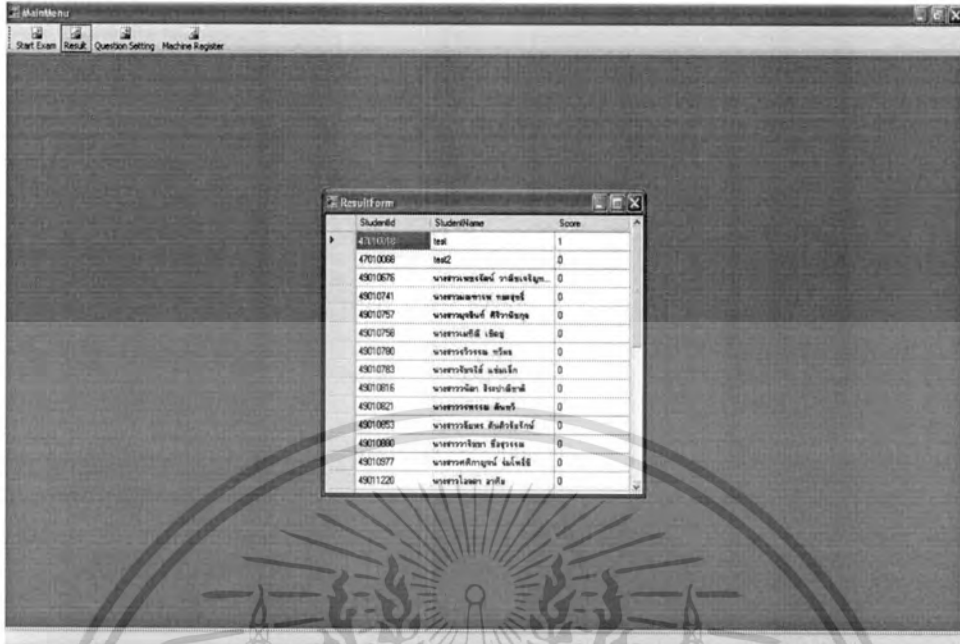


ภาพที่ 4.4 หน้าต่าง Question Setting



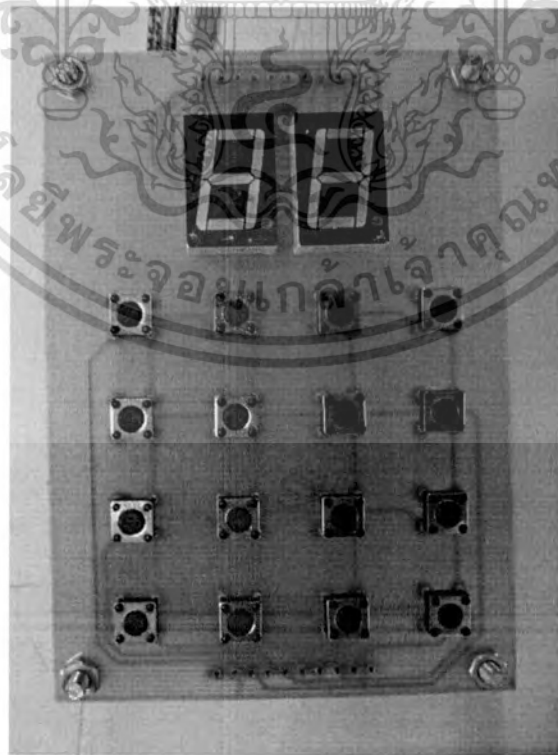
ภาพที่ 4.5 หน้าต่าง Start Exam

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



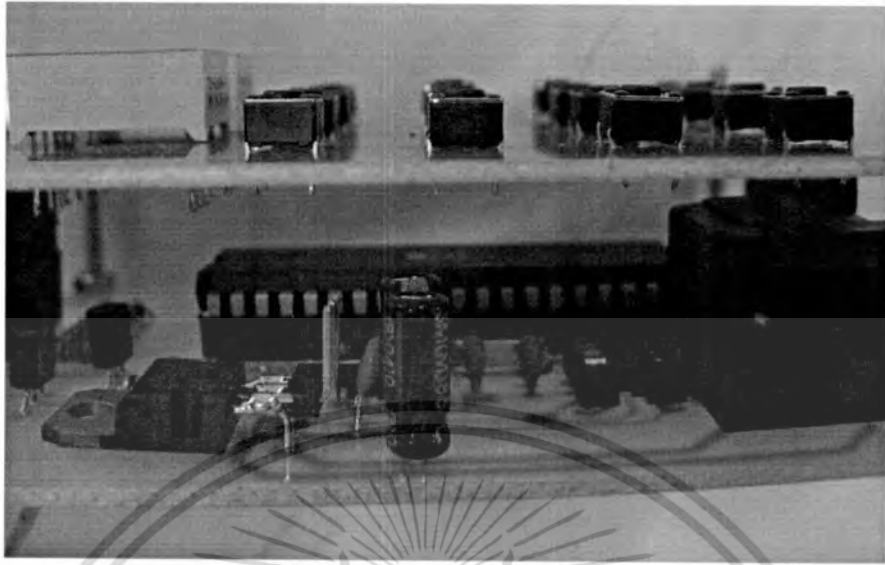
ภาพที่ 4.6 หน้าต่าง Result

จากการออกแบบวงจร และได้ลงอุปกรณ์ตามวงจรที่ได้ออกแบบไป จะได้อุปกรณ์ที่มีรูปร่าง  
ดังนี้

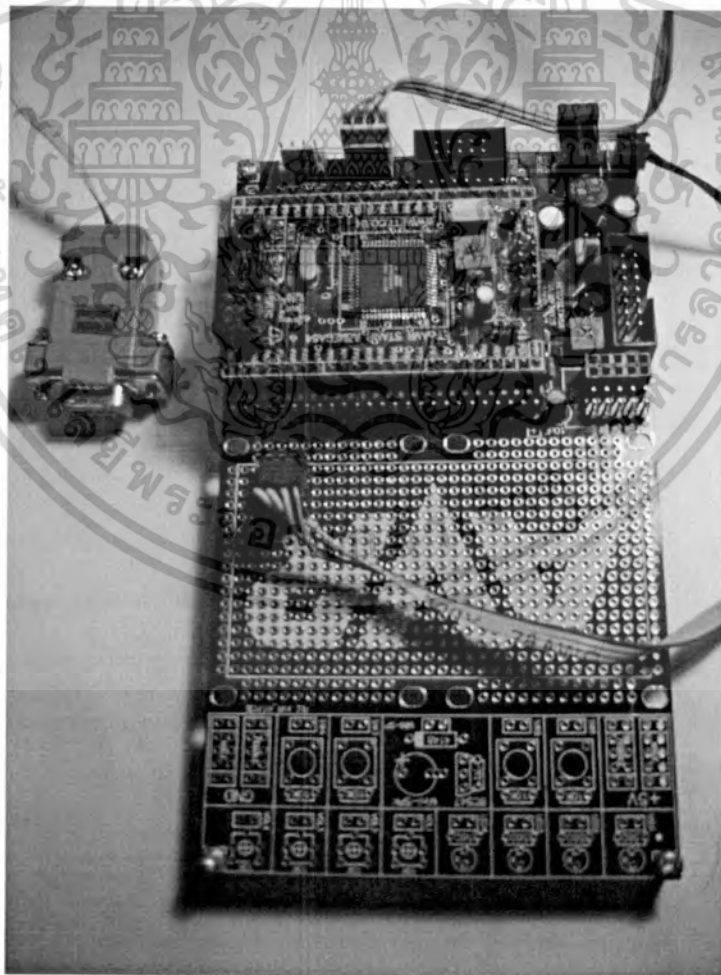


ภาพที่ 4.7 ตัวถูก(1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

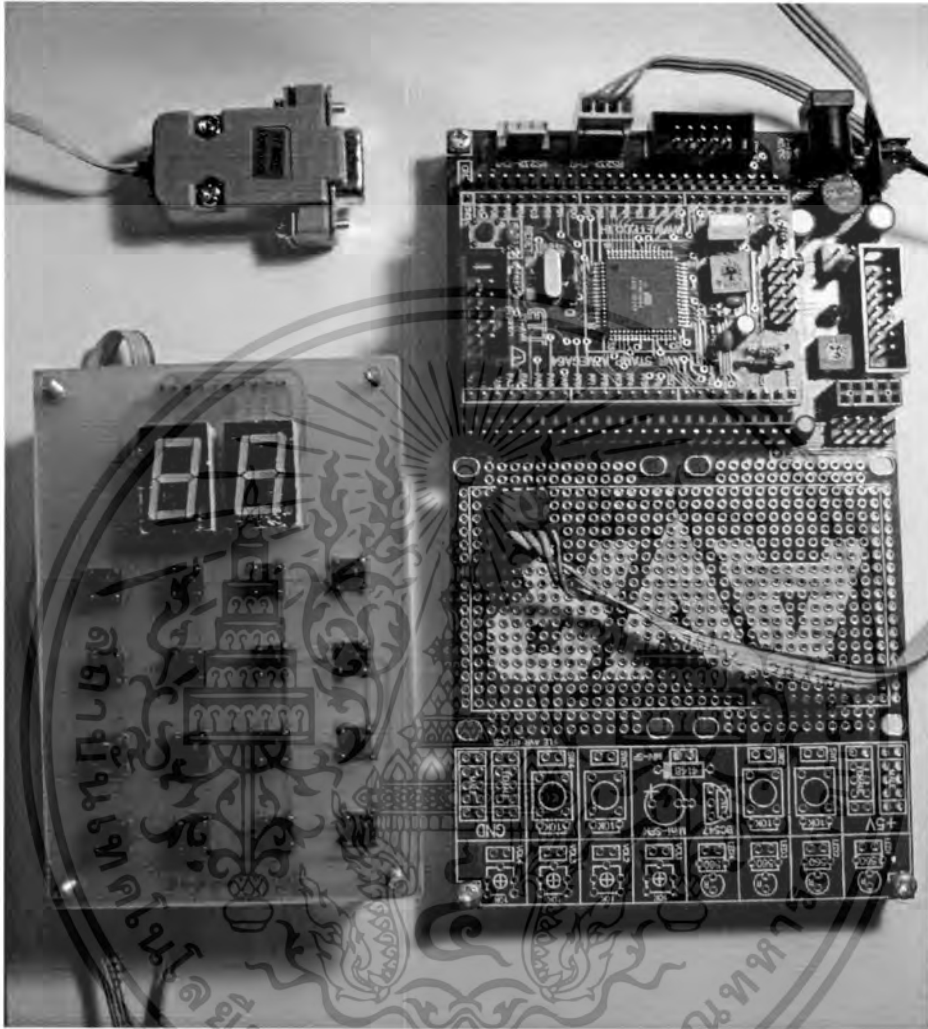


ภาพที่ 4.8 ตัวถุก(2)



ภาพที่ 4.9 ตัวแม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.10 รวมอุปกรณ์

หลังจากได้ค่ออุปกรณ์เรียบร้อยแล้ว ก็ได้มีการนำไปทดลองใช้

ระบบสามารถทำงานได้ครบตามที่ได้ออกแบบไว้ คือ ผู้เรียนสามารถส่งรหัสนักศึกษา เพื่อเป็นการตรวจสอบการเข้าเรียนและเวลาที่เข้าเรียน ส่งคำตอบในการตอบคำถาม โดยผ่านเครื่องลูก แต่ละเครื่องที่ตนมีอยู่ ทั้งคำตอบและรหัสนักศึกษาที่ส่งมาจะถูกส่งไปยังเครื่องแม่ เพื่อที่เครื่องแม่ จะทำการส่งข้อมูลทั้งหมดไปยังคอมพิวเตอร์ คอมพิวเตอร์จะทำการจัดเก็บข้อมูลลงในฐานข้อมูล และแสดงผลคะแนนของนักศึกษาเป็นรายบุคคล ผลคะแนนนี้จะจัดเก็บเป็นไฟล์นามสกุล .txt

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### สรุปผลและวิจารณ์ผลการทดลอง

#### 5.1 สรุปผล

จากการทดลอง ระบบสามารถทำงานตามที่ได้ออกแบบไว้ ในส่วนของหน้าต่างติดต่อผู้ใช้ สามารถจัดการกับฐานข้อมูลได้ตามที่ออกแบบไว้ ตรวจสอบคำตอบและแสดงผลคะแนนได้ ในส่วนของฮาร์ดแวร์ สามารถรับข้อมูลจากเครื่องลูก ส่งผ่านเครื่องแม่ไปยังเครื่องคอมพิวเตอร์ได้

#### 5.2 วิจารณ์ผลการทดลอง

จากผลการทดลอง พบว่าระบบสามารถทำงานได้ แต่ยังพบว่ามีความยุ่งยากในการใช้งาน เนื่องจากใช้สายแลน จึงใช้งานยากเพราะสายแลนแข็ง คัดยาก ในการเก็บเครื่องลูก เราจะเก็บสายโดยม้วนสายเป็นวงกลมล้อมรอบเครื่องลูก ดังนั้นสายจะโค้งงอทำให้ใช้งานได้ไม่สะดวก

#### 5.3 แนวทางการพัฒนาในอนาคต

1. สามารถเปลี่ยนจากสายแลนเป็นระบบไร้สาย เพื่อความสะดวกในการใช้งานและการเคลื่อนย้ายอุปกรณ์
2. สามารถเปลี่ยนปุ่มกดเป็นคีย์แพด เพื่อให้สะดวกต่อการใช้งานและดูสวยงาม
3. พัฒนาซอฟต์แวร์ให้สะดวกต่อการใช้งาน

## Frequencies

### Statistics

		มีความรู้ความ สามารถในห้วง ข้อที่บรรยาย	สามารถถ่ายทอด ความรู้และเนื้อหา ได้อย่างชัดเจน	มีเทคนิคการบ รรยายที่น่าสนใจ	ตอบข้อซักก ถามได้ชัดเจน ตรงประเด็น	มีความรู้ความ สามารถในห้วง ข้อที่บรรยาย
N	Valid	81	81	81	81	81
	Missing	0	0	0	0	0

### Statistics

		สามารถถ่ายทอด ความรู้และเนื้อหา ได้อย่างชัดเจน	มีเทคนิคการบ รรยายที่น่าสนใจ	ตอบข้อซักก ถามได้ชัดเจน ตรงประเด็น	ความเหมาะสม ของสถานที่	อาหารและเ ครื่องดื่ม	ระยะเวลาในก ารจัดฝึกอบรม
N	Valid	81	81	81	81	81	81
	Missing	0	0	0	0	0	0

### Statistics

		ความพร้อมของ โสตทัศนูปกรณ์ (เสียง/คอมพิวเตอร์/LCD)	การประชาสัมพันธ์ โครงการ	ความน่าสนใจ ของโครงการ	ความสะดวกในก ารประสานงานก ่อนการฝึกอบรม	ความสะดวก ในการประสาน งานระหว่าง การฝึกอบรม
N	Valid	81	81	81	81	81
	Missing	0	0	0	0	0

### Statistics

		มีเนื้อหาการจัด สวดคล้องกับช ื่อโครงการ	เอกสารประกอ บการฝึกอบรม	การจัดฝึกอบรม บรรลุตามวัด ถูประสงค์ของ โครงการ	ได้รับความรู้ตรง กับความต้องการ	ได้รับความรู้เพิ่ มมากขึ้นหลัง การฝึกอบรม
N	Valid	81	81	81	81	81
	Missing	0	0	0	0	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Statistics**

		สามารถนำคว ามรู้ที่ได้ไปปร ะยุกต์ใช้ได้	AA1	AA2	BB1	CC1	DD1
N	Valid	81	81	81	81	81	81
	Missing	0	0	0	0	0	0

**Statistics**

		TAA	TOT
N	Valid	81	81
	Missing	0	0

**Frequency Table**

**มีความรู้ความสามารถในหัวข้อที่บรรยาย**

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	ดี	23	28.4	28.4	28.4
	ดีมาก	58	71.6	71.6	100.0
	Total	81	100.0	100.0	

**สามารถถ่ายทอดความรู้และเนื้อหาได้อย่างชัดเจน**

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	พอใช้	1	1.2	1.2	1.2
	ดี	28	34.6	34.6	35.8
	ดีมาก	52	64.2	64.2	100.0
	Total	81	100.0	100.0	

**มีเทคนิคการบรรยายที่น่าสนใจ**

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	พอใช้	4	4.9	4.9	4.9
	ดี	26	32.1	32.1	37.0
	ดีมาก	51	63.0	63.0	100.0
	Total	81	100.0	100.0	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตอบข้อซักถามได้ชัดเจน ตรงประเด็น**

	Frequency	Percent	Valid Percent	Cumulative Percent
Valid พอใช้	1	1.2	1.2	1.2
ดี	46	56.8	56.8	58.0
ดีมาก	34	42.0	42.0	100.0
Total	81	100.0	100.0	

**มีความรู้ความสามารถในหัวข้อที่บรรยาย**

	Frequency	Percent	Valid Percent	Cumulative Percent
Valid ดี	24	29.6	29.6	29.6
ดีมาก	57	70.4	70.4	100.0
Total	81	100.0	100.0	

**สามารถถ่ายทอดความรู้และเนื้อหาได้อย่างชัดเจน**

	Frequency	Percent	Valid Percent	Cumulative Percent
Valid ดี	25	30.9	30.9	30.9
ดีมาก	56	69.1	69.1	100.0
Total	81	100.0	100.0	

**มีเทคนิคการบรรยายที่น่าสนใจ**

	Frequency	Percent	Valid Percent	Cumulative Percent
Valid ควรปรับปรุง	1	1.2	1.2	1.2
พอใช้	5	6.2	6.2	7.4
ดี	23	28.4	28.4	35.8
ดีมาก	52	64.2	64.2	100.0
Total	81	100.0	100.0	

**ตอบข้อซักถามได้ชัดเจน ตรงประเด็น**

	Frequency	Percent	Valid Percent	Cumulative Percent
Valid พอใช้	4	4.9	4.9	4.9
ดี	36	44.4	44.4	49.4
ดีมาก	41	50.6	50.6	100.0
Total	81	100.0	100.0	

**ความเหมาะสมของสถานที่**

	Frequency	Percent	Valid Percent	Cumulative Percent
Valid พอใช้	6	7.4	7.4	7.4
ดี	47	58.0	58.0	65.4
ดีมาก	28	34.6	34.6	100.0
Total	81	100.0	100.0	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**อาหารและเครื่องดื่ม**

	Frequency	Percent	Valid Percent	Cumulative Percent
Valid ควรปรับปรุง	3	3.7	3.7	3.7
พอใช้	9	11.1	11.1	14.8
ดี	24	29.6	29.6	44.4
ดีมาก	45	55.6	55.6	100.0
Total	81	100.0	100.0	

**ระยะเวลาในการจัดฝึกอบรม**

	Frequency	Percent	Valid Percent	Cumulative Percent
Valid ควรปรับปรุง	5	6.2	6.2	6.2
พอใช้	18	22.2	22.2	28.4
ดี	41	50.6	50.6	79.0
ดีมาก	17	21.0	21.0	100.0
Total	81	100.0	100.0	

**ความพร้อมของสื่อทัศนูปกรณ์ (เสียง/คอมพิวเตอร์/LCD)**

	Frequency	Percent	Valid Percent	Cumulative Percent
Valid ควรปรับปรุง	2	2.5	2.5	2.5
พอใช้	12	14.8	14.8	17.3
ดี	36	44.4	44.4	61.7
ดีมาก	31	38.3	38.3	100.0
Total	81	100.0	100.0	

**การประชาสัมพันธ์โครงการ**

	Frequency	Percent	Valid Percent	Cumulative Percent
Valid ควรปรับปรุง	9	11.1	11.1	11.1
พอใช้	16	19.8	19.8	30.9
ดี	43	53.1	53.1	84.0
ดีมาก	13	16.0	16.0	100.0
Total	81	100.0	100.0	

**ความน่าสนใจของโครงการ**

	Frequency	Percent	Valid Percent	Cumulative Percent
Valid พอใช้	5	6.2	6.2	6.2
ดี	28	34.6	34.6	40.7
ดีมาก	48	59.3	59.3	100.0
Total	81	100.0	100.0	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



**ได้รับความรู้เพิ่มมากขึ้นหลังการฝึกอบรม**

	Frequency	Percent	Valid Percent	Cumulative Percent
Valid พอใช้	1	1.2	1.2	1.2
ดี	30	37.0	37.0	38.3
ดีมาก	50	61.7	61.7	100.0
Total	81	100.0	100.0	

**สามารถนำความรู้ที่ได้ไปประยุกต์ใช้ได้**

	Frequency	Percent	Valid Percent	Cumulative Percent
Valid พอใช้	4	4.9	4.9	4.9
ดี	31	38.3	38.3	43.2
ดีมาก	46	56.8	56.8	100.0
Total	81	100.0	100.0	



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บรรณานุกรม

1. ประจัน พลังสันติกุล. 2549. การเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์ AVR ด้วยภาษา C กับ WinAVR (C Compiler). 1. กรุงเทพฯ : บริษัท แอปซอฟต์เทค จำกัด.
2. Thaimicrotron. "RS-232." [Online].  
Available: <http://www.Thaimicrotron.com/>. 2007
3. JOBPUB. "RS-232." [Online].  
Available: <http://www.JOBPUB.com/>. 2007
4. Thaicyperu. "Database." [Online].  
Available: <http://www.thaicyperu.go.th/>. 2007



# ภาคผนวก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก

### ภาคผนวก ก Code Slave

```
#include <stdio.h>

#include <avr/io.h>

#include <avr/interrupt.h>

//*****

#define F_CPU 16000000UL

#include <util/delay.h>

#include <compat/deprecated.h>

#include <avr/eeprom.h>

//*****

#define KeyN_A 10

#define KeyN_B 11

#define KeyN_C 12

#define KeyN_D 13

#define NoneKey 16

#define Sub 17

#define Send 14
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define Clear 15

#define Enter 14

#define Cancel 15

#define NumID 8

#define FrameSide 5

#define StartFrame 0x02
#define StopFrame 0x03
#define Request_ID 0x04
#define Request_Ans 0x05

//*****

//0,1,2,3,4,5,6,7,8,9,A,b,C,d,E,F

//unsigned char DisSeg[16] =
{0x18,0x7B,0x2C,0x29,0x4B,0x89,0x88,0x3B,0x08,0x9,0x0A,0xC8,0x9C,0x68,0x8C,0x8E};

//{0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0xF8,0x80,0x90,0x88,0x83,0xC6,0xA1,0x86,0x8E,0x
FF}

//{0xF9,0xA4,0xB0,0x88,0x99,0x92,0x82,0x83,0xF8,0x80,0x90,0xC6,0xFF,0xC0,0xFF,0xA1,0x
FF};
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

const unsigned char DisSeg[] =
{0x18,0x7B,0x2C,0x29,0x4B,0x89,0x88,0x3B,0x08,0x09,0x0A,0xC8,0x9C,0x68,0xFF,0xFF,0x
FF,0xEF};

//const unsigned char DisSeg[] =
{0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0xF8,0x80,0x90,0x88,0x83,0xC6,0xA1,0xFF,0xFF,0xF
F};

const unsigned char ChangKeyToValue[] = {1,2,3,10,4,5,6,11,7,8,9,12,14,0,15,13,16,17};

const unsigned char KeyChar[] =
{'0','1','2','3','4','5','6','7','8','9','A','B','C','D','S','c','0','-'};

const unsigned char OutCheckKey[4] = {0xFE,0xFD,0xFB,0xF7};

unsigned char EnDot = 0;

unsigned char ID[NumID];

unsigned char ADDH = 0;

unsigned char ADDL = 0;

unsigned char Ans = 0;

unsigned char Recv[FrameSide];

unsigned char RecvCount = 0;

unsigned int c;

unsigned char wrecv = 0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

short WaitSendAns = 0;//0 No Send,1 Wait Send

short WaitSendID = 1;

//*****

void delay_us(unsigned int i){

    for(;i>0;i--){

        _delay_us(1);

    }

}

void delay_ms(unsigned int i){

    for(;i>0;i--){

        _delay_ms(1);

    }

}

//*****

void DisplaySegment(unsigned char Seg1,unsigned char Seg2){

    PORTA = DisSeg[Seg1];

    PORTA = DisSeg[Seg1];

    if(EnDot == 1){

        PORTA &=0xF7;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//PORTA &=0x8F;

}

sbi(PORTB,0);

//sbi(PORTB,1);

delay_us(10);

//cbi(PORTB,1);

cbi(PORTB,0);

PORTA = DisSeg[Seg2];

sbi(PORTB,1);

//sbi(PORTB,0);

delay_us(10);

//cbi(PORTB,1);

cbi(PORTB,1);

}

//*****

unsigned char ScanKey(){

    unsigned char ck;

    unsigned char i;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
unsigned char ref;

for(i = 0;i<4;i++){

    PORTC = OutCheckKey[i];

    PINC = PINC | 0xF0;

    //delay_ms(10);

    ck = PINC & 0xF0;

    switch(ck){

        case 0xE0:

            ref = (4*i);

            return ChangKeyToValue[ref];

        case 0xD0:

            ref = (4*i) + 1;

            return ChangKeyToValue[ref];

        case 0xB0:

            ref = (4*i) + 2;

            return ChangKeyToValue[ref];

        case 0x70:

            ref = (4*i) + 3;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        return ChangKeyToValue[ref];

    }

}

PORTA = 0xFF;

PORTB = 0x00;

return NoneKey;
}
/*
unsigned char ScanKey(){
    unsigned char ck;

    unsigned char i;

    unsigned char ref;

    for(i = 0;i<4;i++){

        PORTC = OutCheckKey[i];

        PINC = PINC | 0xF0;

        //delay_ms(10);

        ck = PINC & 0xF0;

        switch(ck){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
case 0xE0:

    ref = (4*i) + 3;

    return ChangKeyToValue[ref];

case 0xD0:

    ref = (4*i) + 2;

    return ChangKeyToValue[ref];

case 0xB0:

    ref = (4*i) + 1;

    return ChangKeyToValue[ref];

case 0x70:

    ref = (4*i);

    return ChangKeyToValue[ref];

}

}

PORTA = 0xFF;

PORTB = 0x00;

return NoneKey;

}*/

//*****
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void ResetID(){

    for(unsigned i = 0;i<NumID;i++){

        ID[i] = 0;

    }

}

//*****

short SetSlaveAddress(){

    unsigned char KeyAH,KeyAL,Key,i;

    do{

        DisplaySegment(ADDH,ADDL);

        KeyAH = ScanKey();

    }while((KeyAH ==

NoneKey)||((KeyAH==KeyN_A)||((KeyAH==KeyN_B)||((KeyAH==KeyN_C)||((KeyAH==KeyN_

D)));

    if(KeyAH == Enter){

        return 1;

    }

    if(KeyAH == Cancel){

        return 0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

for(i=0;i<100;i++){

    DisplaySegment(ADDL,KeyAH);

}

do{

    DisplaySegment(ADDL,KeyAH);

    KeyAL = ScanKey();

}while((KeyAL ==
NoneKey)||((KeyAL==KeyN_A)||((KeyAL==KeyN_B)||((KeyAL==KeyN_C)||((KeyAL==KeyN_D)
));

if(KeyAL == Enter){

    ADDH = ADDL;

    ADDL = KeyAH;

    eeprom_write_byte(0x0000,ADDH);

    eeprom_write_byte(0x0001,ADDL);

    return 1;

}

if(KeyAL == Cancel){

    return 0;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

for(i=0;i<100;i++){

    DisplaySegment(KeyAH,KeyAL);

}

while(1){

    DisplaySegment(KeyAH,KeyAL);

    Key = ScanKey();

    if(Key == Enter){

        ADDH = KeyAH;

        ADDL = KeyAL;

        eeprom_write_byte(0x0000,ADDH);

        eeprom_write_byte(0x0001,ADDL);

        return 1;

    }

    if(Key == Cancel){

        return 0;

    }

}

return 0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

//*****

short SetID(){

    unsigned char Step,KeyID,i;

    unsigned char BufID[NumID];

    EnDot = 1;

    for(Step = 0;Step<NumID;Step++){

        do{

            DisplaySegment(Step+1,ID[Step]);

            KeyID = ScanKey();

        }while((KeyID ==

NoneKey)||((KeyID==KeyN_A)||((KeyID==KeyN_B)||((KeyID==KeyN_C)||((KeyID==KeyN_D));

            if(KeyID == Cancel){

                EnDot = 0;

                return 0;

            }else if(KeyID == Enter){

                for(i = 0;i<Step;i++){

                    ID[i] = BufID[i];

                }

            }

        }

    }

}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        EnDot = 0;

        return 1;

    }else{

        BufID[Step] = KeyID;

    }

    for(i=0;i<100;i++){

        DisplaySegment(Step+1,BufID[Step]);

    }

}

Step--;

do{

    DisplaySegment(Step+1,BufID[Step]);

    KeyID = ScanKey();

}while((KeyID != Cancel)&&(KeyID != Enter));

if(KeyID == Enter){

    for(i = 0;i < NumID;i++){

        ID[i] = BufID[i];

    }

    EnDot = 0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        return 1;

    }else{

        EnDot = 0;

        return 0;

    }

}

//*****

void put_string(unsigned char *s){

    PORTB |= 0x04;

    delay_us(5);

    while(*s != 0){

        loop_until_bit_is_set(UCSRA,UDRE);

        UDR = *s++;

    }

    delay_us(10);

    PORTB &= 0xFB;

}

//*****

void put_char(unsigned char ch){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

PORTB |= 0x04;

delay_us(5);

loop_until_bit_is_set(UCSRA,UDRE);

UDR = ch;

delay_us(10);

PORTB &= 0xFB;
}

//*****
void SendID(){
    unsigned char _SendID[NumID+4];
    unsigned char i ;
    _SendID[0] = StartFrame;
    _SendID[1] = ADDH + 0x30;
    _SendID[2] = ADDL + 0x30;

    for(i=3;i<(NumID+3);i++){

        _SendID[i] = ID[i-3] + 0x30;

    }

    _SendID[i] = StopFrame;

    PORTB |= 0x04;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
delay_us(5);

for(i = 0;i<(NumID+4);i++){

    loop_until_bit_is_set(UCSRA,UDRE);

    UDR = _SendID[i];

}

delay_us(10);

PORTB &= 0xFB;
}

//*****
void SendAns(){

    unsigned char _SendAns[5];

    unsigned char i ;

    _SendAns[0] = StartFrame;

    _SendAns[1] = ADDH + 0x30;

    _SendAns[2] = ADDL + 0x30;

    _SendAns[3] = Ans;

    _SendAns[4] = StopFrame;

    PORTB |= 0x04;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

Key = ScanKey();

}while((Key != Send)&&(Key != Cancel));

if(Key == Send){

    WaitSendAns = 1;

    //SendAns();

}else if(Key == Cancel){

    Ans = 0;

    WaitSendAns = 0;

}

}

}

//*****

static void USART_Init(unsigned int baud){

    UBRRH = (unsigned char)(baud>>8);

    UBRL = (unsigned char)baud;

    UCSRB = (1<<RXEN) | (1<<TXEN);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
delay_us(5);

for(i = 0;i<5;i++){

    loop_until_bit_is_set(UCSRA,UDRE);

    UDR = _SendAns[i];

}

delay_us(10);

PORTB &= 0xFB;

}

//*****

void WaitSend(unsigned char Buf){

    unsigned char i;

    unsigned char Key;

    Ans = KeyChar[Buf];

    do{

        for(i=0;i<50;i++){

            DisplaySegment(Sub,Buf);

        }

        for(i=0;i<50;i++){

            DisplaySegment(NoneKey,NoneKey);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
UCSRB |= (1<<RXCIE) | (1<<TXCIE);

sei();

UCSRC = (1<<URSEL) | (3<<UCSZ0);

}

//*****

void Timer_Init(){

    TCCR0 = (1<<CS02) | (0<<CS01) | (0<<CS00);

}

//*****

void Initial(){

    DDRA = 0xFF;

    DDRB = 0x07;

    DDRC = 0x0F; // 0 input;1 output

    PORTB = 0x03;

    Timer_Init();

    USART_Init(103); //71:11.5200;103 16(9600)

    cbi(PORTB,0);

    cbi(PORTB,1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ADDH = eeprom_read_byte(0x0000);

ADDL = eeprom_read_byte(0x0001);

ResetID();

}

//*****

int main(void){

    unsigned char Key = 0;

    unsigned char i = 0;

    unsigned char CountSetSlaveAddress = 0;

    unsigned char Buf = 0;

    unsigned int CountSetID = 0;

    unsigned int CountLoop = 0;

    Initial();

    do{

        Key = ScanKey();

        DisplaySegment(Sub,Sub);

        if(Key == KeyN_A){ //Key A

            CountSetID++;

            CountLoop = 0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}else{

    CountSetID = 0;

    CountLoop++;

}

}while((CountLoop <1200)&&(CountSetID <1000));

//*****

if(CountSetID>=1000){SetSlaveAddress

do{

    for(i=0;i<50;i++){

        DisplaySegment(ADDH,ADDL);

    }

    for(i=0;i<50;i++){

        DisplaySegment(NoneKey,NoneKey);

    }

    Key = ScanKey();

}while(Key != Enter);

delay_ms(2000);

SetSlaveAddress();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

delay_ms(1000);

CountSetSlaveAddress = 0;

}

//*****

do{ //Set ID

    EnDot = 1;

    for(i=0;i<50;i++){

        DisplaySegment(NoneKey,NoneKey);

    }

    EnDot = 0;

    for(i=0;i<50;i++){

        DisplaySegment(NoneKey,NoneKey);

    }

    Key = ScanKey();

}while(Key != Enter);

delay_ms(2000);

SetID();

CountSetID = 0;

//*****

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while(1){

    Key = ScanKey();

    if(Key != NoneKey){

        switch(Key){

            case KeyN_A:

                WaitSend(KeyN_A);

                break;

            case KeyN_B:

                WaitSend(KeyN_B);

                break;

            case KeyN_C:

                WaitSend(KeyN_C);

                break;

            case KeyN_D:

                WaitSend(KeyN_D);

                break;

        }

    }else{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

TCNT0 = 0;

TIMSK = (1<<TOIE0);

TIFR |= (1<<TOV0);

return;
}

//*****
ISR (USART_TXC_vect){

return;

}

//*****
ISR (TIMER0_OVF_vect){

unsigned char i;

TIMSK = (0<<TOIE0);

if((Recv[0] == (ADDH+0x30))&&(Recv[1] == (ADDL+0x30))){

    if((Recv[2] == Request_ID))

    {

        SendID();

        WaitSendID = 0;

    }else if((Recv[2] == Request_Ans)&&(WaitSendAns == 1)){

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        SendAns();

        WaitSendAns = 0;

    }

}

RecvCount = 0;

return;

}
```

### ภาคผนวก ข Code Master

```
#include <stdio.h>

#include <avr/io.h>

#include <avr/interrupt.h>

//*****

#define F_CPU 16000000UL

#include <util/delay.h>

#include <compat/deprecated.h>

//*****

#define NumSlave 60

#define NumID 8

#define FrameSide 12
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define StartFrame    0x02

#define StopFrame     0x03

#define Request_ID    0x04

#define Request_Ans  0x05
```

```
unsigned int c;
```

```
unsigned char CheckAddress;
```

```
unsigned char CheckID[NumSlave];
```

```
unsigned char CheckAns[NumSlave];
```

```
unsigned char Recv[FrameSide];
```

```
unsigned char RecvCount = 0;
```

```
short Wait = 0;
```

```
/*******
```

```
void delay_us(unsigned int i){
```

```
    for(;i>0;i--){
```

```
        _delay_us(1);
```

```
    }
```

```
}
```

```
void delay_ms(unsigned int i){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
for(;i>0;i--){

    _delay_ms(1);

}

}

/*****

void putUART0_string(unsigned char *s){

    while(*s != 0){

        loop_until_bit_is_set(UCSR0A,UDRE0);

        UDR0 = *s++;

    }

}

/*****

void putUART1_string(unsigned char *s){

    while(*s != 0){

        loop_until_bit_is_set(UCSR1A,UDRE1);

        UDR1 = *s++;

    }

}

void ResetCheck(){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    unsigned char i;

    for(i = 0;i < NumSlave;i++){

        CheckID[i] = 0;

        CheckAns[i] = 0;

    }

}

void SendIDtoPC(){

}

void SendAnstoPC(){

}

void SendRequestIDToSlave(unsigned char Addr){

    unsigned char _SendRequestID[5];

    unsigned char i ;

    /**

    _SendRequestID[3] = StartFrame;

    _SendRequestID[0] = (Addr/10) + 0x30;

    _SendRequestID[1] = (Addr%10) + 0x30;

    _SendRequestID[2] = Request_ID;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

_SendRequestID[4] = StopFrame;

*/

_SendRequestID[0] = StartFrame;

_SendRequestID[0] = (Addr/10) + 0x30;

_SendRequestID[1] = (Addr%10) + 0x30;

_SendRequestID[2] = Request_ID;

_SendRequestID[4] = StopFrame;

PORTA = 0x01;
delay_us(10);
for(i = 0;i<5;i++){
    loop_until_bit_is_set(UCSR1A,UDRE1);
    UDR1 = _SendRequestID[i];
    delay_us(10);
}

delay_us(50);

PORTA = 0x00;
}

void SendRequestAnsToSlave(unsigned char Addr){

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

unsigned char _SendRequestAns[5];

unsigned char i ;

/**

_SendRequestAns[3] = StartFrame;

_SendRequestAns[0] = (Addr/10) + 0x30;

_SendRequestAns[1] = (Addr%10) + 0x30;

_SendRequestAns[2] = Request_Ans;

_SendRequestAns[4] = StopFrame;

*/

_SendRequestAns[0] = StartFrame;

_SendRequestAns[0] = (Addr/10) + 0x30;

_SendRequestAns[1] = (Addr%10) + 0x30;

_SendRequestAns[2] = Request_Ans;

_SendRequestAns[4] = StopFrame;

PORTA = 0x01;

delay_us(10);

for(i = 0;i<5;i++){

loop_until_bit_is_set(UCSR1A,UDRE1);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        UDR1 = _SendRequestAns[i];

        delay_us(10);

    }

    delay_us(50);

    PORTA = 0x00;

}

//*****
void putUART0_char(unsigned char ch){
    loop_until_bit_is_set(UCSR0A,UDRE0);

    UDR0 = ch;
}

//*****
void putUART1_char(unsigned char ch){

    loop_until_bit_is_set(UCSR1A,UDRE1);

    UDR1 = ch;

}

//*****

static void USART0_Init(unsigned int baud){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
UBRR0H = (unsigned char)(baud>>8);

UBRR0L = (unsigned char)baud;

UCSR0B = (1<<RXEN0) | (1<<TXEN0);

UCSR0B |= (1<<RXCIE0) | (1<<TXCIE0);

UCSR0C = (1<<UCSZ01) | (1<<UCSZ00);
}
//*****
static void USART1_Init(unsigned int baud){

UBRR1H = (unsigned char)(baud>>8);

UBRR1L = (unsigned char)baud;

UCSR1B = (1<<RXEN1) | (1<<TXEN1);

UCSR1B |= (1<<RXCIE1) | (1<<TXCIE1);

UCSR1C = (1<<UCSZ11) | (1<<UCSZ10);
}
//*****
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void Timer_Init(){

    TCCR0 = (1<<CS02) | (0<<CS01) | (0<<CS00);

}

//*****

void Initial(){

    DDRA = 0xFF;

    PORTA= 0x00;

    USART0_Init(103);//9600;

    USART1_Init(103);//9600;

    Timer_Init();

    sei();

    ResetCheck();

}

//*****

int main(void)

{

    unsigned char Key;

    unsigned char i;

    Initial();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//putUART1_string("\nTest Master Uart1\n\r");

putUART0_string("\n Master \n\r");

delay_ms(2000);

while(1){

    for(i = 0;i<NumSlave;i++){

        CheckAddress = i;

        if(CheckID[i] == 0){

            //CheckID[i] = 1;

            SendRequestIDToSlave(i);

            RecvCount = 0;

            Wait = 0;

            TIFR |= (1<<TOV0);

            TCNT0 = 0;

            TIMSK = (1<<TOIE0);

            delay_ms(50);

            while(Wait == 0);

        }else if(CheckAns[i] == 0){

            SendRequestAnsToSlave(i);

            RecvCount = 0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Wait = 0;

        TIFR |= (1<<TOV0);

        TCNT0 = 0;

        TIMSK = (1<<TOIE0);

        delay_ms(50);

        while(Wait == 0);
    }
}
return 0;
}

//*****
ISR (USART0_RX_vect){

    c = UDR0;

    loop_until_bit_is_set(UCSR0A,UDRE0);

    UDR0 = c;

    putUART1_char(c);

    return;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//*****
ISR (USART0_TX_vect){

    return;

}

//*****

ISR (USART1_RX_vect){

    TIMSK = (0<<TOIE0);

    c = UDR1;

    loop_until_bit_is_set(UCSR1A,UDRE1);

    if(RecvCount<FrameSide){

        Recv[RecvCount] = c;

        RecvCount++;

    }

    TCNT0 = 0;

    TIMSK = (1<<TOIE0);

    TIFR = (1<<TOV0);

    return;

}

//*****

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ISR (USART1_TX_vect){

    return;

}

//*****

ISR (TIMER0_OVF_vect){

    unsigned char i;

    TIMSK = (0<<TOIE0);

    if(RecvCount> 5){ //ID

        CheckID[(((Recv[1]-0x30)*10)+(Recv[2]-0x30))] = 1;

        for(i = 0;i<RecvCount;i++){

            loop_until_bit_is_set(UCSR0A,UDRE0);

            UDR0 = Recv[i];

        }

    }

    else if(RecvCount>0){ //Ans

        CheckAns[(((Recv[1]-0x30)*10)+(Recv[2]-0x30))] = 1;

        for(i = 0;i<RecvCount;i++){

            loop_until_bit_is_set(UCSR0A,UDRE0);

            UDR0 = Recv[i];

        }

    }

}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }

}

Wait = 1;

RecvCount = 0;

return;
}

```

### ภาคผนวก ก MainForm.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Windows.Forms;

namespace ProjectSystem
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new MainForm());
        }

        public static int ClassroomId { get; set; }
    }
}

```

## ภาคผนวก ง MachineForm.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO.Ports;
using System.Diagnostics;
using System.Configuration;

namespace ProjectSystem
{
    public partial class MachineForm : Form
    {
        private SerialPort comport;

        public MachineForm()
        {
            InitializeComponent();
            string port = ConfigurationManager.AppSettings["Port"];
            int baudRate =
int.Parse(ConfigurationManager.AppSettings["BaudRate"]);
            int bitRate =
int.Parse(ConfigurationManager.AppSettings["DataBit"]);
            comport = new SerialPort(port, baudRate, Parity.None,
bitRate);
        }

        private void MachineForm_Load(object sender, EventArgs e)
        {
            viewStudentClassTableAdapter.FillByClassRoomId(databaseprojectDataSet
            .ViewStudentClass, Program.ClassRoomId);

            //.....Port Open.....//
            comport.Open();
            comport.DataReceived+=new
            SerialDataReceivedEventHandler(port_DataReceived);
            //.....//

        }
        private void port_DataReceived(object sender,
            SerialDataReceivedEventArgs e)
        {
            try
            {
                System.Threading.Thread.Sleep(1000);

                int bytes = comport.BytesToRead;

                byte[] buffer = new byte[bytes];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

buffer // Read the data from the port and store it in our
        comport.Read(buffer, 0, bytes);

//.....
.//

        int i = 0;

        while (true)
        {
            byte data = 0;
            do
            {
                data = buffer[i]; //ok

                i++;
                if (i >= buffer.Length) break;
            }
            while (data != 0x02);
            if (i >= buffer.Length) break;

            char a = Convert.ToChar(buffer[i]);
            char b = Convert.ToChar(buffer[i + 1]);
            string machineId = Convert.ToString(a) +
Convert.ToString(b);
            i += 2; //ok
            string dataString = "";

            while (true)
            {
                a = Convert.ToChar(buffer[i]);
                dataString += Convert.ToString(a);
                i++;
                if (i >= buffer.Length) break; //ok
                if (buffer[i] == 0x03) break;
            }

            //Debug.WriteLine("ClassRoomId="+Program.ClassRoomId + "MachineId = "
+ machineId + ", Data = " + dataString);

            studentClassTableAdapter.AddStudentClass(Program.ClassRoomId,
machineId, dataString);
            if (i >= buffer.Length) break; //ok

        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        private void MachineForm_FormClosed(object sender,
FormClosedEventArgs e)
        {
            comport.Close();
        }

        private void timer1_Tick(object sender, EventArgs e)
        {

viewStudentClassTableAdapter.FillByClassRoomId(databaseprojectDataSet
.ViewStudentClass, Program.ClassRoomId);
        }

        private void dataGridView1_CellContentClick(object sender,
DataGridViewCellEventArgs e)
        {
        }
    }
}

```

### ภาคผนวก ง QuestionSelectForm.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace ProjectSystem
{
    public partial class QuestionSelectForm : Form
    {
        public QuestionSelectForm()
        {
            InitializeComponent();
        }

        private void QuestionSelectForm_Load(object sender, EventArgs
e)
        {
            // TODO: This line of code loads data into the
            'databaseprojectDataSet.Question' table. You can move, or remove it,
            as needed.
            LoadData();
        }

        private void LoadData()
        {

```

```

this.questionTableAdapter.FillByAvailableQuestion(this.databaseproject
DataSet.Question, Program.ClassRoomId);

this.questionTableAdapter.FillBySelectedQuestion(this.databaseproject
DataSet1.Question, Program.ClassRoomId);
}

private void addButton_Click(object sender, EventArgs e)
{
    if (availableListBox.SelectedItems.Count > 0)
    {
classQuestionTableAdapter.AddClassQuestion(Program.ClassRoomId,
(int)availableListBox.SelectedValue, selectedListBox.Items.Count +
1);
        LoadData();
    }
}

private void removeButton_Click(object sender, EventArgs e)
{
    if (selectedListBox.SelectedItems.Count > 0)
    {
classQuestionTableAdapter.DeleteClassQuestion(Program.ClassRoomId,
(int)selectedListBox.SelectedValue);
        LoadData();
    }
}

private void okButton_Click(object sender, EventArgs e)
{
    this.Close();
}
}
}

```

### ภาคผนวก ง ExamForm.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO.Ports;
using System.Configuration;

namespace ProjectSystem
{
    public partial class ExamForm : Form
    {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

private SerialPort comport;
private int classQuestionId;
public ExamForm()
{
    InitializeComponent();
    string port = ConfigurationManager.AppSettings["Port"];
    int baudRate =
int.Parse(ConfigurationManager.AppSettings["BaudRate"]);
    int bitRate =
int.Parse(ConfigurationManager.AppSettings["DataBit"]);
    comport = new SerialPort(port, baudRate, Parity.None,
bitRate);
}

private void ExamForm_Load(object sender, EventArgs e)
{
    // TODO: This line of code loads data into the
'databaseprojectDataSet.ViewClassQuestion' table. You can move, or
remove it, as needed.

this.viewClassQuestionTableAdapter.FillByClassRoomId(this.databasepro
jectDataSet.ViewClassQuestion, Program.ClassRoomId);
    // TODO: This line of code loads data into the
'databaseprojectDataSet.Answer' table. You can move, or remove it, as
needed.

    DataRowView classQuestionRow =
viewClassQuestionBindingSource.Current as DataRowView;
    int questionId = (int)classQuestionRow["QuestionId"];
    classQuestionId =
(int)classQuestionRow["ClassQuestionId"];

this.answerTableAdapter.FillByQuestionId(this.databaseprojectDataSet.
Answer, questionId);
    comport.Open();
    comport.DataReceived += new
SerialDataReceivedEventHandler(port_DataReceived);
}

private void
viewClassQuestionBindingSource_CurrentChanged(object sender,
EventArgs e)
{
    DataRowView classQuestionRow =
viewClassQuestionBindingSource.Current as DataRowView;
    int questionId = (int)classQuestionRow["QuestionId"];
    classQuestionId =
(int)classQuestionRow["ClassQuestionId"];

this.answerTableAdapter.FillByQuestionId(this.databaseprojectDataSet.
Answer, questionId);
}

private void port_DataReceived(object sender,
SerialDataReceivedEventArgs e)
{
    try
    {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

System.Threading.Thread.Sleep(1000);

int bytes = comport.BytesToRead;
byte[] buffer = new byte[bytes];

// Read the data from the port and store it in our
buffer
comport.Read(buffer, 0, bytes);

//.....
.//

int i = 0;
while (true)
{
    byte data = 0;
    do
    {
        data = buffer[i]; //ok
        i++;
        if (i >= buffer.Length) break;
    }
    while (data != 0x02);
    if (i >= buffer.Length) break;

    char a = Convert.ToChar(buffer[i]);
    char b = Convert.ToChar(buffer[i + 1]);
    string machineId = Convert.ToString(a) +
Convert.ToString(b);
    i += 2; //ok
    string dataString = "";

    while (true)
    {
        a = Convert.ToChar(buffer[i]);
        dataString += Convert.ToString(a);
        i++;
        if (i >= buffer.Length) break; //ok
        if (buffer[i] == 0x03) break;
    }

    //Debug.WriteLine("ClassRoomId="+Program.ClassRoomId +"MachineId = "
+ machineId + ", Data = " + dataString);

    studentAnswerTableAdapter1.AddStudentAnswer(Program.ClassRoomId,
machineId, classQuestionId, dataString);
        if (i >= buffer.Length) break;    //ok

}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void ExamForm_FormClosed(object sender,
FormClosedEventArgs e)
{
    comport.Close();
}

private void timer1_Tick(object sender, EventArgs e)
{
    try //myexception
    {
        int remain =
(int)studentAnswerTableAdapter1.StudentAnswerRemain(classQuestionId,
Program.ClassRoomId);
        answerRemainLabel.Text = remain.ToString();
        if (remain == 0)
        {
            viewClassQuestionBindingSource.MoveNext();
        }
    }
    catch(Exception ex)//myexception
    {
        MessageBox.Show(ex.Message);
    }
}
}
}

```

### ภาคผนวก ข ResultForm.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO;
using System.Data.SqlClient;
namespace ProjectSystem
{
    public partial class ResultForm : Form
    {
        public ResultForm()
        {
            InitializeComponent();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }

    private void ResultForm_Load(object sender, EventArgs e)
    {
        this.studentScoreTableAdapter.Fill(databaseprojectDataSet.StudentScore, Program.ClassRoomId);
        //.....Test.....//
        FileInfo file = new FileInfo(@"C:/printfile/test3.txt");
        StreamWriter writer = file.CreateText();
        writer.WriteLine("StudentID \t StudentName \t
Score");

        //writer.WriteLine(databaseprojectDataSet.StudentScore[1][1].ToString());

        //writer.WriteLine(databaseprojectDataSet.StudentScore[1][2].ToString());

        foreach (DataRow row in databaseprojectDataSet.StudentScore.Rows)
        {
            writer.WriteLine();
            foreach (DataColumn c in databaseprojectDataSet.StudentScore.Columns)
            {
                writer.Write(row[c].ToString()+"\t");
            }
            writer.Close();
        }
        //.....//
    }
}

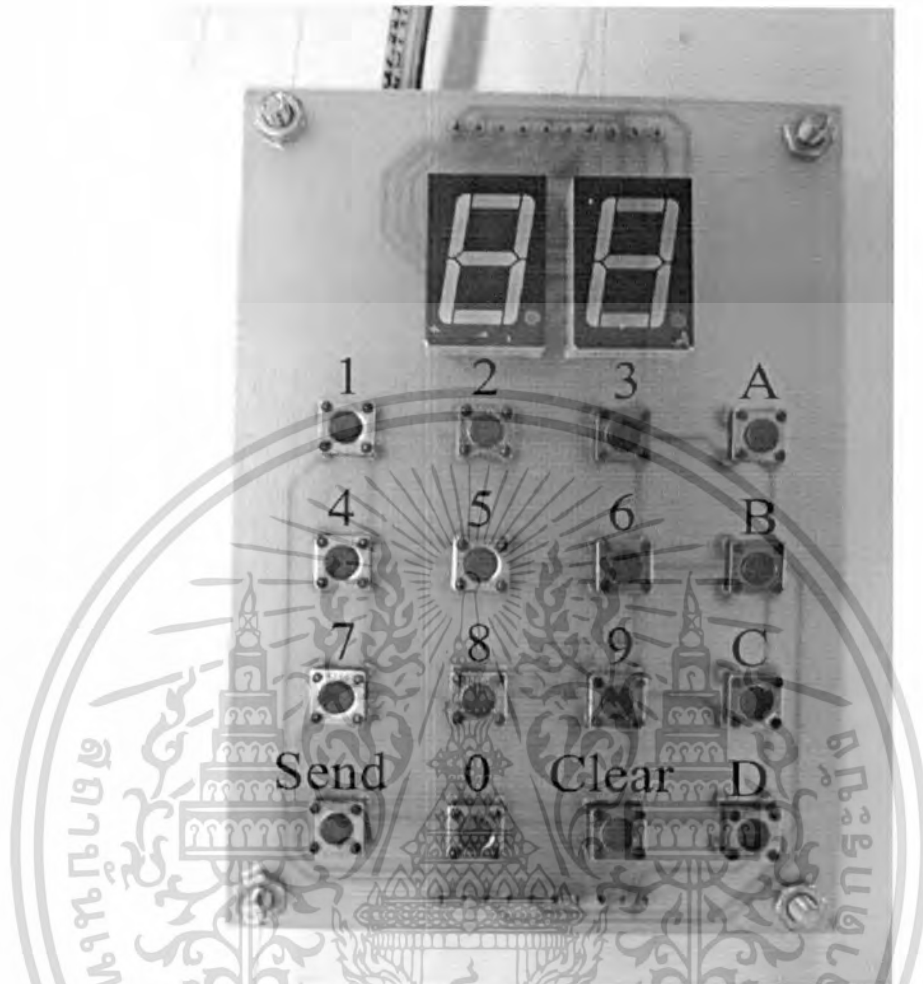
```

### ภาคผนวก ข คู่มือการใช้งานเครื่อง Slave

#### การตั้งค่าหมายเลขเครื่อง Slave

- 1) กดรีเซตที่ปุ่มรีเซตบนเครื่อง (ปุ่มรีเซตจะอยู่บนบอร์ดชั้นที่สอง)
- 2) กดปุ่ม A ค้างจนกว่าจะมีตัวอักษรกระพริบ
- 3) กดปุ่ม Send
- 4) ใส่หมายเลขเครื่อง
- 5) กดปุ่ม Send

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ ผ.1 อธิบายปุ่มกด

### การใส่รหัสนักศึกษา

- 1) เมื่อเห็นจุดกระพริบแสดงบน seven segment ด้านซ้ายมือ ก็กดปุ่ม send
- 2) หลังจากกดปุ่ม send แล้วจะเห็นเลขหนึ่งแสดงบน seven segment ด้านซ้ายมือ หมายถึง รหัสนักศึกษาตัวแรก ให้ใส่รหัสนักศึกษาตัวแรกลงไป
- 3) เมื่อใส่รหัสนักศึกษาตัวแรกแล้ว seven segment ด้านซ้ายมือจะแสดงเลขสอง หมายถึง รหัสนักศึกษาตัวที่สอง ให้ใส่รหัสนักศึกษาตัวที่สอง จะเป็นอย่างไรจนครบแปดครั้ง (รหัส นักศึกษามีแปดตัว)
- 4) เมื่อใส่รหัสนักศึกษาครบแล้ว ก็กดปุ่ม send

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 5) เมื่อใส่รหัสนักศึกษาเรียบร้อยแล้ว seven segment ด้านซ้ายมือ จะแสดงเครื่องหมายขีดตรงกลาง

#### การตอบคำถาม

- 1) ให้เลือกคำตอบโดยกดปุ่ม A,B,C หรือ D
- 2) กดปุ่ม send

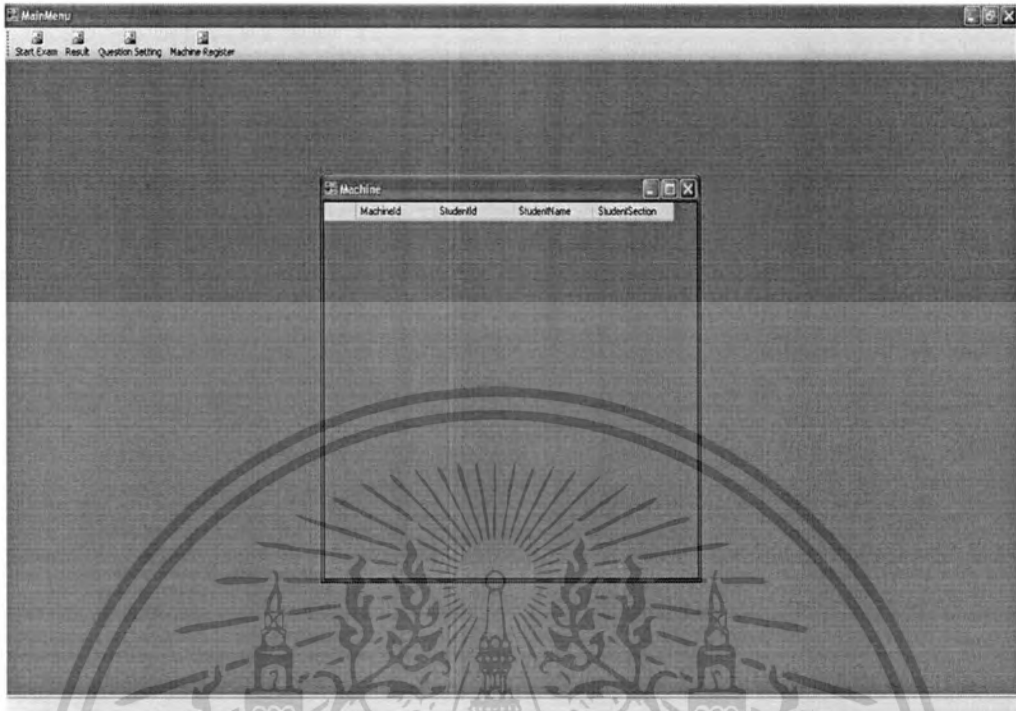
#### **ภาคผนวก ณ วิธีใช้งานโปรแกรม**

- 1) เปิดโปรแกรม ProjectSystem.exe จะพบหน้าต่างให้เลือกวิชาเรียน

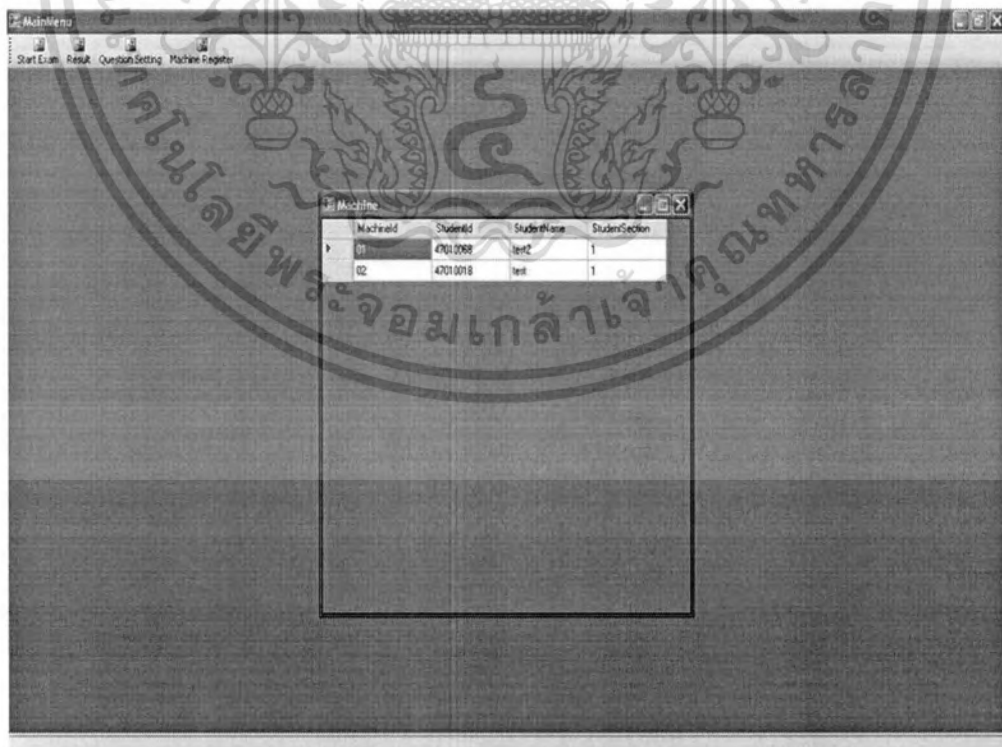


- 2) เลือกวิชาเรียนแล้วกด Sign In
- 3) เลือกปุ่ม Machine Register จะพบหน้าต่างดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

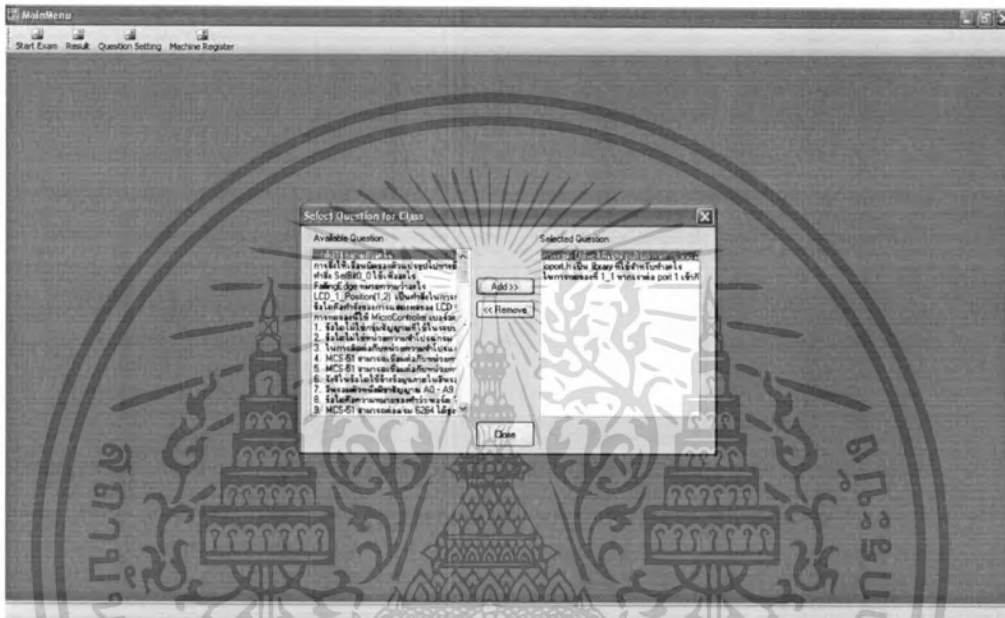


- 4) เมื่อนักศึกษาส่งรหัสมาครบทุกคนแล้ว ให้กดปุ่มรีเซตที่เครื่องแม่ (Master) 1 ครั้ง จะพบว่า มีรายชื่อนักศึกษาขึ้นมา ดังรูป

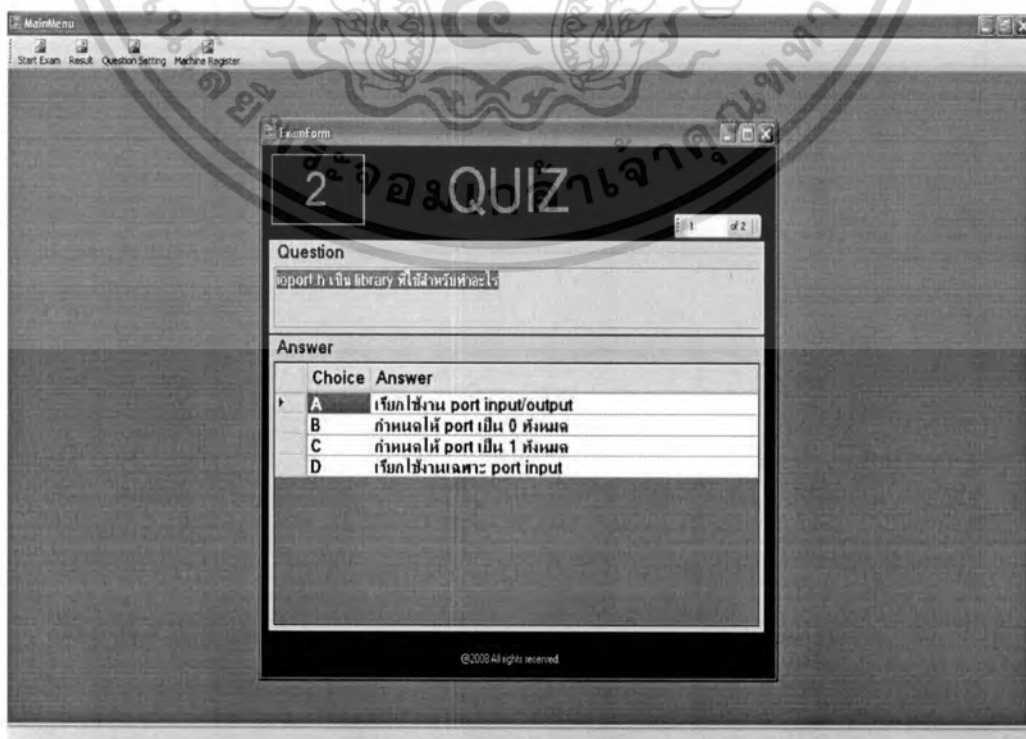


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 5) ปิดหน้าต่าง หลังจากนั้นกดปุ่ม Question Setting เพื่อเลือกคำถาม โดยคลิกเลือกคำถามที่ต้องการจากกล่องด้านซ้ายมือ แล้วกด Add>> ถ้าต้องการเปลี่ยนคำถามที่เลือกให้ เลือกคำถามที่ต้องการเปลี่ยนที่กล่องด้านขวามือ แล้วกด <<Remove เมื่อเลือกคำถามครบแล้ว กด Close ดังรูป



- 6) เลือกปุ่ม Start Exam เพื่อเริ่มต้นการทดสอบ

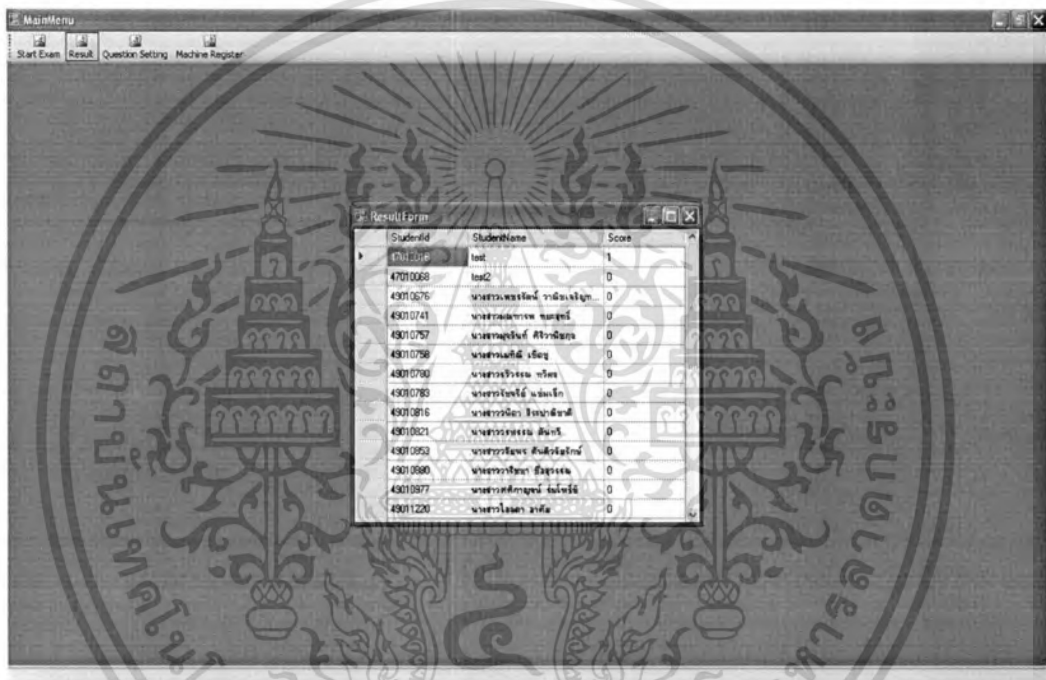


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่กล่องด้านบนขวา จะแสดงจำนวนข้อของคำถาม และแสดงว่าคำถามข้อนั้นเป็นคำถามข้อที่เท่าไร? กล่องด้านซ้ายมือ จะแสดงจำนวนครั้งที่ยังไม่ส่งคำตอบในข้อนั้นๆ

7) เมื่อนักศึกษาตอบคำถามครบทุกข้อแล้วให้ปิดหน้าต่างนี้ไป

8) กดที่ปุ่ม Result เพื่อดูคะแนนของนักศึกษา และผลคะแนนนี้จะถูกเก็บไว้ใน File นามสกุล .txt



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้