

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

อุปกรณ์เก็บข้อมูลเครื่องขายของอัตโนมัติโดยใช้ GPRS

An equipment for collecting data from automatic vending machines  
by GPRS



โดย  
นายพลากร พุ่มเพ็ญ  
นายเสถียรพงศ์ ใจเย็น

พ.2550  
2549

เลขหมู่.....  
เลขทะเบียน..... 71968  
วัน,เดือน,ปี..... - 7 ส.ย. 2550

b. 61155  
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

ผ่านการตรวจรับงานแล้ว  
(ลงชื่อ).....ผู้ตรวจ

ผ่านการตรวจรูปเล่มแล้ว

(ลงชื่อ).....ผู้ตรวจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# อุปกรณ์เก็บข้อมูลเครื่องขายของอัตโนมัติโดยใช้ GPRS

An equipment for collecting data from automatic vending machines  
by GPRS



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2549

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง **อุปกรณ์เก็บข้อมูลเครื่องขายของอัตโนมัติโดยใช้ GPRS**

**An equipment for collecting data from automatic vending machines by GPRS**

ผู้จัดทำ

1. นายพดกร คุ้มเพ็ญ 47015748

2. นายเสถียรพงศ์ ใจเย็น 47015751

.....  
( ผศ.สุรพล บุญจันทร์ )

อาจารย์ที่ปรึกษา

( ผศ.สุรพล บุญจันทร์ )



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## อุปกรณ์เก็บข้อมูลเครื่องขายของอัตโนมัติโดยใช้ GPRS

### An equipment for collecting data from automatic vending machines by GPRS

โดย นายพลากร ฟูมเพ็ญ 47015748

นายเสถียรพงศ์ ใจเย็น 47015751

อาจารย์ที่ปรึกษา ผศ.สุรพล บุญจันทร์

#### บทคัดย่อ

เนื่องจาก เครื่องขายของอัตโนมัติ ที่เปิดให้บริการอย่างแพร่หลายในปัจจุบัน เมื่อสินค้าในเครื่องขายของอัตโนมัติหมด ทางผู้ประกอบการจะไม่สามารถทราบและทำการแก้ไขได้ในทันที ทำให้เกิดผลกระทบต่อการค้าเงินธุรกิจประเภทนี้มาก

โครงการนี้จึงได้จัดทำระบบแจ้งเตือนเมื่อสินค้าในตู้ขายของใกล้จะหมด โดยที่เครื่องขายของอัตโนมัติจะทำการติดตั้ง Micro switch เพื่อทำการตรวจว่าสินค้าภายในเครื่องใกล้จะหมด เมื่อพบว่าสินค้าหมด ระบบจะสั่งให้โทรศัพท์ไร้สายซึ่งการควบคุมโดยใช้ชุดคำสั่งของ AT-Command ทำการส่งข้อมูลผ่านระบบ GPRS เข้าสู่ Web mail โดยจะแจ้งผู้ประกอบการให้ทราบว่าตำแหน่งของเครื่องอยู่ที่ไหนและสินค้าประเภทใดหมด

#### ABSTRACT

Sell automatic machines which is open service extensively now have the problem that the entrepreneur cannot know that some goods are run out. To solve this problem, this project then purposes the system which can inform the entrepreneur when goods in the cabinet are out of order. In this system, the Micro-switch is employed to check goods. The system then uses the stock of the wireless telephone to send data via Web mail for inform the entrepreneur.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ได้จัดทำขึ้นเป็นผลสำเร็จ ซึ่งทางคณะผู้จัดทำต้องขอขอบพระคุณ ท่านบูรพาจารย์ทั้งหลาย ผู้เขียนเอกสารและตำราอ้างอิงต่าง ๆ โดยเฉพาะอาจารย์ที่ปรึกษาโครงการ ศศ. สุรพล บุญจันทร์ เป็นอย่างสูง ที่ช่วยถ่ายทอดความรู้ และมอบโอกาสที่ดีในการพัฒนาปริญญานิพนธ์ รวมถึงคำแนะนำที่เป็นประโยชน์

ขอขอบคุณ อาจารย์ทุกท่านที่ไม่ได้กล่าวถึงในที่นี้ ที่กรุณาประสิทธิประสาทวิชา ความรู้ รวมถึงแนวทางในการคิด และ แนะนำปฏิบัติแก่คณะผู้จัดทำจนทำให้ปริญญานิพนธ์ฉบับนี้สำเร็จ ผลตามเป้าหมาย

สุดท้ายนี้ทางคณะผู้จัดทำขอให้คุณความดีใด ๆ ที่เกิดจาก ปริญญานิพนธ์ฉบับนี้มอบให้แก่ บิดา มารดา ผู้ให้ความอนุเคราะห์แก่คณะผู้จัดทำในทุก ๆ ด้าน ครูบาอาจารย์ทุกท่านที่ได้มอบวิชา ความรู้ให้ รวมถึงเพื่อน ๆ ที่ทั้งให้คำปรึกษาและคำแนะนำที่ทำให้คณะผู้จัดทำมีกำลังใจทำปริญญานิพนธ์นี้ให้สำเร็จตามเป้าหมาย

นายพลกร พุ่มเฟื้อย  
นายเสถียรพงศ์ ใจเย็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ

	หน้า
บทคัดย่อ.....	ก
สารบัญ.....	ข
สารบัญรูป.....	ง
สารบัญตาราง.....	ฉ
<b>บทที่ 1 บทนำ.....</b>	<b>1</b>
1.1 ที่มาและความสำคัญของ โครงการงาน .....	1
1.2 วัตถุประสงค์.....	1
1.3 ขอบเขตของโครงการงาน.....	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	2
<b>บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง.....</b>	<b>3</b>
2.1 General Packet Radio Service.....	3
2.2 AT-COMMAND.....	4
2.3 การสื่อสารข้อมูลแบบอนุกรม.....	6
2.3.1 พื้นฐานการสื่อสารอนุกรม.....	6
2.3.2 การต่อ MCS-51 กับ RS-232.....	10
2.4 อีเมล (E-Mail) และวิธีการใช้งานในเบื้องต้น.....	11
2.4.1 โครงสร้างและรูปแบบของชื่ออีเมล.....	11
2.4.2 ประเภทต่าง ๆ ของอีเมล ที่มีเปิดให้บริการ.....	11
2.4.3 โปรโตคอลในการส่งและรับเมล.....	12
2.5 การตรวจสอบคำสั่งการใช้งาน POP3 โดยใช้ โปรแกรม telnet.....	13
2.5.1 การเชื่อมต่อ .....	13
2.5.2 การ Login เข้าสู่ mail server.....	13
2.5.3 ทดสอบการใช้คำสั่งต่าง ๆ ที่ใช้ควบคุมการทำงานของ POP3.....	14
2.6 การออกแบบฐานข้อมูล.....	16
2.6.1 ระบบฐานข้อมูล (Database System).....	16
2.6.2 การออกแบบฐานข้อมูลด้วยอี-อาร์โมเดล.....	16
2.6.3 โครงสร้างของภาษาเอสคิวแอล (SQL).....	17
2.6.4 ลักษณะการใช้งานของภาษา SQL.....	19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้เผยแพร่หรือการนำ  
 ไปทำกำไรในทางอื่นใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

2.6.5 การบันทึกข้อมูล, การปรับปรุงข้อมูล และการลบข้อมูล.....	20
2.6.6 การเรียกค้นข้อมูล (SELECT).....	22
<b>บทที่ 3 การออกแบบและการสร้าง.....</b>	<b>24</b>
3.1 การเชื่อมต่อโทรศัพท์มือถือ กับ Computer.....	24
3.1.1 ขั้นตอนการทดสอบ .....	25
3.2 การทำงานของผู้ขายของและเครื่องส่ง.....	26
3.3 การออกแบบฐานข้อมูล.....	29
3.3 การสร้างโปรแกรมที่ใช้ในการอ่าน E-Mail .....	33
3.3.1 ทำการออกแบบฐานข้อมูลโดยการสร้าง ER Diagram.....	30
3.3.2 สร้างฐานข้อมูล โดยใช้ MS Access 2000.....	32
3.4 การสร้างโปรแกรมที่ใช้ในการอ่าน E-Mail และจัดการฐานข้อมูล.....	33
3.4.1 โครงสร้างของโปรแกรม เก็บข้อมูลเครื่องขายของอัตโนมัติ.....	34
<b>บทที่ 4 การทดลองและผลการทดลอง.....</b>	<b>36</b>
4.1 วงจรควบคุมการส่ง E-mail.....	36
4.2 การทดสอบทำงานของโปรแกรมเก็บข้อมูลเครื่องขายของอัตโนมัติโดยใช้ GPRS.....	37
<b>บทที่ 5 สรุป และ วิจัย.....</b>	<b>40</b>
5.1 สรุปผลการดำเนินงาน.....	40
5.2 ปัญหาที่พบและแนวทางการแก้ไข.....	41
5.3 การพัฒนาต่อ.....	41
ภาคผนวก	
กิตติกรรมประกาศ	
เอกสารอ้างอิง	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

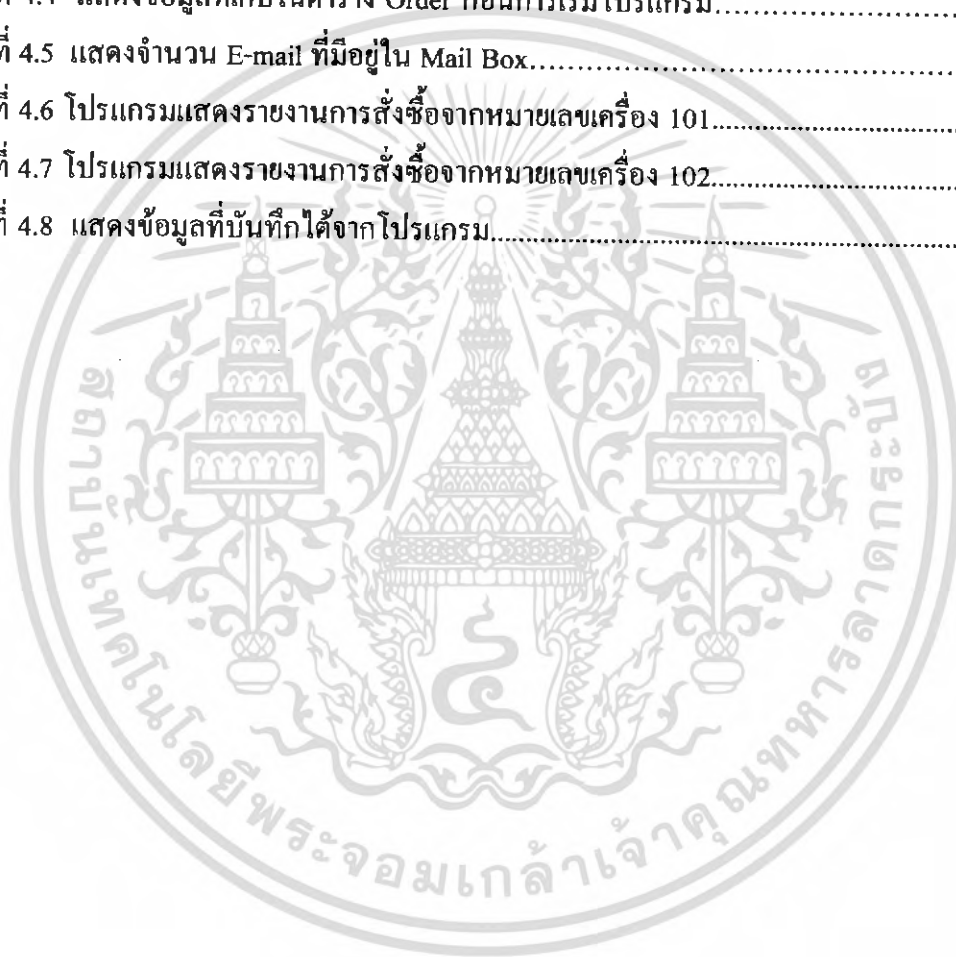
## สารบัญรูป

รูป	หน้า
รูปที่ 2.1 ตัวอย่างคำสั่งพื้นฐานและ LIST ข้อความใน STO SENT.....	5
รูปที่ 2.2 ตัวอย่างการอ่านข้อความ.....	5
รูปที่ 2.3 ลักษณะการรับส่งข้อมูลแบบอนุกรมแบบขนาน.....	6
รูปที่ 2.4 การรับส่งข้อมูลแบบ Simplex, Half duplex และ Full duplex.....	7
รูปที่ 2.5 เฟรมรหัส ASCII ของตัว "A".....	8
รูปที่ 2.6 ขั้วต่อแบบ DB-9 และหน้าที่ของขาต่างๆ.....	9
รูปที่ 2.7 โครงสร้างภายใน MAX232 และการเชื่อมต่อกับ MCS-51.....	10
รูปที่ 2.8 แสดงผลตอบสนองการเชื่อมต่อ POP3 Server.....	13
รูปที่ 2.9 แสดงการ Login เข้าสู่ระบบ mail server.....	13
รูปที่ 2.10 แสดงผลการส่งคำสั่ง STAT.....	14
รูปที่ 2.11 แสดงผลการส่งคำสั่ง LIST.....	14
รูปที่ 2.12 แสดงการใช้คำสั่ง RETR.....	15
รูปที่ 2.13 แสดงผลการใช้คำสั่ง DELE และคำสั่ง RSET.....	15
รูปที่ 3.1 The Connection Diagram.....	24
รูปที่ 3.2 โทรศัพท์มือถือ Ericsson รุ่น T65 และสาย Data Link.....	24
รูปที่ 3.3 Dialog Box Port Setting จากโปรแกรม Hyper Terminal.....	25
รูปที่ 3.4 คำสั่งในการเชื่อมต่อ TCP/IP Stack Chip และการส่ง E-Mail.....	26
รูปที่ 3.5 วงจรควบคุมและวงจรจับมอเตอร์.....	27
รูปที่ 3.6 ผังการทำงานเครื่องขายของอัตโนมัติ.....	28
รูปที่ 3.7 ตู้ขายของและวงจรควบคุมการทำงาน.....	29
รูปที่ 3.8 แสดง Entity ของฐานข้อมูล.....	30
รูปที่ 3.9 แสดงการเชื่อมความสัมพันธ์ ระหว่าง Entity สินค้า และ Entity เครื่อง.....	30
รูปที่ 3.10 แสดงการกำหนด Attribute ของ Entity.....	30
รูปที่ 3.11 แสดง Attribute ของ Relationship.....	31
รูปที่ 3.12 แสดงตาราง Order, Product และ ตาราง Machine ในฐานข้อมูล DatabasePRO..	32
รูปที่ 3.13 แสดงการเชื่อมความสัมพันธ์ระหว่างตารางโดยใช้ MS Access 2003.....	32
รูปที่ 3.14 ผังงานของโปรแกรมอ่านอีเมล และจัดการฐานข้อมูล.....	33
รูปที่ 3.15 หน้าต่าง Interface ในส่วนของ การเชื่อมต่อ POP3 และการอ่าน E-mail.....	34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป(ต่อ)

รูปที่ 3.16 หน้าต่าง Interface ในส่วนของ การจัดการฐานข้อมูล.....	35
รูปที่ 4.1 ช่องรับเหรียญ และสวิตช์เลือกสินค้า.....	36
รูปที่ 4.2 ช่องรับสินค้าเมื่อทำการกดสวิตช์.....	36
รูปที่ 4.3 ทำการส่ง E-mail ออกไปให้กับ โปรแกรมเก็บข้อมูล.....	37
รูปที่ 4.4 แสดงข้อมูลที่เก็บในตาราง Order ก่อนการเริ่ม โปรแกรม.....	37
รูปที่ 4.5 แสดงจำนวน E-mail ที่มีอยู่ใน Mail Box.....	38
รูปที่ 4.6 โปรแกรมแสดงรายงานการสั่งซื้อจากหมายเลขเครื่อง 101.....	38
รูปที่ 4.7 โปรแกรมแสดงรายงานการสั่งซื้อจากหมายเลขเครื่อง 102.....	39
รูปที่ 4.8 แสดงข้อมูลที่บันทึกได้จากโปรแกรม.....	39



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญตาราง

ตาราง	หน้า
ตารางที่ 3.1 แสดงการเปลี่ยน Entity สินค้า เป็นตารางสินค้า.....	31
ตารางที่ 3.2 แสดงการเปลี่ยน Entity เครื่องขายของ เป็นตารางเครื่องขายของ.....	31
ตารางที่ 3.3 แสดงการเปลี่ยน Relationship เป็นตารางการสั่งซื้อ.....	31



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1

## บทนำ

### 1.1 ที่มาและความสำคัญของโครงการ

เนื่องจากปัจจุบันมีการใช้งานอุปกรณ์ต่างๆ ที่ใช้ในชีวิตประจำวันมักมีระบบอัตโนมัติเพื่ออำนวยความสะดวกแก่ผู้บริโภค เครื่องขายของอัตโนมัติก็เป็นสิ่งหนึ่งที่สร้างความสะดวกสบายแก่ผู้บริโภค แต่เครื่องขายของอัตโนมัติที่มีอยู่ในปัจจุบันยังมีข้อเสียอยู่ตรงที่ เมื่อสินค้าในเครื่องขายของอัตโนมัติหมด ทางผู้ประกอบการจะไม่สามารถทราบสิ่งที่เกิดขึ้น และไม่สามารถทำการแก้ไขต่างๆ ได้ในทันที ทำให้เกิดผลกระทบต่อการค้าเงินธุรกิจประเภทนี้มาก

ด้วยเหตุผลข้างต้นทำให้ผู้จัดทำมีความสนใจในการจัดทำโครงการซึ่งมีเนื้อหาเกี่ยวกับการทำระบบแจ้งเตือนเมื่อสินค้าในตู้หมดของเครื่องขายของอัตโนมัติ โดยที่เครื่องขายของอัตโนมัติ จะทำการติดตั้ง Micro switch เพื่อทำการตรวจว่าสินค้าใกล้จะหมด เมื่อพบว่าสินค้าหมด ระบบจะส่งให้โทรศัพท์ไร้สายซึ่งการควบคุมโดยชุดคำสั่งของ AT - Command ทำการส่งข้อมูลผ่านระบบ GPRS เข้าสู่ Web mail เพื่อจะแจ้งผู้ประกอบการให้ทราบว่าตำแหน่งของเครื่องอยู่ที่ไหนและสินค้าประเภทใดหมด โดยที่เครื่องฝั่งรับซึ่งในที่นี้ใช้เป็นเครื่องคอมพิวเตอร์ที่ได้ทำการติดตั้งโปรแกรมที่ใช้ในการตรวจเช็คและทำการเปิดอ่าน Mail โดยจะทำการ Refresh ตามเวลาที่ตั้งเอาไว้ ซึ่งข้อมูลที่เปิดอ่านจะถูกนำมาเก็บไว้ในลักษณะของฐานข้อมูล

### 1.2 วัตถุประสงค์

- 1.2.1 เพื่อศึกษาการใช้งานและนำไปประยุกต์ใช้งาน MCS - 51 ได้จริง
- 1.2.2 เพื่อศึกษาการทำงาน Protocol SMTP ของ Mobile ในการส่ง E - mail
- 1.2.3 เพื่อศึกษาการทำงาน Protocol POP3 ในการรับ E - mail
- 1.2.4 เพื่อนำข้อมูลที่ศึกษาและวงจร Module ที่มีประยุกต์ใช้เป็นตัวส่งข้อมูลได้
- 1.2.5 เพื่อทำการเก็บข้อมูลจากเครื่องขายของอัตโนมัติ ซึ่งจะสร้างความสะดวกแก่ผู้ประกอบการ ให้สามารถเรียกดูข้อมูลการจำหน่ายสินค้า เพื่อการประเมินแผนการขายในอนาคตได้

### 1.3 ขอบเขตของโครงการงาน

- 1.3.1 สามารถส่ง E-Mail ด้วยมือถือโดยผ่าน TCP/IP Stack Chip
- 1.3.2 สามารถใช้ MCS – 51 เป็นตัว Control TCP/IP Stack Chip
- 1.3.3 สามารถสร้างโปรแกรมเพื่ออ่าน E-mail จาก Server POP3 และติดต่อฐานข้อมูลได้
- 1.3.4 สามารถทำการเก็บข้อมูลการขายสินค้าลงในฐานข้อมูล และเรียกดูข้อมูลที่เก็บไว้ในฐานข้อมูลได้

### 1.4 ประโยชน์ที่คาดว่าจะได้รับ

- 1.4.1 มีความรู้ MCS – 51 เพิ่มขึ้น และประยุกต์ใช้ได้จริง
- 1.4.2 มีความรู้เกี่ยวกับการทำงาน Protocol SMTP ของ Mobile ในการส่ง E-mail
- 1.4.3 มีความรู้เกี่ยวกับระบบการทำงานของ Protocol POP3 ในการรับ E-mail
- 1.4.4 สร้างตัวส่งข้อมูลจากการศึกษาวงจร Module ที่มีได้
- 1.4.5 มีความรู้ในการออกแบบฐานข้อมูล และสามารถใช้งานฐานข้อมูลที่ออกแบบให้เกิดประโยชน์ได้สูงสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

### ทฤษฎีและหลักการที่เกี่ยวข้อง

#### 2.1 General Packet Radio Service

GPRS (General Packet Radio Service) บริการต่างๆที่ผ่านทาง Radio Interface ในระหว่าง ผู้ใช้ต้นทางและปลายทางซึ่งไม่ว่าจะเป็น Application Server หรือแม้แต่ตัวโทรศัพท์เคลื่อนที่เองก็ตามจะถูกแปลงเป็น Packet ซึ่งมี IP Address กำกับอยู่ภายใน ซึ่งจะไม่เหมือนเดิมที่เคยใช้กัน (เดิมที่ใช้กันคือระบบ Radio Frame ที่ใช้กันในการส่งข้อมูลเสียงพูดบนระบบ GSM)

GPRS ไม่ได้เป็นลักษณะที่จะสามารถให้บริการได้ด้วยตัวของระบบเอง แต่ตัวมันเองเป็นเพียงแค่ Bearer ให้กับ Application ต่างๆ ที่ต้องการใช้ความเร็วที่เพิ่มมากกว่าปกติในระบบ GSM ที่เคยรองรับอยู่เดิมมาก่อน และระบบ GPRS จะต้องต่อไปยัง Packet Data Network ที่เป็น IP Network อีกต่อหนึ่ง

ดังนั้นผู้ให้บริการเครือข่ายโทรศัพท์เคลื่อนที่ ที่จะเปิดให้ใช้ในระบบ GPRS ได้นั้นจะต้องทำการติดตั้งระบบเครือข่าย ที่ประกอบด้วยหน่วยหลักๆ 2 หน่วยด้วยกันคือ

1. SGSN ( Serving GPRS Supports Node)
2. GGSN ( Gateway GPRS Support Node)

โดยทั้งสองหน่วยหลักขององค์ประกอบนี้จะถูกเชื่อมต่อเข้าด้วยกัน โดยมีอุปกรณ์อื่นๆ ที่เป็น ตัวช่วยเพื่อไปร่วมใช้ Radio Interface จาก Base Station โดยผ่านตัวควบคุมที่เรียกว่า PCU (Packet Control Unit) ที่ติดตั้งไว้ที่ BSC (Base Station Controller) ทั้งนี้อาจมองได้ว่า GPRS Network เป็นอีก Network หนึ่ง ซึ่งเข้าถึง Mobile Phone ผ่านทาง Radio Interface ของระบบ GSM Network เดิม โดยเป็นบริการที่เกี่ยวข้องเนื่องกับการรับส่งข้อมูลเป็น Packet โดยตรง

ตามทฤษฎีแล้ว GPRS สามารถให้บริการที่ความเร็วสูงสุดถึง 171.2 kbps โดยต้องอาศัยการใช้ช่วงเวลา (timeslot) ทั้งแปดช่วงของทั้งหมดที่มี ซึ่งนั่นหมายถึงความเร็วสูงสุดที่สูงขึ้นถึงสามเท่าของการส่งข้อมูลผ่านสาย บนเครือข่ายโทรศัพท์ปัจจุบัน และสูงขึ้นมากกว่าการเชื่อมต่อแบบ CSD ในเครือข่าย GSM ถึงสิบเท่า

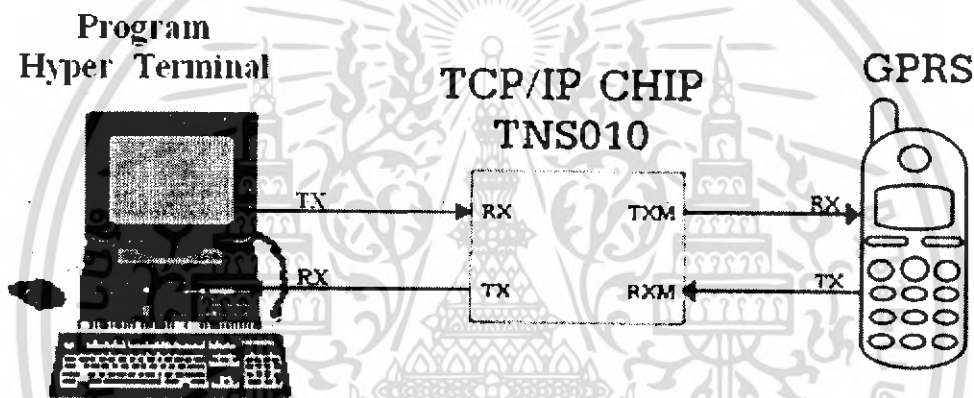
GPRS ยังรองรับการให้บริการในรูปแบบใหม่ที่ไม่สามารถให้บริการได้บนเครือข่าย GSM เดิมเพราะข้อจำกัดด้านความเร็วในการรับส่งข้อมูลในแบบ CSD (9.6 kbps) และข้อจำกัดของขนาดของข้อมูลที่สามารถรับส่งได้ในแบบ SMS (160 ตัวอักษร) GPRS ทำให้สามารถให้บริการในรูปแบบต่างๆ ที่ไม่เคยมีมาก่อนบนเครือข่ายโทรศัพท์เคลื่อนที่

### บทที่ 3

#### การออกแบบและการสร้าง

##### 3.1 การเชื่อมต่อโทรศัพท์มือถือ กับ Computer

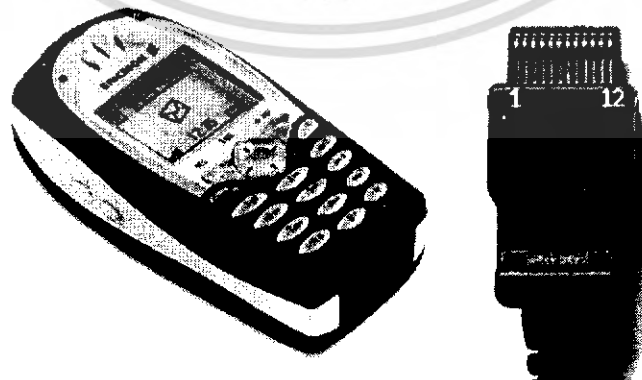
ในการเริ่มต้นขั้นตอนแรกจะ ทำการติดต่อให้กับ TCP/IP Stack Chip กับ Computer โดยใช้โปรแกรม HyperTerminal ส่งชุดคำสั่ง AT COMMAND ผ่าน Serial Port เพื่อสั่งให้ TCP/IP Stack Chip ส่งมือถือส่ง E-Mail แล้วดูผลตอบสนองจาก E-mail ที่ได้รับ



รูปที่ 3.1 The Connection Diagram

อุปกรณ์ที่ใช้ : โทรศัพท์มือถือ, สาย Data link, Computer (PC)

ซอฟต์แวร์ที่ใช้ : Hyper Terminal



รูปที่ 3.2 โทรศัพท์มือถือ Ericsson รุ่น T65 และสาย Data Link

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักการทดสอบ : ใช้โปรแกรม Hyper Terminal ส่งผ่านชุดคำสั่ง AT COMMAND ไปยัง TCP/IP Stack Chip เพื่อสั่งให้ TCP/IP Stack Chip ส่งมือถือส่ง E-Mail

### 3.1.1 ขั้นตอนการทดสอบ

- เซตอัตราการรับและ ส่งข้อมูลระหว่าง TCP/IP Stack Chip กับ PC ให้ตรงกัน สำหรับ TCP/IP Stack Chip อัตราการ รับ-ส่งข้อมูลสามารถเซตได้คือ 19200 bps ส่วนอัตรารับส่งข้อมูลของ PC เซตได้จาก Port Setting ดังรูปที่ 3.3



รูปที่ 3.3 Dialog Box Port Setting จากโปรแกรม Hyper Terminal

- เริ่มทำการทดสอบโดยส่งชุดคำสั่ง AT COMMAND ไปยัง TCP/IP Stack Chip เริ่มต้นจากคำสั่ง at+i(enter) หากระบบสามารถเชื่อมต่อได้ TCP/IP Stack Chip ได้โทรศัพท์มือถือจะตอบกลับมาว่า +I\_OK จากนั้นจึงทดสอบคำสั่งพื้นฐานต่างๆ เพื่อสั่งการ TCP/IP Stack Chip ให้ทำการส่ง E-mail ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

tesl@project - HyperTerminal
File Edit View Call Transfer Help
+I_READY
at+i
+I_OK
at+idetect
57600
+I_OK
at+isetsd=*99***1#
+I_OK
at+iopen
+I_OK
at+ismtpsend=mail.cscoms.com,test@project.com,mdragon_voice@yahoo.co.th,test,project
+I_OK
-
Connected 0:06:33 Auto detect 19200 8-N-1 NRM

```

### รูปที่ 3.4 คำสั่งในการเชื่อมต่อ TCP/IP Stack Chip และการส่ง E-mail

at+i : เช็ควงจรความพร้อมระหว่าง TCP/IP Stack Chip กับ Hyper Terminal

at+idetect: เช็ควงจร baud rate ระหว่าง TCP/IP Stack Chip กับ มือถือ

at+isetsd = \*99\*\*\*1# : เลขหมาย dialing number

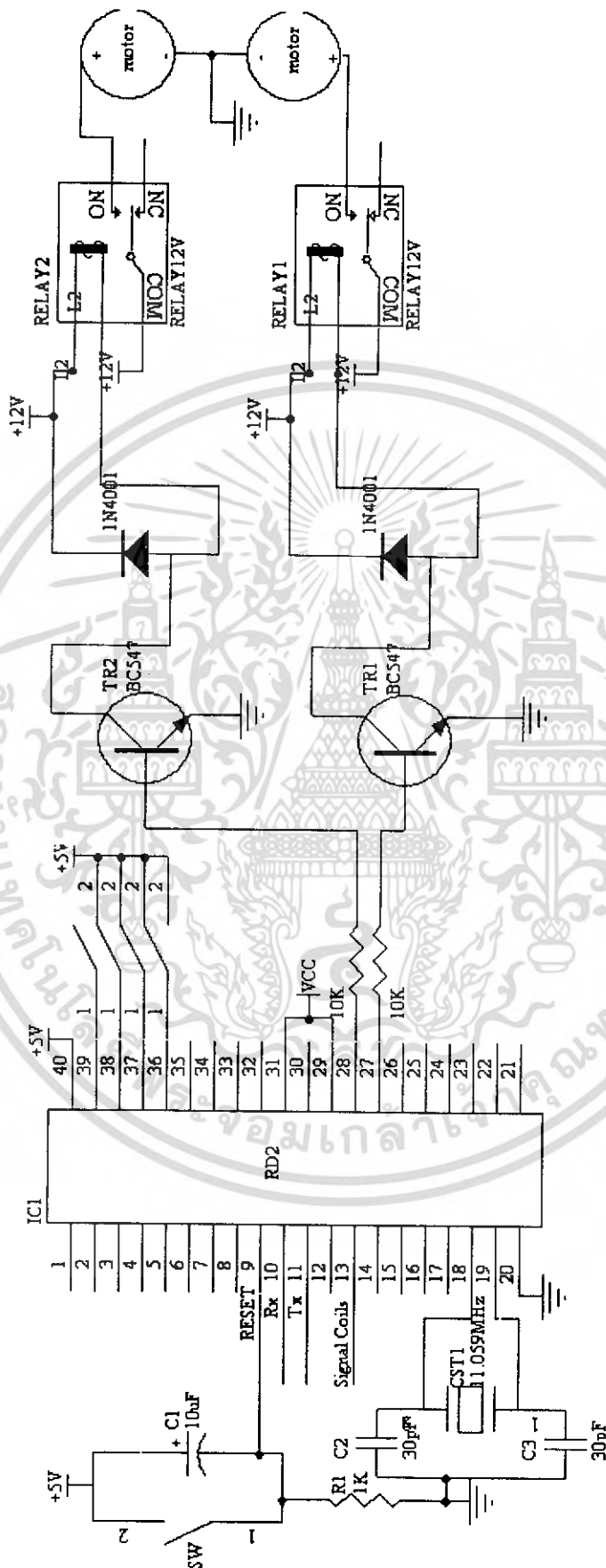
at+iopen: ทำการเปิด Protocol PPP

at+ismtpsend = mail server, mail sender, mail receiver, subject, body

เมื่อทำการใช้คำสั่งต่างๆ ดังในรูปแล้ว TCP/IP Stack Chip จะสั่งให้ มือถือส่ง E-mail ออกไป ทางด้าน mail ของผู้รับก็จะได้รับ mail ที่ได้ส่งออกไป

### 3.2 ส่วนของวงจร Microcontroller ภายในตู้สินค้าเพื่อควบคุมส่วนต่างๆ

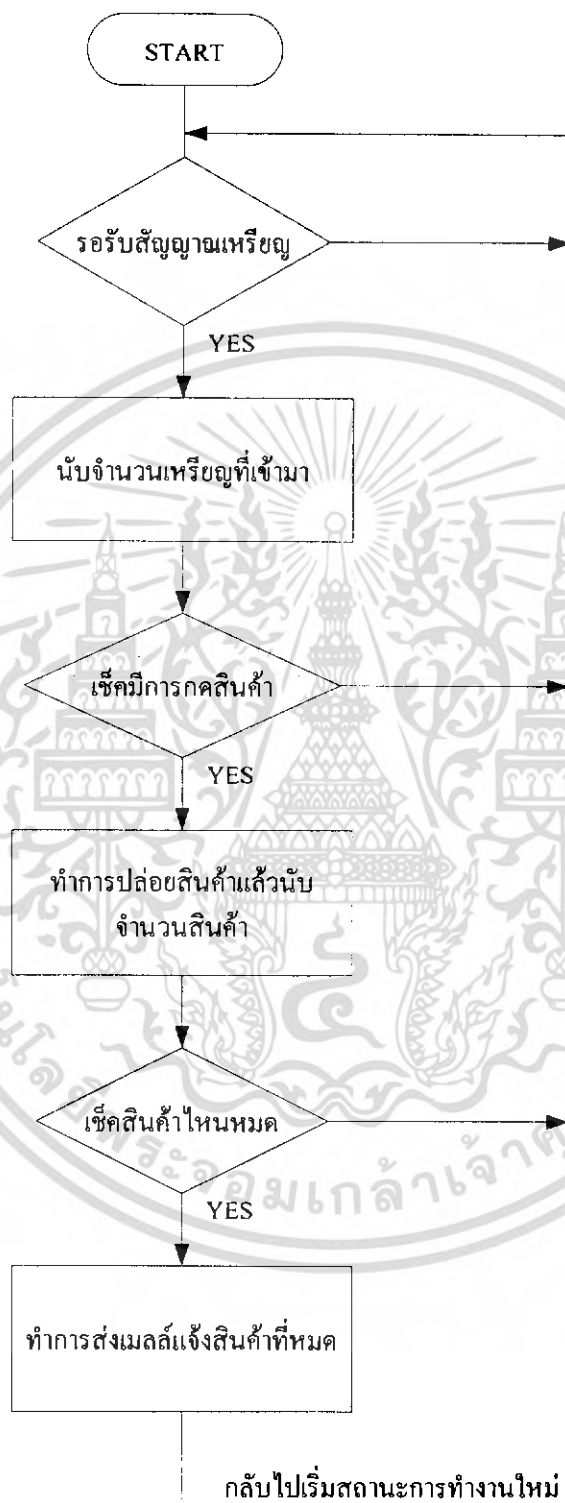
วงจร Microcontroller พื้นฐาน เพื่อใช้ในการเชื่อมต่อกับสัญญาณ Signal Coils, สวิตซ์ สิ้นค้าและ วงจรขับเคลื่อนภายในตู้ขายของอัตโนมัติ



รูปที่ 3.5 วงจรควบคุมและวางขั้วมอเตอร์

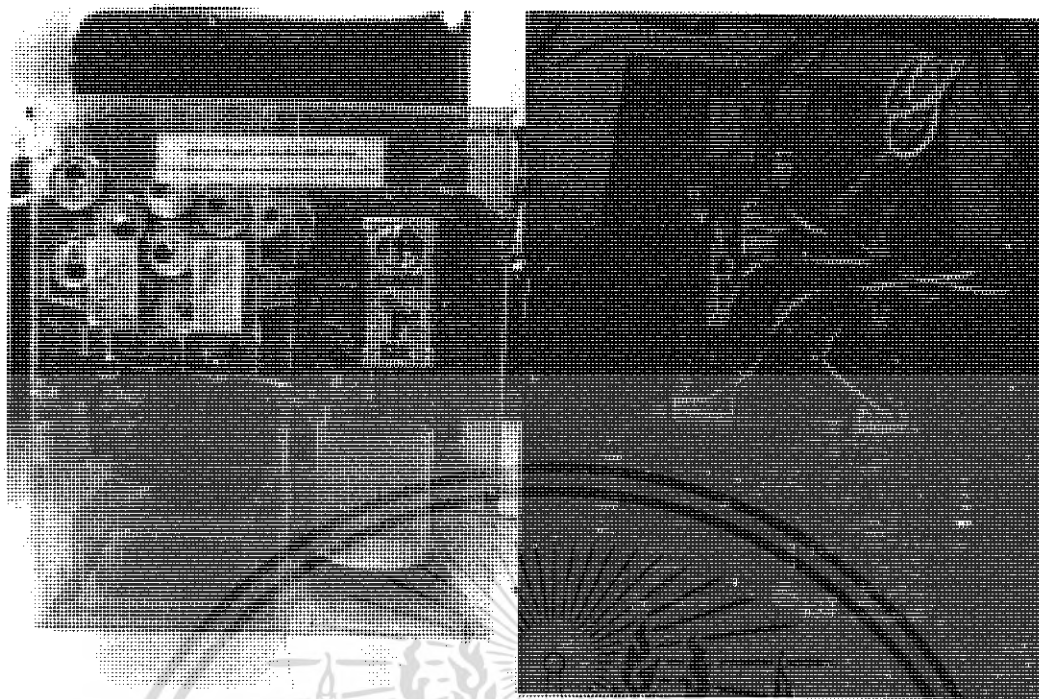
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## FlowChart การทำงานของตู้ขายของ



รูปที่ 3.6 ผังการทำงานเครื่องขายของอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 ตู้ขายของและวงจรควบคุมการทำงาน

#### หลักการทำงาน

1. ใใส่เหรียญเพื่อส่งสัญญาณให้ controller เก็บจำนวนเหรียญ
2. กดสินค้าที่ต้องการ ไมโครสวิทช์จะส่งวงจรขั้วมอเตอร์ หมุนเอาสินค้าออกมา
3. นับจำนวนสินค้าแล้วเซ็นเซอร์สินค้าว่าหมดหรือยัง หากหมดให้ controller ส่งคำสั่งให้กับ TCP/IP CHIP เพื่อส่ง E-mail ออกไป
4. กลับมาทำงานตามปกติแล้วเซ็นเซอร์สินค้าชนิดอื่นต่อไป

#### 3.3 การออกแบบฐานข้อมูล

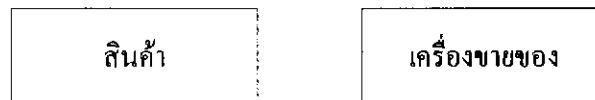
ออกแบบฐานข้อมูลเพื่อเก็บข้อมูลการขายสินค้าจากเครื่องขายของอัตโนมัติ โดยต้องการให้สามารถทำการเรียกดูข้อมูล ดังต่อไปนี้

- แสดง ชื่อสินค้า, จำนวนสินค้าที่ขายไป และ ที่อยู่ของเครื่องขายของอัตโนมัติ
- แสดงวันที่ ที่ทำการบันทึกข้อมูลลงในฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.1 ทำการออกแบบฐานข้อมูลโดยการสร้าง ER Diagram

- กำหนด Entity ของระบบ



รูปที่ 3.8 แสดง Entity ของฐานข้อมูล

- สร้างความสัมพันธ์ ระหว่าง Entity



รูปที่ 3.9 แสดงการเชื่อมความสัมพันธ์ ระหว่าง Entity สินค้า และ Entity เครื่อง

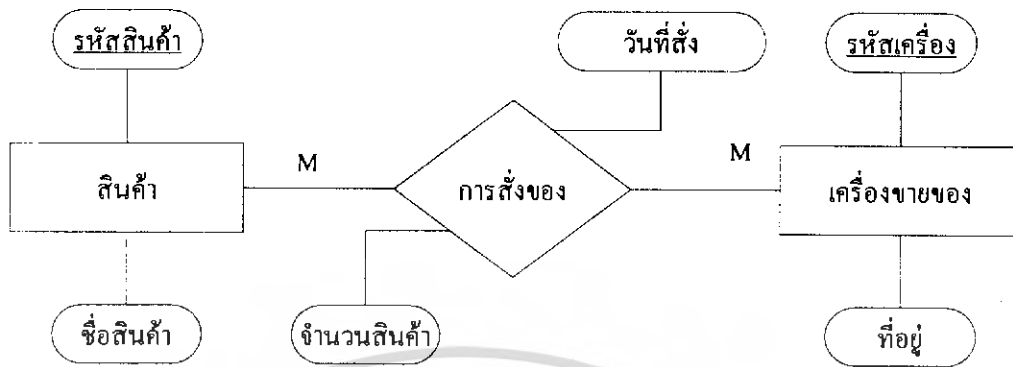
- กำหนด Attribute ของ Entity แต่ละตัว



รูปที่ 3.10 แสดงการกำหนด Attribute ของ Entity

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- กำหนด Attribute ให้กับ Relationship และ คีย์หลักของ Entity แต่ละตัว



รูปที่ 3.11 แสดง Attribute ของ Relationship

- เปลี่ยน ER Diagram เป็นตารางที่ต้องสร้าง ตารางที่ 1 และ 2 สร้างจาก Entity

ตาราง สินค้า

รหัสสินค้า	ชื่อสินค้า
------------	------------

ตารางที่ 3.1 แสดงการเปลี่ยน Entity สินค้า เป็นตาราง สินค้า

ตาราง เครื่องขายของ

รหัสเครื่องขายของ	ที่อยู่เครื่องขายของ
-------------------	----------------------

ตารางที่ 3.2 แสดงการเปลี่ยน Entity เครื่องขายของ เป็นตาราง เครื่องขายของ

ตาราง ที่ 3 สร้างจาก Relationship

ตาราง การสั่งซื้อ

รหัสเครื่องขายของ	รหัสสินค้า	จำนวนสินค้าที่สั่ง	วันที่สั่งสินค้า
-------------------	------------	--------------------	------------------

ตารางที่ 3.3 แสดงการเปลี่ยน Relationship เป็นตาราง การสั่งซื้อ

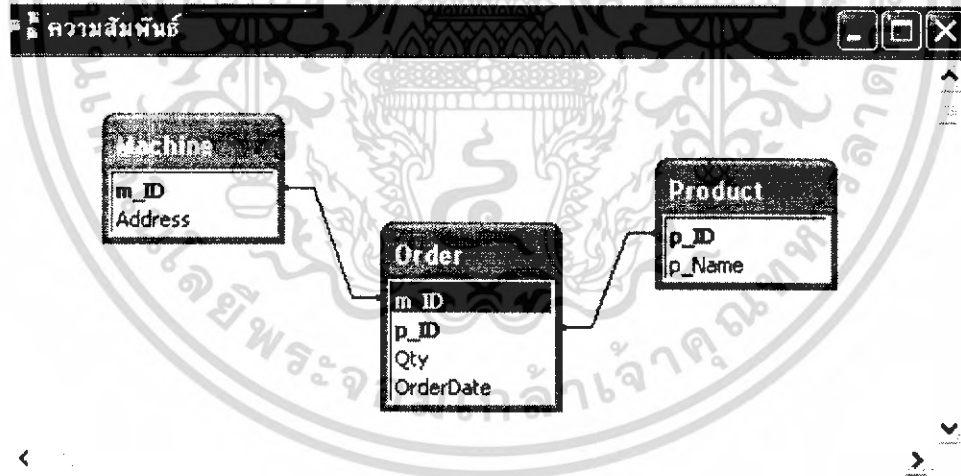
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 สร้างฐานข้อมูล โดยใช้ MS Access 2000 ชื่อว่า DatabasePRO โดยใช้ตารางที่ออกแบบจาก ER Diagram และทำการเชื่อมความสัมพันธ์ให้กับตารางทั้งหมดในฐานข้อมูล

Machine : ตาราง		Product : ตาราง	
ชื่อเขตข้อมูล	ชนิดข้อมูล	ชื่อเขตข้อมูล	ชนิดข้อมูล
m_ID	Text	p_ID	Text
Address	Text	p_Name	Text

Order : ตาราง	
ชื่อเขตข้อมูล	ชนิดข้อมูล
m_ID	Text
p_ID	Text
Qty	Number
OrderDate	Date/Time

รูปที่ 3.12 แสดงตาราง Order, Product และ ตาราง Machine ในฐานข้อมูล DatabasePRO

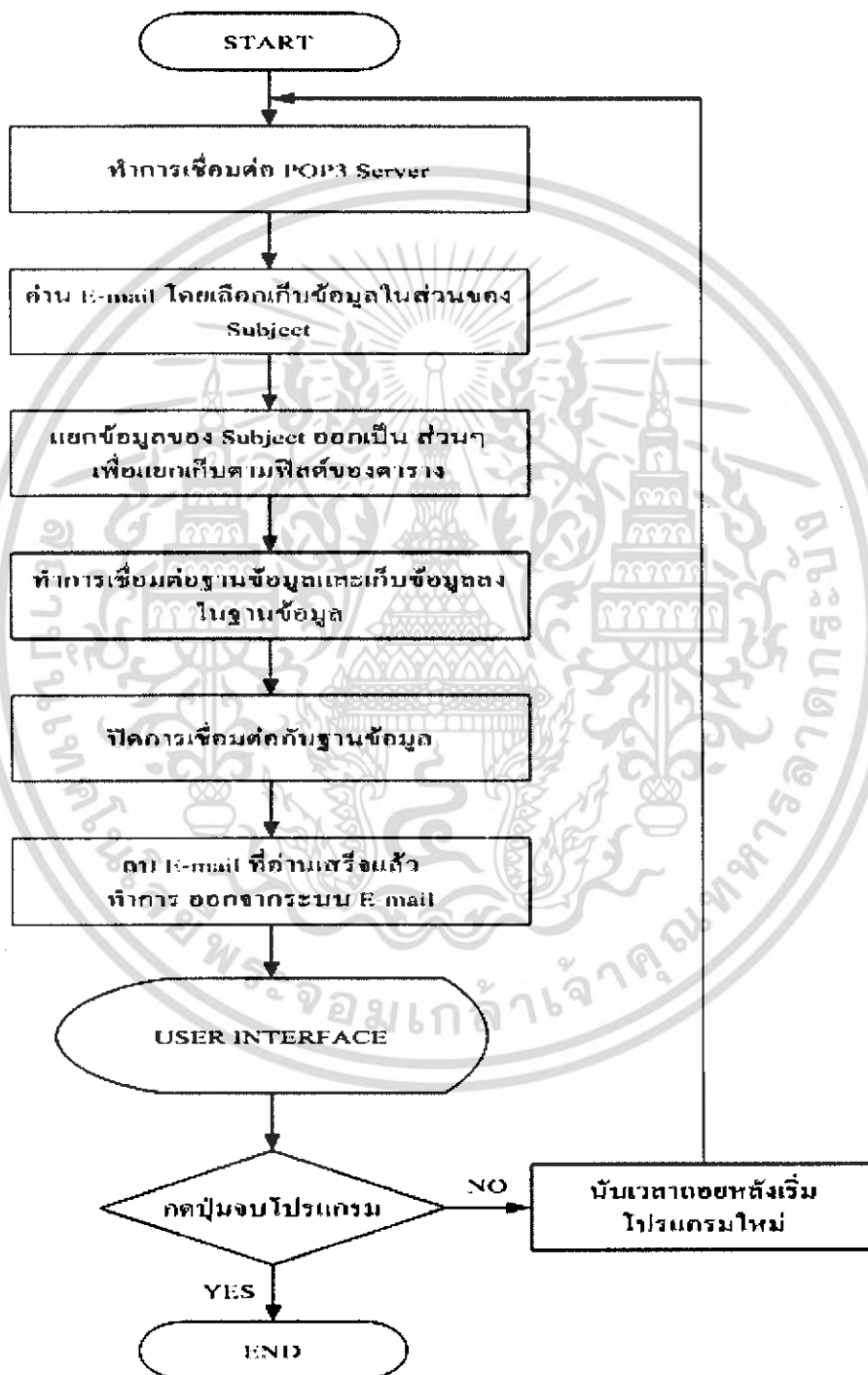


รูปที่ 3.13 แสดงการเชื่อมความสัมพันธ์ระหว่างตารางโดยใช้ MS Access 2003

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4 การสร้างโปรแกรมที่ใช้ในการอ่าน E-Mail และจัดการฐานข้อมูล

สร้างโปรแกรมที่ใช้ในการอ่าน E-Mail ที่เขียนโดย Visual C#.NET โดยการนำเอาคำสั่งต่าง ๆ ที่ใช้บน POP3 มาใช้ในการควบคุมการทำงานของโปรแกรม ซึ่งหลักในการเขียนโปรแกรมแสดงผังงาน (Flow Chart) ดังรูปที่ 3.14



รูปที่ 3.14 ผังงานของโปรแกรมอ่านอีเมลล์ และจัดการฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.1 โครงสร้างของโปรแกรม เก็บข้อมูลเครื่องขายของอัตโนมัติโดยใช้ GPRS จะแบ่งเป็นส่วนหลัก ๆ อยู่ทั้งหมด 2 ส่วนด้วยกันคือ

- โปรแกรมในส่วน การเชื่อมต่อ POP3 และอ่าน E-mail
- โปรแกรมในส่วนของการจัดการฐานข้อมูล

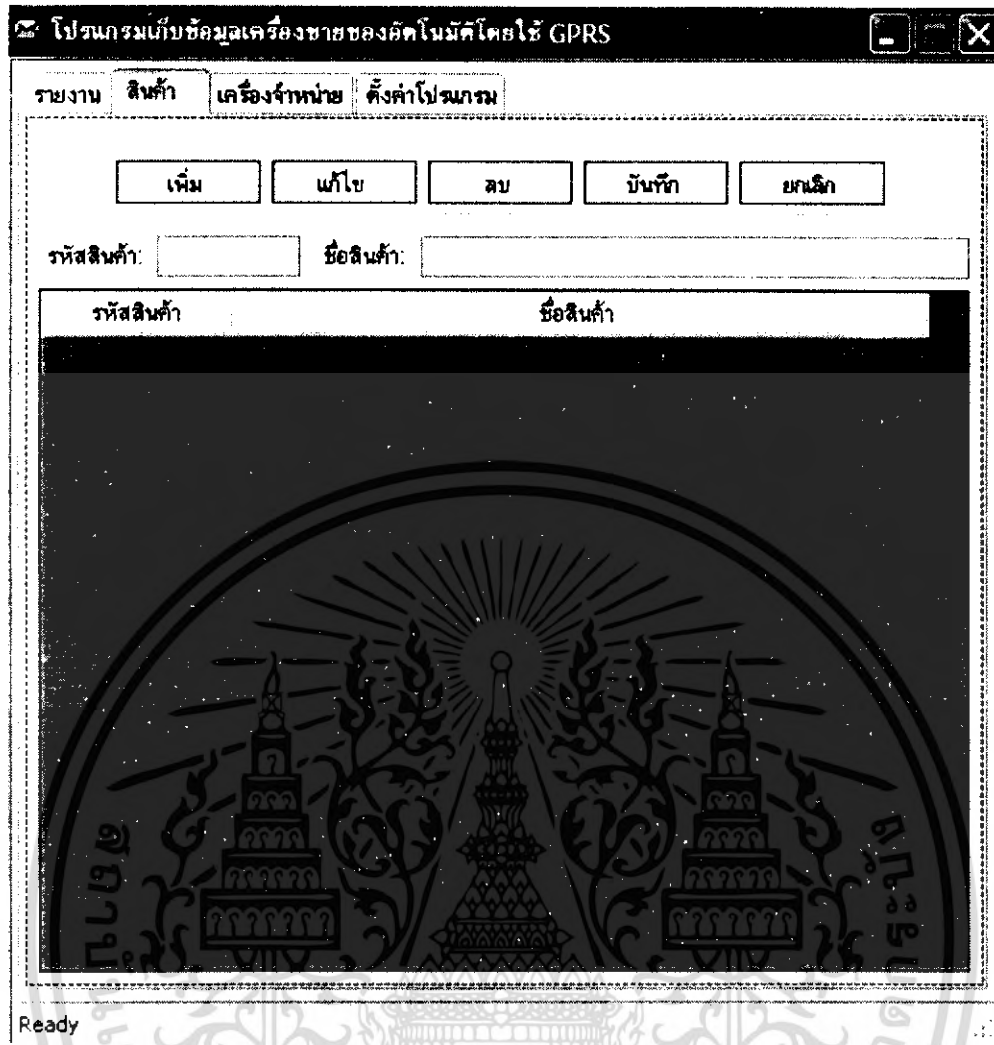
ส่วนของโปรแกรมสำหรับการเชื่อมต่อ POP3 และ การอ่าน E-mail สร้างหน้าต่าง Interface โดยให้ระบุการตั้งค่าของ POP3 Server, Port, Username, Password เพื่อใช้ในการเชื่อมต่อกับ POP3 Server และในส่วนของ Time interval จะใช้ในการตั้งค่า Timer ของโปรแกรม เพื่อ ตั้งเวลาให้ โปรแกรมเริ่มทำงานทุก ๆ กี่นาที

The screenshot shows a software window titled "โปรแกรมเก็บข้อมูลเครื่องขายของอัตโนมัติโดยใช้ GPRS". The window contains a form with the following fields and controls:

- POP3
- POP3 Server:
- Port:
- Username:
- Password:
- Time interval:
- Buttons: OK, Cancel
- Status bar: Ready

รูปที่ 3.15 หน้าต่าง Interface ในส่วนของ การเชื่อมต่อ POP3 และการอ่าน E-mail

ส่วนของโปรแกรมสำหรับการจัดการฐานข้อมูล ในการออกแบบให้สามารถทำการแสดง เป็น 3 ส่วนด้วยกัน คือ ฟอรัมรายงาน, ฟอรัมสินค้า และฟอรัมเครื่องจำหน่าย โดยในฟอรัมของ รายงานจะเรียกแสดง ข้อมูลจากฐานข้อมูล DatabasePRO ออกมาแสดงในรูปแบบของรายงาน ส่วนฟอรัมของสินค้า และ เครื่องจำหน่าย จะสามารถ เรียกแสดงข้อมูลจากตาราง สินค้า และ เครื่องขายของ ตามลำดับ โดยสามารถทำการเพิ่ม, แก้ไข และลบข้อมูลออกจากตารางดังกล่าวได้ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่บนสื่อใดๆ ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.16 หน้าต่าง Interface ในส่วนของการจัดการฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

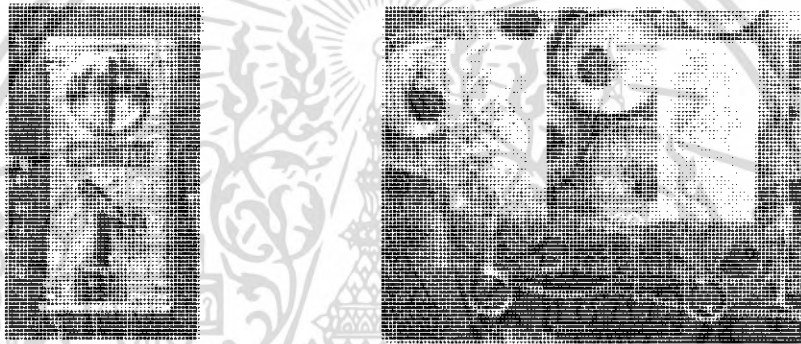
### การทดลองและผลการทดลอง

#### 4.1 การทดสอบใช้ผู้ขายของอัตโนมัติและ ส่งข้อมูลออกไปยังโปรแกรมเก็บข้อมูล

เป็นการทดสอบจำลองการทำงานของตู้ขายของอัตโนมัติทั่วไปและทำการเช็คสินค้าหมดหรือยังถ้าหมดให้ทำการส่ง E-mail ออกไป

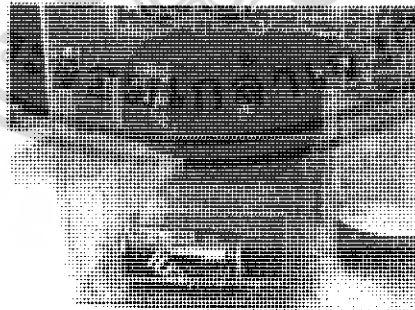
การทดสอบการทำงาน

ทำการใส่เหรียญลงในช่องหยอดเหรียญในที่นี้เราจะเช็คเฉพาะเหรียญ 10 บาทเท่านั้น



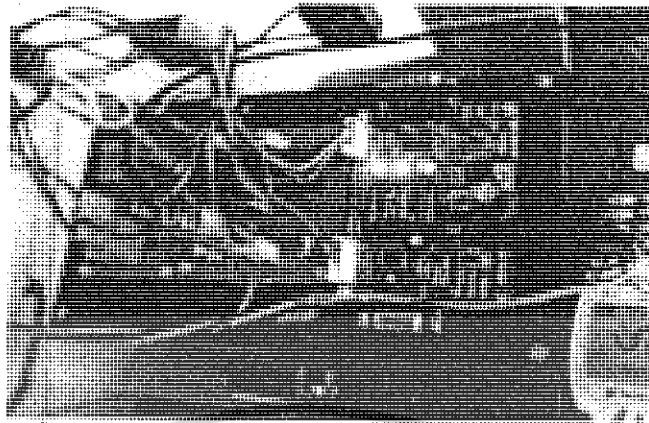
รูปที่ 4.1 ช่องรับเหรียญ และสวิตช์เลือกสินค้า

ทำการเลือกสินค้าที่ต้องการ โดยสวิตช์จะสั่งให้ Microcontroller ส่งสัญญาณให้วงจรรับทำงานส่งสินค้าที่เราต้องการออกมา



รูปที่ 4.2 ช่องรับสินค้าเมื่อทำการกดสวิตช์

โปรแกรมจะทำการนับสินค้าที่ออกไปเรื่อยๆ จนครบจำนวนที่โปรแกรมไว้ Microcontroller จะส่งคำสั่ง ออกทาง Serial Port ให้กับ TCP/IP Chip ให้ทำการส่ง E-mail ออกไป เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 ที่งานส่ง Email ออกไปให้เก็บโปรแกรมเป็นข้อมูล

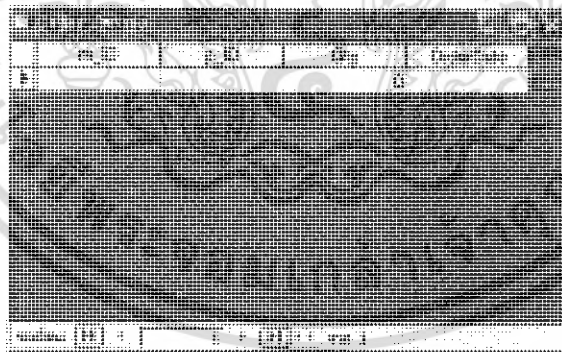
ทำการรับข้อมูลสินค้าและเช็คสินค้าชนิดอื่นว่าหมดหรือยังต่อไป

#### 4.2 การทดสอบทำงานของ โปรแกรมเก็บข้อมูลเครื่องขายของอัตโนมัติโดยใช้ GPRS

การทดสอบเพื่อให้ทราบว่าโปรแกรมมีการทำงานที่ถูกต้อง โดยทำการเปรียบเทียบข้อมูล  
ที่เก็บไว้ในฐานข้อมูลชื่อ DatabasePRO ก่อนที่จะมีการ RUN โปรแกรม

เงื่อนไขในการทดลอง

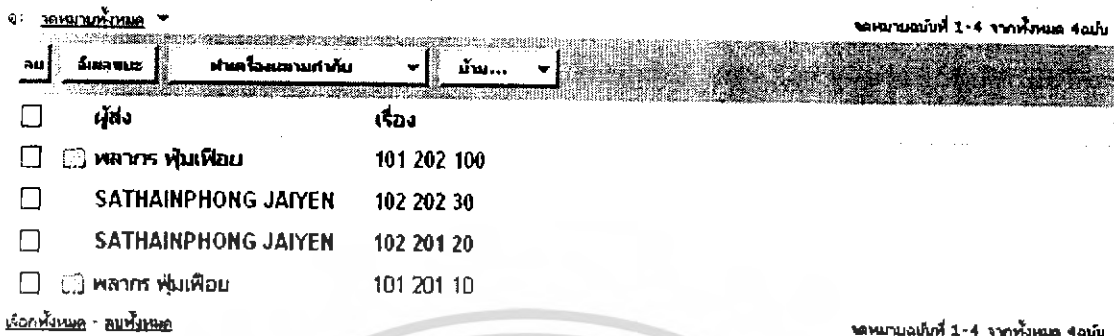
4.2.1 ให้ตอนเริ่มต้น ตาราง Order 'ไม่ได้เก็บข้อมูลใด ๆ เอาไว้เลย



รูปที่ 4.4 แสดงข้อมูลที่เก็บในตาราง Order ก่อนการเริ่มโปรแกรม

## 4.2.2 ตรวจสอบข้อมูลที่อยู่ใน E-mail ก่อนที่จะมีการทำงานของโปรแกรม

### กล่องรับจดหมาย

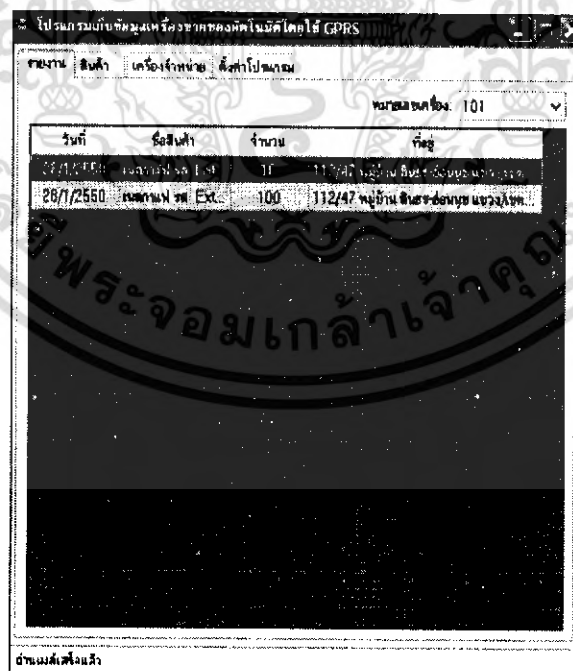


รูปที่ 4.5 แสดงจำนวน E-mail ที่มีอยู่ใน Mail Box

จากรูปจะเห็นได้ว่ามี E-mail อยู่ใน Mail Box ทั้งหมด 4 ฉบับด้วยกัน ซึ่งเมื่อโปรแกรมทำงาน จะทำการเก็บเอาส่วนของ หัวเรื่อง (Subject) มาทำการแยกข้อความแล้วเก็บไว้ในฐานข้อมูล

### 4.2.3 เปิดโปรแกรมเพื่อเริ่มการทำงาน

การทำงานของโปรแกรมจะเริ่มทำงานทันทีที่มีการ เปิดโปรแกรมขึ้นมา โดยจะทำการส่งค่าต่าง ๆ ที่ตั้งค่าเอาไว้แล้วก่อนหน้านี้ เพื่อทำการเชื่อมต่อกับ POP3 Server



รูปที่ 4.6 โปรแกรมแสดงรายงานการสั่งซื้อจากหมายเลขเครื่อง 101

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วันที่	สินค้า	จำนวน	ที่ตั้ง
28/1/2550	เซลล์เครื่อง 1st	20	27/11 อาคารไปรษณีย์กลางบางรัก
28/1/2550	เซลล์เครื่อง 2nd	30	87/110 อาคารไปรษณีย์กลางบางรัก

รูปที่ 4.7 โปรแกรมแสดงรายงานการสั่งซื้อจากหมายเลขเครื่อง 102

4.2.4 ทำการเปรียบเทียบผลระหว่างค่าที่แสดงในรายงานของโปรแกรมกับข้อมูลในตาราง Order ในฐานข้อมูล DatabasePRO

Order : ตาราง				
	m_ID	p_ID	Qty	OrderDate
	101	201	10	28/1/2550
	101	202	100	28/1/2550
	102	201	20	28/1/2550
	102	202	30	28/1/2550
			0	

รูปที่ 4.8 แสดงข้อมูลที่บันทึกได้จากโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### สรุป และ วิจารณ์

#### 5.1 สรุปผลการดำเนินงาน

5.1.1 ศึกษาการทำงานและคำสั่งต่าง ๆ เพื่อให้เข้าใจโครงสร้างของ โปรโตคอล SMTP และนำข้อมูลที่ได้มาใช้ในการเขียนโปรแกรมควบคุมการส่ง E-Mail

5.1.2 ศึกษาการใช้คำสั่งต่างๆ ของ AT COMMAND เพื่อให้เข้าใจและสามารถนำมาใช้ในการเขียนโปรแกรมควบคุมการส่ง E-Mail

5.1.3 ศึกษาการเขียนโปรแกรมควบคุม MCS - 51 เพื่อให้เข้าใจและ สามารถนำมาใช้ควบคุมในเครื่องส่งของอัตโนมัติและควบคุมการส่ง E-Mail

5.1.4 ศึกษาการทำงานและคำสั่งต่าง ๆ เพื่อให้เข้าใจโครงสร้างของ POP3 และนำข้อมูลที่ได้มาใช้ในการเขียนโปรแกรมควบคุมการอ่าน E-Mail

5.1.5 ศึกษาการเขียนโปรแกรมด้วย Visual C#.NET และทำการออกแบบโปรแกรม READ E-MAIL ขึ้นมาเพื่อใช้ในการ ติดต่อกับ POP3 Server โดยอาศัยโครงสร้างและคำสั่งของ POP3 ที่ได้ศึกษามาแล้วใช้

5.1.6 ทำการเขียนโปรแกรม READ E-MAIL ให้สามารถเชื่อมต่อ และทำการดาวน์โหลด (Download) ไฟล์ของ E-Mail ทางหน้าต่างของโปรแกรม รวมไปถึงการจัดการกับ E-Mail ที่อยู่ใน Inbox ของ USER นั้น ๆ เช่น การเรียกดู และการลบ E-mail ออกจาก Inbox เป็นต้น

5.1.7 โปรแกรมที่เขียนขึ้นเพื่อเปิดอ่าน E-mail แล้วทำการเก็บข้อมูลลงในฐานข้อมูลได้ และสนับสนุนให้ผู้ใช้สามารถทำการจัดการกับฐานข้อมูลได้ ซึ่งได้แก่

- เรียกดูข้อมูลการขายสินค้า (ระบุที่อยู่เครื่อง, ชื่อสินค้า และจำนวนที่ขายไป)
- สามารถเพิ่ม, แก้ไข ข้อมูลในส่วนของสินค้า
- สามารถเพิ่ม, แก้ไข ข้อมูลในส่วนของเครื่องขายของอัตโนมัติ
- สามารถทำการตั้งค่าการเชื่อมต่อ POP3 รวมถึงการตั้งเวลาในการอ่าน E-mail

## 5.2 ปัญหาที่พบและแนวทางการแก้ไข

5.2.1 ในส่วนโปรแกรม microcontroller มอเตอร์เวลาเริ่มหมุนจะเกิด noise ทำให้รบกวนการทำงานของ microcontroller โปรแกรมเกิดอาการรวน จึงใช้ตัวเก็บประจุต่อคร่อมลงไป

5.2.2 สัญญาณคลื่นระบบบางช่วงไม่สามารถเชื่อมต่อได้ ซึ่งถือว่าเป็นปัญหาทางเครือข่ายจะต้องแก้ไข โดยการส่งข้อมูลทำการติดต่อกออกไปเรื่อยๆ จนกว่าจะสามารถติดต่อกได้

5.2.3 สัญญาณมอเตอร์มีผลรบกวนบอร์ด TCP/IP Chip และ ไมโครคอนโทรลเลอร์ ทำให้โปรแกรมเกิดการไม่เสถียรภาพ

## 5.3 แนวทางการพัฒนาต่อ

5.3.1 ทำการแก้ไขโปรแกรมไมโครคอนโทรลเลอร์ในส่วนที่เป็น BUG บางส่วนเพื่อให้โปรแกรมทำงานได้สมบูรณ์

5.3.2 ทำการปรับแต่งในส่วนของมือถือให้สามารถส่งสัญญาณในตึกที่มีคิซิดได้

5.3.3 ทำการปรับแต่งตู้สินค้าให้เพิ่มจำนวนสินค้าให้มากขึ้น

5.3.4 เพิ่มความสามารถของโปรแกรม ให้สามารถเก็บรายละเอียดของระบบให้ครอบคลุมยิ่งขึ้น ตัวอย่างเช่นการเก็บรายละเอียดของ วันที่ และเวลาที่มีการส่งและบันทึกข้อมูลเนื่องจากจะเป็นประโยชน์อย่างยิ่งในการอ้างอิงถึงข้อมูลที่ถูกต้องยิ่งขึ้น และสามารถนำเอาวันที่จำหน่ายมาใช้เป็น เงื่อนไขในการเรียกดูรายการขายเป็นรายสัปดาห์ รายเดือน หรือช่วงเวลาที่กำหนดได้

5.3.5 เพิ่มความสามารถของโปรแกรมให้สามารถทำการคำนวณเพื่อ ทำรายการสั่งซื้อ และพิมพ์รายการการสั่งซื้อสินค้าออกมาทางเครื่องพิมพ์ได้

## เอกสารอ้างอิง

- [1] จิตรศักดิ์ เหลืองอุไร (เรียบเรียง), Peter W. Gofton, **ตำมัลการใช้งานการสื่อสารอนุกรมบนPC Mastering Serial Communication**. 2538.
- [2] ชีรวัฒน์ ประกอบผล. **ภาษาแอสเซมบลี สำหรับ MCS-51**. กรุงเทพฯ: สมาคมส่งเสริมเทคโนโลยี, 2546. หน้า 119-225
- [3] นิรุช อำนวยศิลป์. **เขียนโปรแกรมบนวินโดวส์ ด้วย Visual C++ และ MFC**. กรุงเทพฯ: ค่านสุทธากาพิมพ์, 2548. 264 หน้า
- [4] นิรุช อำนวยศิลป์. **Network and Protocol Programming using C/C++**. กรุงเทพฯ: จ.เจริญการพิมพ์ จำกัด, 2548. 210 หน้า
- [5] นิรุช อำนวยศิลป์. **สร้างโปรแกรมบน Windows ด้วย Microsoft Visual C++ Visual 6.0**. พิมพ์ครั้งที่ 10, กรุงเทพฯ: ชัคเชส มีเดีย, 2549. 398 หน้า
- [6] มณีโชติ สมานไทย. **คู่มือการออกแบบฐานข้อมูลและภาษา SQL ฉบับผู้เริ่มต้น**. นนทบุรี: อินโฟเพรส, 2546. 304 หน้า
- [7] สุภชัย สมพานิช. **คู่มือการเขียนโปรแกรมและการใช้งาน Visual C#.NET ฉบับสมบูรณ์**. นนทบุรี: อินโฟเพรส, 2546. 575 หน้า
- [8] สมยศ จุณณะปิยะ. **การประยุกต์ใช้งานไมโครคอนโทรลเลอร์ ตระกูล MCS-51**. พิมพ์ครั้งที่ 5, กรุงเทพฯ: สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2546.
- [9] SUPPORT GUIDE FOR THE NOKIA PHONES AND AT-COMMAND. Nokia co.Ltd.  
Retrieved June 11, 2004, [www.nokia.com](http://www.nokia.com)
- [10] [www.thaicyberu.go.th/OfficialTCU/main/2543\\_09\\_DatabaseSystem/public\\_html/index.htm](http://www.thaicyberu.go.th/OfficialTCU/main/2543_09_DatabaseSystem/public_html/index.htm)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## โปรแกรมภาษาแอสเซมบลีใน Microcontroller เพื่อควบคุมสัญญาณและติดต่อกับ TCP/IP Chip

```
ORG 0000H
LJMP MAIN

ORG 0003H          ; รอรับค่าจาก INT0
RETI

ORG 0013H          ; รอรับค่าจาก INT1

INC B              ; Signal Coil
RETI

ORG 100H
MAIN: MOV PCON, #80H      ; SMOD = 1
      MOV TMOD, #20H     ; Timer 1 Mode 2
      MOV TH1, #0FDH     ; 19,200 Baud
      MOV SCON, #50H     ; Serial Mode 1 REN = 1
      SETB TR1           ; Start Timer
      MOV IE, #85H       ; Enable INT0, INT1
      MOV P0, #00H
      MOV R0, #00H
      MOV R1, #00H
      MOV R2, #00H
      MOV P2, #00H
      ACALL CN
      ACALL S_AT_I
      ACALL S_AT_SD
      ACALL S_AT_DT
      ACALL CHECK
      RET
```

### ; โปรแกรมย่อยเช็คการกดสวิทซ์สัญญาณ

```
CHECK: MOV R0, B
        CJNE R0, #00H, CHECK1
        SJMP CHECK
```

### ; โปรแกรมย่อยในการข้ามเตอร์และเช็คสัญญาณหมด

```
CHECK1: JNB P0.0, CHECK2
         MOV B, #00H
         CLR P0.0
         INC R1
         SETB P2.7
LOOP:   JNB P0.2, LOOP
         CLR P2.7
         CLR P0.2
         CJNE R1, #06H, CHECK
         ACALL MAIN1
         RET
CHECK2: JNB P0.1, CHECK1
         MOV B, #00H
         CLR P0.1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        INC R2
        SETB P2.6
LOOP1:   JNB P0.3, LOOP1
        CLR P2.6
        CLR P0.3
        CJNE R2, #06H, CHECK
        ACALL MAIN2
        RET

```

**; โปรแกรมย่อยเมื่อสินค้าหมดให้ส่ง Email**

```

MAIN1:   ACALL S_AT_OP
        ACALL S_AT_SMTP
        ACALL DELAY
        ACALL S_AT_CL
        ACALL CHECK
        RET
MAIN2:   ACALL S_AT_OP
        ACALL S_AT_SMTP1
        ACALL DELAY
        ACALL S_AT_CL
        ACALL CHECK
        RET

```

**; โปรแกรมย่อยเช็คการติดต่อกับ TCP/IP Chip**

```

CN:      ACALL I_OK
        CJNE A, #59H, CN
        RET

```

**; โปรแกรมย่อยเช็คความพร้อม**

```

S_AT_I:  MOV DPTR, #AT_I
        ACALL SEND
        ACALL I_OK
        CJNE A, #4BH, S_AT_I
        RET

```

**; โปรแกรมย่อยทำการเซตหมายเลขติดต่อ GPRS**

```

S_AT_SD: MOV DPTR, #AT_SD
        ACALL SEND
        ACALL I_OK
        CJNE A, #4FH, S_AT_SD
        RET

```

**; โปรแกรมย่อยทำการติดต่อกับมือถือ**

```

S_AT_DT: MOV DPTR, #AT_DT
        ACALL SEND
        ACALL I_OK
        CJNE A, #30H, S_AT_DT
        MOV P2, #02H
        RET

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**; โปรแกรมย่อยในการเชื่อมต่อ GPRS**

```
S_AT_OP:
    MOV DPTR, #AT_OP
    ACALL SEND
    ACALL I_OK
    ACALL TIME
    CJNE A, #4BH, S_AT_OP
    RET
```

**; โปรแกรมย่อยทำการส่ง Email**

```
S_AT_SMTP:
    MOV DPTR, #AT_SMTP
    ACALL SEND
    RET
```

```
S_AT_SMTP1:
    MOV DPTR, #AT_SMTP1
    ACALL SEND
    RET
```

**; โปรแกรมย่อยทำการปิดการเชื่อมต่อ GPRS**

```
S_AT_CL:
    MOV DPTR, #AT_CL
    ACALL SEND
    MOV P2, #02H
    RET
```

**; โปรแกรมย่อยส่งข้อมูลออกทาง Serial Port**

```
SEND:
    CLR TI
LP_SD:
    CLR A
    MOVC A, @A+DPTR
    INC DPTR
    CJNE A, #0FFH, TX_CHAR
    RET
TX_CHAR:
    MOV SBUF, A
    JNB TI, $
    CLR TI
    SJMP LP_SD
```

**; โปรแกรมย่อยรับค่าจาก Serial Port เพื่อเช็คข้อความ Error**

```
I_OK:
    MOV DPTR, #DATA1
    MOV R0, #10
CHLOOP:
    JNB RI, $
    MOV A, SBUF
    CLR RI
    MOVX @DPTR, A
    INC DPTR
    DJNZ R0, CHLOOP
    MOV DPTR, #DATA1+9
    MOVX A, @DPTR
    RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**; ข้อความที่ใช้ในการส่งออก Serial Port**

```
AT_I:      DB 'at+i',0DH,0FFH
AT_SD:     DB 'at+isetd=*99***1#',0DH,0FFH
AT_DT:     DB 'at+idetct',0DH,0FFH
AT_OP:     DB 'at+iopen',0DH,0FFH
AT_CL:     DB 'at+iclose',0DH,0FFH
AT_SMTP:   DB
            'at+ismtpsend=mail.cscoms.com,PJ@yahoo.com,mbgons@yahoo.co.th,101 201
            20,M1',0DH,0FFH
AT_SMTP1:  DB
            'at+ismtpsend=mail.cscoms.com,PJ@yahoo.com,mbgons@yahoo.co.th,101 202
            20,M1',0DH,0FFH
```

**: บัฟเฟอร์ที่ใช้ในการเก็บค่าเพื่อในการส่งและรับจาก Serial Port**

```
DATA1:     DB 0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH
            DB 0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH
            DB 0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH
            DB 0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH
            DB 0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH
            DB 0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH
            DB 0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH
            DB 0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH
            DB 0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH
            DB 0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH

DELAY:     MOV R7,#30H
DELAY1:    MOV R6,#0FFH
DELAY2:    MOV R5,#0FFH
DELAY3:    NOP
            NOP
            NOP
            NOP
            NOP
            NOP
            DJNZ R5,DELAY3
            DJNZ R6,DELAY2
            DJNZ R7,DELAY1
            RET

TIME:      MOV R7,#01H
TIME1:     MOV R6,#0FFH
TIME2:     MOV R5,#0FFH
TIME3:     NOP
            NOP
            NOP
            NOP
            NOP
            NOP
            DJNZ R5,TIME3
            DJNZ R6,TIME2
            DJNZ R7,TIME1
            RET
            END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
##### frmMain.cs #####
```

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Collections;
using System.Data.OleDb;
using System.Threading;

namespace Pop3ToDB
{
    public partial class frmMain : Form
    {
        private string pop_server, username, password;
        private int port, TimeInterval;
        private bool edit_mode = false;

        private void ReadMail()
        {
            sttbarLabel1.Text = "กำลังอ่านเมล";
            try
            {
                Pop3 obj = new Pop3();
                obj.Connect(pop_server, username, password);
                ArrayList list = obj.List();
                DB MyDB = new DB();
                foreach (Pop3Message msg in list)
                {
                    Pop3Message msg2 = obj.Retrieve(msg);
                    string temp = msg2.message.Replace('\n', ' ');
                    temp = temp.Replace('\r', ' ');
                    string[] WordArray = temp.Split(new char[] { ' ' });
                    int x = 0;
                    foreach (string Words in WordArray)
                    {
                        x++;
                        if (Words == "Subject:")
                        {
                            if (MyDB.ConnectDB())
                            {
                                MyDB.QueryDB("INSERT INTO
                                [Order] ( m_ID, p_ID, Qty, OrderDate )
                                VALUES ('" + WordArray[x] + "', '" +
                                WordArray[x + 1] + "', '" + WordArray[x + 2]
                                + "', '" + DateTime.Today.ToShortDateString()
                                + "')");
                                MyDB.CloseDB();
                                obj.Delete(msg);
                            }
                            break;
                        }
                    }
                }
            }
            catch (Pop3Exception ee)
            {
                MessageBox.Show(ee.ToString(), "Error", MessageBoxButtons.
                OK, MessageBoxIcon.Error);
            }
        }
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        sttbarLabel1.Text = "อ่านเมลเสร็จแล้ว";
    }

    private void GetSetting()
    {
        DB MyDB = new DB();
        if (MyDB.ConnectDB())
        {
            MyDB.QueryDB("SELECT Setting.* FROM Setting;");
            txtPOPServer.Text = MyDB.pDataTable.Rows[0][0].ToString();
            txtPort.Text = MyDB.pDataTable.Rows[0][1].ToString();
            txtUsername.Text = MyDB.pDataTable.Rows[0][2].ToString();
            txtPassword.Text = MyDB.pDataTable.Rows[0][3].ToString();
            cmbTimeInterval.SelectedIndex =
                Int32.Parse(MyDB.pDataTable.Rows[0][4].ToString());
            MyDB.CloseDB();
        }
    }

    public frmMain()
    {
        InitializeComponent();
    }

    private void frmMain_Load(object sender, EventArgs e)
    {
        DB MyDB = new DB();
        if (MyDB.ConnectDB())
        {
            MyDB.QueryDB("SELECT Setting.* FROM Setting;");
            pop_server = MyDB.pDataTable.Rows[0][0].ToString();
            port = Int32.Parse(MyDB.pDataTable.Rows[0][1].ToString());
            username = MyDB.pDataTable.Rows[0][2].ToString();
            password = MyDB.pDataTable.Rows[0][3].ToString();
            TimeInterval =
                index2integer(MyDB.pDataTable.Rows[0][4].ToString());
            MyDB.CloseDB();
            timer1.Interval = TimeInterval;
            timer2.Interval = 5000;

            ThreadStart threadDelegate = new ThreadStart(ReadMail);
            Thread newThread = new Thread(threadDelegate);
            newThread.Start();
            timer1.Enabled = true;
            timer2.Enabled = true;

            LoadOrder();
        }
    }

    private int index2integer(string p)
    {
        int ret;
        switch (p)
        {
            case "0":
                ret = 300000; break; //milisec.
            case "1":
                ret = 600000; break;
            case "2":
                ret = 900000; break;
            case "3":
                ret = 1800000; break;
            case "4":
                ret = 3600000; break;
            default:
                ret = 300000; break;
        }
    }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    return ret;
}

private void frmMain_FormClosing(object sender, FormClosingEventArgs e)
{
    if (MessageBox.Show("ยืนยันการออกจากโปรแกรม?", "ยืนยัน",
        MessageBoxButtons.YesNo, MessageBoxIcon.Question) ==
        DialogResult.Yes)
        e.Cancel = false;
    else
    {
        timer1.Enabled = false;
        e.Cancel = true;
    }
}

private void tabControl1_Selecting(object sender,
    TabControlCancelEventArgs e)
{
    switch (e.TabPageIndex)
    {
        case 0:
            LoadOrder();
            break;
        case 1:
            txtProductName.Text = "";
            txtProductID.Text = "";
            txtProductID.Enabled = false;
            txtProductName.Enabled = false;
            btnProductAdd.Enabled = true;
            btnProductEdit.Enabled = true;
            btnProductCancel.Enabled = false;
            btnProductSave.Enabled = false;
            btnProductDelete.Enabled = false;
            edit_mode = false;
            LoadProduct();
            break;
        case 2:
            txtMachineAddress.Text = "";
            txtMachineID.Text = "";
            txtMachineID.Enabled = false;
            txtMachineAddress.Enabled = false;
            btnMachineAdd.Enabled = true;
            btnMachineEdit.Enabled = true;
            btnMachineCancel.Enabled = false;
            btnMachineSave.Enabled = false;
            btnMachineDelete.Enabled = false;
            edit_mode = false;
            LoadMachine();
            break;
        case 3:
            GetSetting();
            break;
        default:
            break;
    }
}

private void LoadOrder()
{
    string sql = "SELECT Order.m_ID FROM [Order] GROUP BY Order.m_ID
        ORDER BY Order.m_ID;";
    DB MyDB = new DB();
    if (MyDB.ConnectDB())
    {
        MyDB.QueryDB(sql);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

cmdMachineID.Items.Clear();

for(int x = 0; x < MyDB.pDataTable.Rows.Count; ++x)
    cmdMachineID.Items.Add(MyDB.pDataTable.Rows[x][0].ToString());
MyDB.CloseDB();
}
if (cmdMachineID.Items.Count > 0)
{
    cmdMachineID.SelectedIndex = 0;
    LoadOrder(cmdMachineID.SelectedItem.ToString());
}
}

private void LoadOrder(string m_ID)
{
    string sql = "SELECT Order.OrderDate, Product.p_Name, Order.Qty,
Machine.Address FROM Product INNER JOIN (Machine
INNER JOIN [Order] ON Machine.m_ID = Order.m_ID) ON
Product.p_ID = Order.p_ID WHERE
((Machine.m_ID)='"+m_ID+"')";
DB MyDB = new DB();
if (MyDB.ConnectDB())
{
    MyDB.QueryDB(sql);
    dgReport.DataSource = MyDB.pDataSet;
    dgReport.DataMember = "Query1";
    MyDB.CloseDB();
}
}

private void LoadProduct()
{
    string sql = "SELECT Product.* FROM Product ORDER BY
Product.p_ID";
DB MyDB = new DB();
if (MyDB.ConnectDB())
{
    MyDB.QueryDB(sql);
    dgProduct.DataSource = MyDB.pDataSet;
    dgProduct.DataMember = "Query1";
    MyDB.CloseDB();
}
}

private void LoadMachine()
{
    string sql = "SELECT Machine.* FROM Machine ORDER BY
Machine.m_ID";
DB MyDB = new DB();
if (MyDB.ConnectDB())
{
    MyDB.QueryDB(sql);
    dgMachine.DataSource = MyDB.pDataSet;
    dgMachine.DataMember = "Query1";
    // CustomGridProduct();
    MyDB.CloseDB();
}
}

private void btnSettingCancel_Click(object sender, EventArgs e)
{
    GetSetting();
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

private void btnSettingOK_Click(object sender, EventArgs e)
{
    DB MyDB = new DB();
    if (MyDB.ConnectDB())
    {
        MyDB.QueryDB("UPDATE Setting SET Setting.PopServer = '" +
            txtPOPServer.Text + "', Setting.Port = " + txtPort.Text + ",
            Setting.Username = '" + txtUsername.Text + "',
            Setting.[Password] = '" + txtPassword.Text + "',
            Setting.TimeInterval = " + cmbTimeInterval.SelectedIndex +
            "");
        MyDB.CloseDB();

        MessageBox.Show("ทำการบันทึกเรียบร้อยแล้ว.", "แจ้ง",
            MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}

private void dgProduct_Click(object sender, EventArgs e)
{
    txtProductID.Text = dgProduct.SelectedCells[0].Value.ToString();
    txtProductName.Text =
        dgProduct.SelectedCells[1].Value.ToString();
    btnProductDelete.Enabled = true;
}

private void btnProductEdit_Click(object sender, EventArgs e)
{
    edit_mode = true;
    txtProductID.Text = dgProduct.SelectedCells[0].Value.ToString();
    txtProductName.Text =
        dgProduct.SelectedCells[1].Value.ToString();
    dgProduct.Enabled = false;
    txtProductName.Enabled = true;
    btnProductDelete.Enabled = false;
    btnProductAdd.Enabled = false;
    btnProductEdit.Enabled = false;
    btnProductSave.Enabled = true;
    btnProductCancel.Enabled = true;
}

private void btnProductCancel_Click(object sender, EventArgs e)
{
    edit_mode = false;
    dgProduct.Enabled = true;
    txtProductID.Text = "";
    txtProductName.Text = "";
    txtProductID.Enabled = false;
    txtProductName.Enabled = false;
    btnProductAdd.Enabled = true;
    btnProductCancel.Enabled = false;
    btnProductEdit.Enabled = true;
    btnProductSave.Enabled = false;
}

private void btnProductAdd_Click(object sender, EventArgs e)
{
    edit_mode = false;
    dgProduct.Enabled = false;
    txtProductID.Text = "";
    txtProductName.Text = "";
    txtProductID.Enabled = true;
    txtProductName.Enabled = true;
    btnProductDelete.Enabled = false;
    btnProductAdd.Enabled = false;
    btnProductCancel.Enabled = true;
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยราชภัฏวไลยอลงกรณ์ ไม่อนุญาติให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        btnProductEdit.Enabled = false;
        btnProductSave.Enabled = true;
    }

private void btnProductSave_Click(object sender, EventArgs e)
{
    if (txtProductID.Text != "" && txtProductName.Text != "")
    {
        DB MyDB = new DB();
        if (MyDB.ConnectDB())
        {
            if (edit_mode)
                MyDB.QueryDB("UPDATE Product SET Product.p_Name = '"
                +txtProductName.Text.Trim()+"' WHERE
                (((Product.p_ID)='"+txtProductID.Text.Trim()+"')");");
            else
                MyDB.QueryDB("INSERT INTO [Product] ( p_ID, p_Name )
                VALUES ('" + txtProductID.Text.Trim() + "', '" +
                txtProductName.Text.Trim() + "')");");
            MyDB.CloseDB();
        }

        edit_mode = false;
        dgProduct.Enabled = true;
        txtProductID.Text = "";
        txtProductName.Text = "";
        txtProductID.Enabled = false;
        txtProductName.Enabled = false;
        btnProductAdd.Enabled = true;
        btnProductCancel.Enabled = false;
        btnProductEdit.Enabled = true;
        btnProductSave.Enabled = false;
        LoadProduct();
    }
    else
        MessageBox.Show("กรุณากรอกข้อมูลให้ครบ!", "Error",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
}

private void btnProductDelete_Click(object sender, EventArgs e)
{
    if (txtProductID.Text != "")
    {
        if (MessageBox.Show("ยืนยันการลบสินค้าหมายเลข "+txtProductID.Text, "
        ยืนยัน", MessageBoxButtons.YesNo, MessageBoxIcon.Question) ==
        DialogResult.Yes)
        {
            DB MyDB = new DB();
            if (MyDB.ConnectDB())
            {
                MyDB.QueryDB("DELETE * FROM Product WHERE (Product.p_ID
                ='"+ txtProductID.Text.Trim() + "')");");
                MyDB.CloseDB();
            }
            btnProductDelete.Enabled = false;
            txtProductID.Text = "";
            txtProductName.Text = "";
            LoadProduct();
        }
    }
    else
        MessageBox.Show("เลือกสินค้าที่ต้องการลบ!", "Error",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

private void btnMachineAdd_Click(object sender, EventArgs e)
{
    edit_mode = false;
    dgMachine.Enabled = false;
    txtMachineID.Text = "";
    txtMachineAddress.Text = "";
    txtMachineID.Enabled = true;
    txtMachineAddress.Enabled = true;
    btnMachineDelete.Enabled = false;
    btnMachineAdd.Enabled = false;
    btnMachineCancel.Enabled = true;
    btnMachineEdit.Enabled = false;
    btnMachineSave.Enabled = true;
}

private void btnMachineEdit_Click(object sender, EventArgs e)
{
    edit_mode = true;
    txtMachineID.Text = dgMachine.SelectedCells[0].Value.ToString();
    txtMachineAddress.Text =
dgMachine.SelectedCells[1].Value.ToString();
    dgMachine.Enabled = false;
    txtMachineAddress.Enabled = true;
    btnMachineDelete.Enabled = false;
    btnMachineAdd.Enabled = false;
    btnMachineEdit.Enabled = false;
    btnMachineSave.Enabled = true;
    btnMachineCancel.Enabled = true;
}

private void btnMachineDelete_Click(object sender, EventArgs e)
{
    if (txtMachineID.Text != "")
    {
        if (MessageBox.Show("ยืนยันการลบเครื่องขายสินค้าหมายเลข " +
txtMachineID.Text, "ยืนยัน", MessageBoxButtons.YesNo,
MessageBoxIcon.Question) == DialogResult.Yes)
        {
            DB MyDB = new DB();
            if (MyDB.ConnectDB())
            {
                MyDB.QueryDB("DELETE * FROM Machine WHERE (Machine.m_ID
= '" + txtMachineID.Text.Trim() + "');");
                MyDB.CloseDB();
            }
            btnMachineDelete.Enabled = false;
            txtMachineID.Text = "";
            txtMachineAddress.Text = "";
            LoadMachine();
        }
    }
    else
        MessageBox.Show("เลือกสินค้าที่ต้องการลบ!", "Error",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
}

private void btnMachineSave_Click(object sender, EventArgs e)
{
    if (txtMachineID.Text != "" && txtMachineAddress.Text != "")
    {
        DB MyDB = new DB();
        if (MyDB.ConnectDB())
        {
            if (edit_mode)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        MyDB.QueryDB("UPDATE Machine SET Machine.Address =
        '" + txtMachineAddress.Text.Trim() + "' WHERE
        ((Machine.m_ID)='" + txtMachineID.Text.Trim() +
        "')");
    else
        MyDB.QueryDB("INSERT INTO [Machine] ( m_ID,
        Address ) VALUES ('" + txtMachineID.Text.Trim() + '",
        '" + txtMachineAddress.Text.Trim() + "')");
    MyDB.ClosedB();
}

edit_mode = false;
dgMachine.Enabled = true;
txtMachineID.Text = "";
txtMachineAddress.Text = "";
txtMachineID.Enabled = false;
txtMachineAddress.Enabled = false;
btnMachineAdd.Enabled = true;
btnMachineCancel.Enabled = false;
btnMachineEdit.Enabled = true;
btnMachineSave.Enabled = false;
LoadMachine();
}
else
    MessageBox.Show("กรุณากรอกข้อมูลให้ครบ!", "Error",
    MessageBoxButtons.OK, MessageBoxIcon.Error);
}

private void btnMachineCancel_Click(object sender, EventArgs e)
{
    edit_mode = false;
    dgMachine.Enabled = true;
    txtMachineID.Text = "";
    txtMachineAddress.Text = "";
    txtMachineID.Enabled = false;
    txtMachineAddress.Enabled = false;
    btnMachineAdd.Enabled = true;
    btnMachineCancel.Enabled = false;
    btnMachineEdit.Enabled = true;
    btnMachineSave.Enabled = false;
}

private void dgMachine_Click(object sender, EventArgs e)
{
    txtMachineID.Text = dgMachine.SelectedCells[0].Value.ToString();
    txtMachineAddress.Text =
        dgMachine.SelectedCells[1].Value.ToString();
    btnMachineDelete.Enabled = true;
}

private void cmdMachineID_SelectedIndexChanged(object sender,
    EventArgs e)
{
    LoadOrder(cmdMachineID.Text.ToString());
}

private void timer1_Tick(object sender, EventArgs e)
{
    ThreadStart threadDelegate = new ThreadStart(ReadMail);
    Thread newThread = new Thread(threadDelegate);
    newThread.Start();
    timer2.Enabled = true;
}

private void timer2_Tick(object sender, EventArgs e)
{
    LoadOrder();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

)
)
##### POP3.cs #####

using System;
using System.Windows.Forms;
using System.Collections.Generic;
using System.Text;
using System.Collections;
using System.Net.Sockets;
using System.Diagnostics;

namespace Pop3ToDB
{
    public class Pop3Exception : System.ApplicationException
    {
        public Pop3Exception(string str)
            : base(str)
        {
        }
    }

    public class Pop3Message
    {
        public long number;
        public long bytes;
        public bool retrieved;
        public string message;
    }

    public class Pop3 : System.Net.Sockets.TcpClient
    {
        public void Connect(string server, string username, string password)
        {
            string message;
            string response;

            Connect(server, 110);

            response = Response();
            if (response.Substring(0, 3) != "+OK")
            {
                throw new Pop3Exception(response);
            }

            message = "USER " + username + "\r\n";
            Write(message);
            response = Response();
            if (response.Substring(0, 3) != "+OK")
            {
                throw new Pop3Exception(response);
            }

            message = "PASS " + password + "\r\n";
            Write(message);
            response = Response();
            if (response.Substring(0, 3) != "+OK")
            {
                throw new Pop3Exception(response);
            }
        }

        public void Disconnect()
        {
            string message;
            string response;
            message = "QUIT\r\n";
            Write(message);
            response = Response();
            if (response.Substring(0, 3) != "+OK")

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        {
            throw new Pop3Exception(response);
        }
    }
public ArrayList List()
{
    string message;
    string response;

    ArrayList retval = new ArrayList();
    message = "LIST\r\n";
    Write(message);
    response = Response();
    if (response.Substring(0, 3) != "+OK")
    {
        throw new Pop3Exception(response);
    }

    while (true)
    {
        response = Response();
        if (response == ".\r\n")
        {
            return retval;
        }
        else
        {
            Pop3Message msg = new Pop3Message();
            char[] seps = { ' ' };
            string[] values = response.Split(seps);
            msg.number = Int32.Parse(values[0]);
            msg.bytes = Int32.Parse(values[1]);
            msg.retrieved = false;
            retval.Add(msg);
            continue;
        }
    }
}
public Pop3Message Retrieve(Pop3Message rhs)
{
    string message;
    string response;

    Pop3Message msg = new Pop3Message();
    msg.bytes = rhs.bytes;
    msg.number = rhs.number;

    message = "RETR " + rhs.number + "\r\n";
    Write(message);
    response = Response();
    if (response.Substring(0, 3) != "+OK")
    {
        throw new Pop3Exception(response);
    }

    msg.retrieved = true;
    while (true)
    {
        response = Response();
        if (response == ".\r\n")
        {
            break;
        }
        else
        {
            msg.message += response;
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        return msg;
    }
    public void Delete(Pop3Message rhs)
    {
        string message;
        string response;

        message = "DELE " + rhs.number + "\r\n";
        Write(message);
        response = Response();
        if (response.Substring(0, 3) != "+OK")
        {
            throw new Pop3Exception(response);
        }
    }
    private void Write(string message)
    {
        System.Text.ASCIIEncoding en = new System.Text.ASCIIEncoding();

        byte[] WriteBuffer = new byte[1024];
        WriteBuffer = en.GetBytes(message);

        NetworkStream stream = GetStream();
        stream.Write(WriteBuffer, 0, WriteBuffer.Length);

        Debug.WriteLine("WRITE:" + message);
    }
    private string Response()
    {
        System.Text.ASCIIEncoding enc = new System.Text.ASCIIEncoding();
        byte[] serverbuff = new Byte[1024];
        NetworkStream stream = GetStream();
        int count = 0;
        while (true)
        {
            byte[] buff = new Byte[2];
            int bytes = stream.Read(buff, 0, 1);
            if (bytes == 1)
            {
                serverbuff[count] = buff[0];
                count++;

                if (buff[0] == '\n')
                {
                    break;
                }
            }
            else
            {
                break;
            }
        };

        string retval = enc.GetString(serverbuff, 0, count);
        Debug.WriteLine("READ:" + retval);
        return retval;
    }
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
##### DB.cs #####
```

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Data;
using System.Windows.Forms;
using System.Data.OleDb;

namespace Pop3ToDB
{
    class DB
    {
        private OleDbConnection Conn = new OleDbConnection();
        private OleDbDataAdapter pDataAdapter = null;
        public DataTable pDataTable = null;
        public DataSet pDataSet = null;
        private string DBPath = "F:/AEM PROJECT
                                2006/POP3toDB/Pop3ToDB/DatabasePRO.mdb";

        public bool ConnectDB()
        {
            string strConn = "Provider=Microsoft.Jet.OLEDB.4.0;Data
                               Source="+DBPath+";Jet OLEDB:Engine Type=5";
            if (Conn.State == ConnectionState.Open)
                Conn.Close();
            try
            {
                Conn = new OleDbConnection(strConn);
                Conn.Open();
            }
            catch
            {
                MessageBox.Show("ไม่สามารถเชื่อมต่อฐานข้อมูล " + DBPath + " ได้.",
                                "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
                return false;
            }
            return true;
        }

        public void QueryDB(string SQL)
        {
            string cmdType = SQL.Substring(0, 3).ToUpper();
            OleDbCommand dbCMD = new OleDbCommand(SQL);
            pDataAdapter = new OleDbDataAdapter();
            pDataTable = new DataTable();
            pDataSet = new DataSet();
            switch (cmdType)
            {
                case "SEL":
                    pDataAdapter.SelectCommand = dbCMD;
                    pDataAdapter.SelectCommand.Connection = Conn;
                    pDataAdapter.SelectCommand.ExecuteNonQuery();
                    pDataAdapter.Fill(pDataTable);
                    pDataAdapter.Fill(pDataSet, "Query1");
                    break;
                case "INS":
                    pDataAdapter.InsertCommand = dbCMD;
                    pDataAdapter.InsertCommand.Connection = Conn;
                    pDataAdapter.InsertCommand.ExecuteNonQuery();
                    break;
                case "UPD":
                    pDataAdapter.UpdateCommand = dbCMD;
                    pDataAdapter.UpdateCommand.Connection = Conn;
                    pDataAdapter.UpdateCommand.ExecuteNonQuery();
                    break;
            }
        }
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
case "DEL":
    pDataAdapter.DeleteCommand = dbcMD;
    pDataAdapter.DeleteCommand.Connection = Conn;
    pDataAdapter.DeleteCommand.ExecuteNonQuery();
    break;
}
}
public void CloseDB()
{
    pDataTable.Clear();
    Conn.Close();
}
}
```

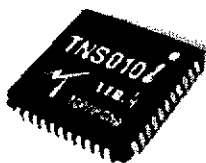


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## TNS010i TCP/IP stack Chip



### Description

TNS010i TCP/IP stack chip is a complete single turnkey product that enables the Internet connection. Not only does it reduce development time to market, the TNS010i allows several real-time efficient transmissions of data and commands between an 8/16/32-bit micro controller (MCU) and the core TCP/IP software engine. With no prior information about the Internet protocol or networking, this device allows the MCU to send and receive web pages, data, and commands in a form of scripting phrases through its built in TCP/IP protocol engine.

Executing only a few simple commands from an MCU through serial pins of TNS010i and you will be able to transmit and receive data to/from web application program. Each command sent across the TNS010i will invoke TNS010i to response back to an MCU with a message. This debugging information helps users keep track of every step during the connection attempt.

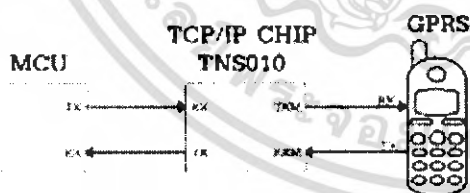
This device can connect to the GPRS phone directly with no additional components. Even the MCU needs to set-up or read out the GPRS parameters, this device automatically bypasses those AT commands to the phone thus eliminating

- The need for two serial-port hardware on the MCU side, and
- A complex circuit for switching among the three modules: MCU, TNS010i, and GPRS.

### Features

- Support basic protocols including TCP/IP, PPP, DNS, HTTP, SMTP, POP3 and TCP/IP socket.
- Easy interface to GPRS phone or Modem.
- Input commands length as long as 1,000 bytes.
- No need for external RAM or ROM.
- Minimal components. Requires only one 18.432 MHz Xtal, 3 Capacitors.
- 19,200 BPS Command communicates with MCU
- Information exchanges with GPRS phone is variable from 1,200 BPS to 57,600 BPS
- low power consumption
- 44-pin PLCC and LQFP package

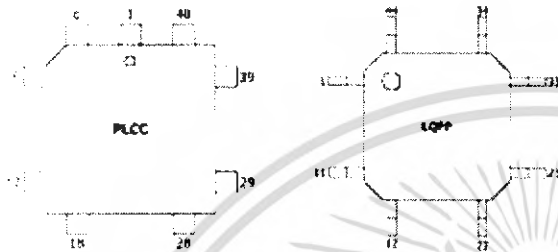
### The Connection Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Electrical Characteristics

Symbol	Description	Min	Type	Max	Unit
V <sub>L</sub>	Input Low Voltage	-0.5		0.9	V.
V <sub>H</sub>	Input High Voltage	1.9		5.5	V.
V <sub>H<sub>L</sub></sub>	Input High Voltage XTAL1,RST	3.5		5.5	V.
I <sub>cc</sub>	Power supply current			60	mA
V <sub>cc</sub>	Power supply voltage	4.75		5.25	V.



## Pin Description

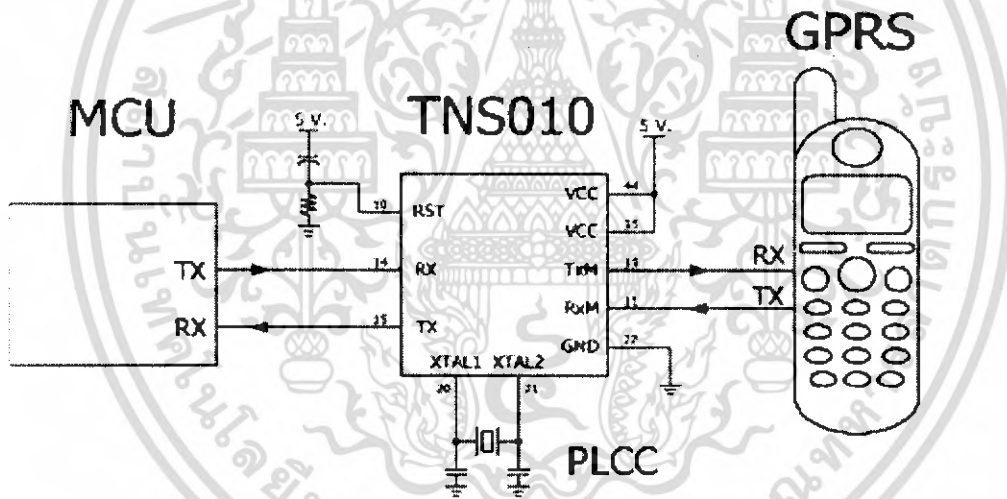
Symbol	PLCC	GPP	In/Out	Name and Function
V <sub>cc</sub>	44, 35	38, 29	I	Power Supply : power supply voltage for operation
Gnd	22	16	I	Ground : 0 V. reference
XTAL1	21	15	I	Crystal 1: 18.432 MHz. Input oscillator amplifier.
XTAL2	20	14	O	Crystal 2: 18.432 MHz. Output oscillator amplifier.
GPRS/modem	18	12	I	GPRS / Land line modem time out select
CD	2	40	I	Carrier detect
DTR	3	41	O	Data terminal ready
DSR	4	42	I	Data set ready
RTS	5	43	O	Request to send
CTS	6	44	I	Clear to send
RING	7	1	I	Ringng Indicator
RST	10	4	I	Reset : hold high at least 0.22 mS
TxM	13	7	O	Transmit to modem/GPRS phone
RxM	11	5	I	Receive from modem/GPRS phone
Tx	15	9	O	Transmit to MCU
Rx	14	8	I	Receive from MCU

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### TCP/IP Software Specifications

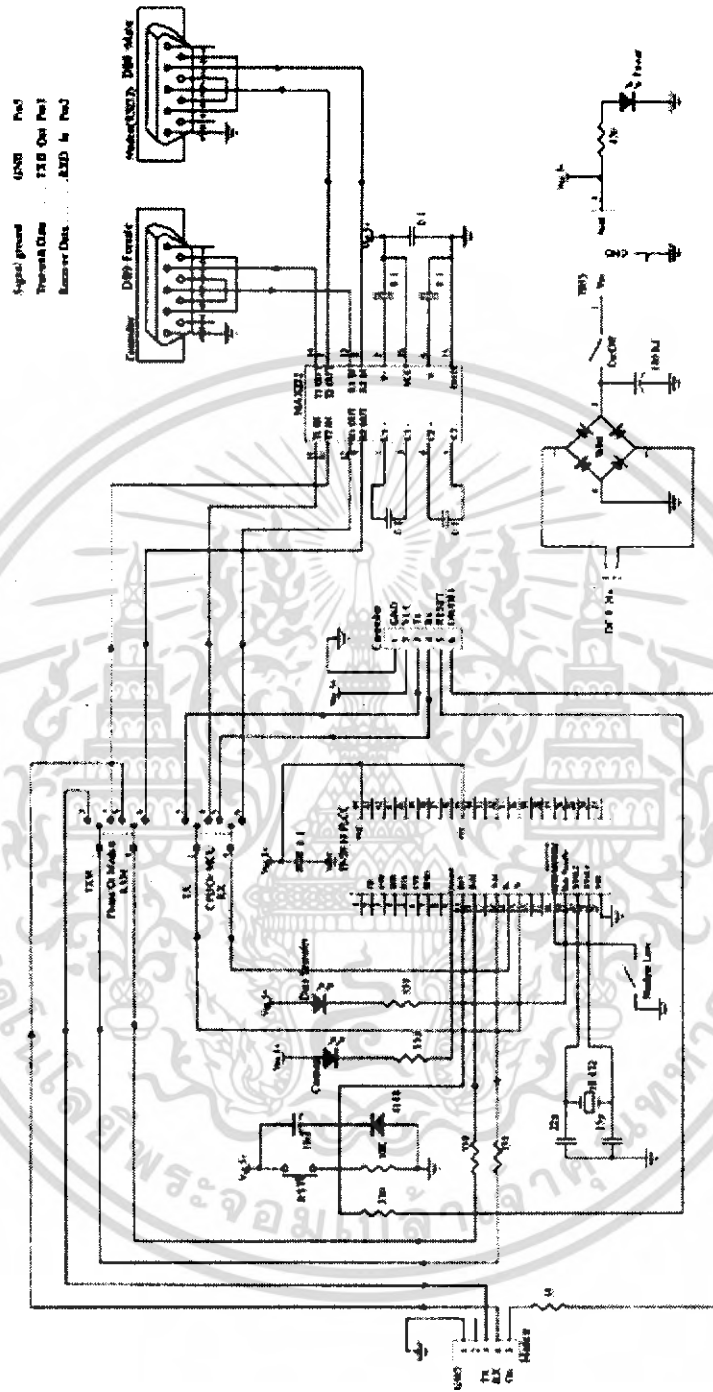
Maximum at+ command length (include at+ itself)	1000 bytes
Maximum TCP segment size	512 bytes
Serial RX buffer length (Ring buffer)	2,500 bytes
Serial TX buffer length	1,000 bytes
FTP	Support
DHAP	Not support
DNS	Support
HTTP	Support
RTSP	Not support
SMTP	Support
POP3	Support
TCP/IP socket	Support

### Typical testing circuit



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DEY-010i-B schematic

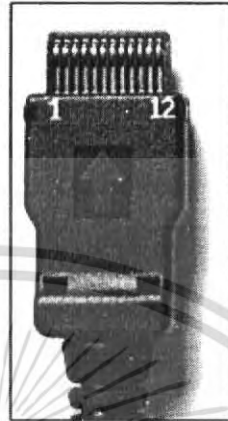


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Popular GPRS phone data link details

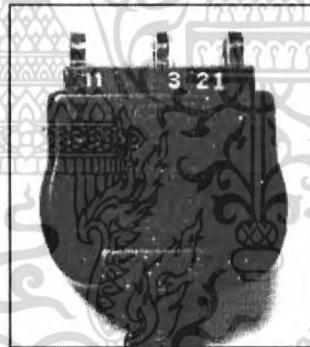
### Siemens S45 connector's pin out

1. Grd
2. -
3. Charge
4. -
5. Tx (out from phone)
6. Rx (into phone)
7. -
8. -
9. -
10. -
11. -
12. -



### Ericsson T65/68/200 connector's pin out

1. Charge
2. Grd
3. -
4. Grd
5. -
6. -
7. TX (out from phone)
8. RX (into phone)
9. Power ON (TRIG LOW) \*\*\*\* ONLY ERICSSON T65/T66 \*\*\*\*
10. -
11. -



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้