



**โปรแกรมซอฟต์แวร์การกระจายข่าวสารด้วยระบบ SMS
(SMS Broadcasting Program)**

นายณัฐพงศ์

นายณัฐวุฒิ

นางสาวชฎารัตน์

โดย

แซมหู

ภูงามเงิน
หาดทวยกาญจน์

รหัสนักศึกษา B5003303

รหัสนักศึกษา B5024353

รหัสนักศึกษา B5027934

รายงานนี้เป็นส่วนหนึ่งของการศึกษาวิชา 427499 โครงการวิศวกรรมโทรคมนาคม และ
วิชา 427494 โครงการศึกษาวิศวกรรมโทรคมนาคม
ประจำภาคการศึกษาที่ 1 และ 2 ปีการศึกษา 2553

หลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมโทรคมนาคม หลักสูตรปรับปรุง พ.ศ. 2545

สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

โครงการ	โปรแกรมซอฟต์แวร์การกระจายข่าวสารด้วยระบบ SMS (SMS Broadcasting Program)
จัดทำโดย	นายณัฐพงศ์ แซ่หู นายณัฐวุฒิ ภูงามเงิน นางสาวชฎารัตน์ หาดทวยกาญจน์
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร.พีระพงษ์ อุซารสกุล
สาขาวิชา	วิศวกรรมโทรคมนาคม
ภาคการศึกษาที่	2/2553

บทคัดย่อ

(Abstract)

การส่งข้อมูลในระบบ SMS (Short Message Service) ในปัจจุบันมีการใช้งานในหลากหลายรูปแบบ ซึ่งระบบนี้เป็นระบบที่มีขนาดใหญ่ มีข้อจำกัดในเรื่องค่าใช้จ่ายที่จะต้องเสียให้กับผู้ให้บริการและเสียเวลาดำเนินการส่งถึงผู้รับในจำนวนหลายคนในเวลาที่ยกจำกัด รวมทั้งจำเป็นต้องมีผู้ที่มีความรู้ทางด้านคอมพิวเตอร์ในการควบคุมด้วย ทำให้ไม่คุ้มค่าในการลงทุน เมื่อใช้กับองค์กรขนาดกลางหรือขนาดเล็ก ดังนั้นผู้จัดทำจึงได้พัฒนาโปรแกรมซอฟต์แวร์การกระจายข่าวสารด้วยระบบ SMS (SMS Broadcasting Program) มาเพื่อแก้ปัญหานี้ โดยมีหลักการคือ การนำข้อมูลของผู้รับมาจัดการในเครื่องคอมพิวเตอร์ แล้วจัดส่งข้อความและกลุ่มเบอร์เป้าหมายจากเครื่องคอมพิวเตอร์ไปยังโทรศัพท์เคลื่อนที่ผ่านบอร์ด GSM Module Wireless CPU ไปยังกลุ่มผู้รับ ซึ่งทางผู้จัดทำได้ใช้ความรู้เกี่ยวกับการเขียนโปรแกรม Visual C# มาพัฒนาเขียนโปรแกรมเพื่อที่จะทำให้บอร์ด GSM Module Wireless CPU ทำงานร่วมกับคอมพิวเตอร์ได้เพื่อความสะดวกต่อการส่งข้อความของผู้ใช้งาน ซึ่งระบบที่ทางผู้จัดทำได้พัฒนานี้ จะมีความปลอดภัยต่อผู้ใช้งาน เนื่องจากต้องทำการ Log in เข้าระบบ ก่อนที่จะทำการส่งข้อความ และยังสามารถกำหนดวันในการส่งข้อความ ซึ่งจะส่งผลดีต่อผู้ใช้งาน

โปรแกรมซอฟต์แวร์การกระจายข่าวสารด้วยระบบ SMS
(SMS Broadcasting Program)

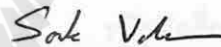
คณะกรรมการสอบโครงการ



(ผู้ช่วยศาสตราจารย์ ดร. พีระพงษ์ อุฑารสกุล)
กรรมการ/อาจารย์ที่ปรึกษาโครงการ



(ผู้ช่วยศาสตราจารย์ เรืออากาศเอก ดร. ประโยชน์ คำสวัสดิ์)
กรรมการ



(อาจารย์ ดร. สมศักดิ์ วาณิชอนันต์ชัย)
กรรมการ

มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้นำรายงานโครงการฉบับนี้ เป็นส่วนหนึ่งของการศึกษาระดับปริญญาตรี สาขาวิชาวิศวกรรมโทรคมนาคม วิชา 427499 โครงการวิศวกรรมโทรคมนาคม และวิชา 427494 โครงการศึกษาวิศวกรรมโทรคมนาคม ประจำปีการศึกษา 2553

กิตติกรรมประกาศ (Acknowledgement)

จากการที่คณะจัดทำรายงานได้รับมอบหมายให้ทำโครงการเรื่อง โปรแกรมซอฟต์แวร์การกระจายข่าวสารด้วยระบบ SMS (SMS Broadcasting Program) ส่งผลให้คณะจัดทำรายงานได้รับความรู้และประสบการณ์ต่างๆเกี่ยวกับการเขียนโปรแกรมเป็นอย่างมาก โดยการทำรายงานนี้ทางคณะจัดทำได้ใช้โปรแกรมวิซวลซีชาร์ป (visual C#) บัดนี้โครงการดังกล่าวพร้อมทั้งรายงานได้สำเร็จลงแล้ว ทั้งนี้ได้รับความร่วมมือและสนับสนุนจากบุคคลต่างๆ ดังนี้

1. ผศ.ดร. พิระพงษ์ อุฑารสกุล (อาจารย์ที่ปรึกษาโครงการ)
2. นางสาวปณิษฐาท์ ออหาญ (เลขานุการสาขาวิศวกรรมโทรคมนาคม)
3. นักศึกษาสาขาวิศวกรรมโทรคมนาคม ทุกชั้นปี

ข้าพเจ้าใคร่ขอขอบพระคุณผู้ที่มีส่วนเกี่ยวข้องทุกท่านที่มีส่วนร่วมในการให้ข้อมูลและเป็นที่ปรึกษาในการทำรายงาน ฉบับนี้จนเสร็จสมบูรณ์ ตลอดจนให้การดูแลและให้ความเข้าใจเกี่ยวกับพื้นฐานการใช้งานโปรแกรม ซึ่งข้าพเจ้าขอขอบพระคุณเป็นอย่างสูงไว้ ณ ที่นี้ด้วย

นายณัฐพงศ์ แซ่หู่

นายณัฐวุฒิ ภูงามเงิน

นางสาวชฎารัตน์ หาดทวยกาญจน์

สารบัญ

เรื่อง	หน้า
บทคัดย่อ.....	ก
กิตติกรรมประกาศ.....	ข
สารบัญ	ค
สารบัญรูป.....	จ
สารบัญตาราง.....	ช
บทที่ 1 บทนำ.....	1
1.1 วัตถุประสงค์.....	3
1.2 ขอบเขตการทำงาน.....	3
1.3 ขั้นตอนการดำเนินงาน.....	3
1.4 ผลที่คาดว่าจะได้รับ.....	4
บทที่ 2 ระบบโทรศัพท์เคลื่อนที่ GSM และบริการข่าวสารสั้น.....	5
2.1 บทนำ.....	5
2.2 ระบบ โทรศัพท์เคลื่อนที่ GSM.....	5
2.3 การใช้โทรศัพท์ในการติดต่อสื่อสาร (Mobile Phone for Human Communications).7	7
2.4 บริการส่งข้อความสั้น (Short Message Service: SMS).....9	9
2.5 หลักการรับส่งข้อความสั้น หรือ SMS ของโทรศัพท์มือถือ.....10	10
2.6 โหมดของการรับส่งข้อมูล SMS.....11	11
2.7 รูปแบบในการส่งข้อมูลในรูปแบบ SMS ผ่านทาง AT Command.....12	12
2.8 ข้อดีของการใช้งานบริการส่งข่าวสารสั้น ในระบบโทรศัพท์เคลื่อนที่ GSM.....12	12
บทที่ 3 โปรแกรม Microsoft Visual C#.Net และฐานข้อมูล SQL.....	14
3.1 บทนำ.....	14
3.2 ทำความรู้จักกับภาษา Visual C#.....	14
3.3 ข้อสังเกตในการเขียน โปรแกรมด้วยภาษา Visual C#.NET.....	15
3.4 บล็อกของคำสั่ง (Block).....	16
3.5 ชนิดของข้อมูล (Data Type).....	16
3.6 ตัวแปร (Variable).....	17
3.7 เครื่องหมายดำเนินการ (Operator).....	18

สารบัญ (ต่อ)

เรื่อง	หน้า
3.8 การเขียนหมายเหตุ (Comment).....	20
3.9 รู้จักภาษา SQL.....	20
บทที่ 4 ข้อมูลของฮาร์ดแวร์ใน เครื่องงาน.....	25
4.1 บทนำ.....	25
4.2 ภาพรวมของอุปกรณ์ต้นแบบ.....	25
4.3 ชุดบอร์ด GSM Module Wireless CPU.....	26
4.4 โครงสร้าง SIM Card โทรศัพท์เคลื่อนที่.....	28
บทที่ 5 การออกแบบระบบและพัฒนาโปรแกรม.....	36
5.1 บทนำ.....	36
5.2 ภาพรวมการออกแบบ แสดงด้วยสถาปัตยกรรมของระบบ.....	37
5.3 คลาสออกแบบ (Design Class).....	39
5.4 แผนภาพแสดงพฤติกรรมเชิงพลวัต.....	40
5.5 แผนผังแสดงความสัมพันธ์.....	41
5.6 พจนานุกรมข้อมูล.....	42
5.7 การติดตั้งโปรแกรมในการใช้งาน.....	45
5.8 ส่วนต้นแบบของ โครงร่างของส่วนเชื่อมโยงกับผู้ใช้.....	49
บทที่ 6 สรุปผลและข้อเสนอแนะ.....	54
6.1 บทนำ.....	54
6.2 ปัญหาและแนวทางในการแก้ไขปัญหา.....	54
6.3 ข้อเสนอแนะ.....	56
6.4 แนวทางในการพัฒนาต่อไป.....	56
6.5 บทสรุป.....	57
ประวัติผู้เขียน.....	61
ภาคผนวก.....	62
บรรณานุกรม.....	95

สารบัญรูป

รายการ	หน้า
รูป 2.1 แสดง รูปแบบการติดต่อสื่อสารแบบ GSM	7
รูป 2.2 แสดง การสนทนาของ A และ B.....	8
รูป 2.3 แสดง ขอบเขตของการส่ง SMS.....	13
รูป 3.1 แสดงตัวอย่างข้อมูลจาก Table Personal.....	21
รูป 3.2 แสดงตัวอย่างข้อมูลจากคำสั่ง <i>Select * from Personal</i>	22
รูป 3.3 แสดงตัวอย่างข้อมูลจากคำสั่ง <i>Select Top 5 * from Personal</i>	23
รูป 3.4 แสดงตัวอย่างข้อมูลจากคำสั่ง <i>Select Name + ' ' + Surname as ชื่อ , MOD_ID from personal</i>	24
รูป 4.1 ภาพมุมมองบอร์ดจากด้านบน.....	25
รูป 4.2 ภาพมุมมองบอร์ดจากด้านข้าง.....	26
รูป 4.3 ลักษณะบรรจุภัณฑ์.....	28
รูป 4.4 ลักษณะ SIM Card.....	29
รูป 4.5 การตรวจสอบสถานะความจำของ SIM Card.....	30
รูป 4.6 การตั้งค่าการใช้งานรหัส PIN.....	31
รูป 4.7 รหัส PUK CODE ที่มาพร้อม SIM Card.....	31
รูป 4.8 การใช้งาน SIM Card นอกเหนือจากโทรศัพท์เคลื่อนที่.....	33
รูป 4.9 รายละเอียดเกี่ยวกับการใช้งาน.....	35
รูป 5.1 แสดงสถาปัตยกรรมของระบบเชิงตรรกะ.....	37
รูป 5.2 แสดงสถาปัตยกรรมของระบบเชิงกายภาพ.....	38
รูป 5.3 แสดง Class Diagram ของการจัดการระบบฐานข้อมูล.....	39
รูป 5.4 แสดง Sequence diagram ของโปรแกรมส่ง SMS.....	40
รูป 5.5 แสดงแผนผังแสดงความสัมพันธ์ของโปรแกรมส่ง SMS.....	41
รูป 5.6 แสดงตัวติดตั้งโปรแกรมเมื่อทำการใส่แผ่น CD RO.....	45
รูป 5.7 หน้าจอแสดงข้อความต้อนรับการติดตั้งโปรแกรม SMS Broadcasting Program.....	45
รูป 5.8 แสดงข้อมูลของโปรแกรม.....	46
รูป 5.9 แสดงถึง License ของโปรแกรม.....	46

สารบัญรูป (ต่อ)

รายการ	หน้า
รูป 5.10 แสดงหน้าจอการเลือกที่ตั้งของโปรแกรมเมื่อทำการ Installation โปรแกรม.....	47
รูป 5.11 แสดงหน้าจอการตอบรับการลงโปรแกรมลงเครื่องผู้ใช้งาน.....	47
รูป 5.12 แสดงการโหลดตัวโปรแกรมเพื่อทำการติดตั้ง.....	48
รูป 5.13 แสดงหน้าจอเมื่อทำการติดตั้งโปรแกรมเสร็จแล้ว.....	48
รูป 5.14 แสดงหน้าจอการตั้งค่าโปรแกรม.....	49
รูป 5.15 แสดงหน้าจอแสดงหน้าหลักของโปรแกรม.....	50
รูป 5.16 แสดงหน้าจอการจัดการการส่งข้อความ.....	51
รูป 5.17 แสดงหน้าจอการจัดการรายชื่อติดต่อ.....	51
รูป 5.18 แสดงหน้าจอการกำหนดผู้ใช้งาน.....	52
รูป 5.19 แสดงหน้าจอการกำหนดกลุ่มที่ใช้ในการติดต่อ.....	52
รูป 5.20 แสดงหน้าจอรายงานการส่งข้อความ.....	53
รูป 5.21 แสดงหน้ารายงานการเข้าใช้ระบบ.....	53



สารบัญตาราง

รายการ	หน้า
ตาราง 1.1 ตารางการเปรียบเทียบราคา.....	2
ตาราง 2.1 ข้อดี – ข้อด้อยของการส่งข่าวสาร.....	13
ตาราง 3.1 ตารางแสดงประเภทของข้อมูล.....	17
ตาราง 3.2 ตารางแสดงเครื่องหมายดำเนินการทางคณิตศาสตร์.....	18
ตาราง 3.3 ตารางแสดงเครื่องหมายดำเนินการทางคณิตศาสตร์.....	19
ตาราง 5.1 ตาราง members.....	42
ตาราง 5.2 ตาราง G group.....	42
ตาราง 5.3 ตาราง M group.....	42
ตาราง 5.4 ตาราง Log user.....	43
ตาราง 5.5 ตาราง report login_error.....	43
ตาราง 5.6 ตาราง report_login_s.....	43
ตาราง 5.7 ตาราง report_send.....	44
ตาราง 5.8 ตาราง send_time.....	44
ตาราง 6.1 ปัญหาและสาเหตุที่พบในขณะที่ดำเนินงานและวิธีการแก้ไข.....	55
ตาราง 6.2 ตารางการเปรียบเทียบการส่ง SMS.....	60

บทที่ 1

บทนำ

ในปัจจุบันสังคมยุคใหม่ ได้มีการนำเทคโนโลยีที่ทันสมัยเข้ามาผสมผสานจนมีบทบาทในชีวิตประจำวันอย่างมาก และที่จะยกตัวอย่างต่อไปนี้ ก็จัดว่าเป็นเทคโนโลยีที่ได้รับความนิยมเป็นอย่างมากเช่นกันคือ การนำเอาเทคโนโลยีเกี่ยวกับการส่งข้อความสั้น หรือ ที่รู้จักกันดีในชื่อ SMS (Short Message Service) เข้ามาช่วยในการกระจายข่าวสารต่างๆ อาทิเช่น การให้บริการแนะนำโปรแกรมของ ผู้ให้บริการ โทรศัพท์เคลื่อนที่เครือข่ายต่างๆ หรือการแนะนำสินค้าใหม่ๆ ของผู้ให้บริการ และยังมีบุคคลที่ต้องการนำเสนอสินค้าและบริการของตนเองใช้บริการเช่าแฟ้มกดพิมพ์ส่งข้อความสั้นเพื่อเป็นการส่งเสริมการขายของบริษัทหรือองค์กร แต่สำหรับการเช่าแฟ้มกดนั้นจะเห็นได้ว่ามีราคาที่ค่อนข้างแพง (เปรียบเทียบราคาได้จากตาราง 1.1) ถ้าผู้ใช้บริการมีขนาดขององค์กรขนาดเล็ก มีปัญหาด้านงบประมาณที่ค่อนข้างน้อย จึงไม่เหมาะสมนัก การเช่าแบบเหมาแฟ้มกดจึงเป็นปัญหาอย่างมาก และปัญหาอีกอย่างของการจัดส่งข้อความสั้นนั้นก็คือ ปัญหาของการจัดส่ง ที่มีความไม่สะดวกนักหากต้องทำการจัดส่งข้อความสั้นด้วยการใช้บริการผ่านทางเครื่องโทรศัพท์เคลื่อนที่ เนื่องจากปัญหาของการพิมพ์ที่มีความไม่สะดวกแล้วยังมีปัญหาของการจัดส่งในลักษณะของจำนวนคนที่มาก การใช้บริการจัดส่งด้วยโทรศัพท์เคลื่อนที่นั้นจึงเป็นปัญหาอย่างมากทางผู้จัดทำจึงได้ทำการพัฒนาโปรแกรมซอฟต์แวร์การกระจายข่าวสารด้วยระบบ SMS (SMS Broadcasting Program) ขึ้นมา เพื่อช่วยในการแก้ปัญหาในจุดนี้ โดยหลักการทำงานด้านการเตรียมการจัดส่งข้อความสั้นนั้น จะทำงานบนคอมพิวเตอร์ โดยผู้ใช้งานเป็นผู้ทำการเตรียมพิมพ์ข้อความที่จะทำการจัดส่งและทำการเพิ่มกลุ่มหรือเพิ่มผู้ที่ต้องการทำการส่งข้อความนั้นไปให้ โดยมีฐานข้อมูลซึ่งส่วนนี้จะเป็นหน้าที่ของ MS-SQL Server 2007 ส่วนกระบวนการของการจัดส่งนั้นจะเป็นหน้าที่หลักของบอร์ด GSM Module Wireless CPU ซึ่งจะกล่าวในบทที่ 4 ต่อไป



โปร ชอม SMS
เพียงเดือนละ 119 บาท
รับฟรี SMS 500 ข้อความ

- ค่าบริการเดือนละ 119 บาท
- โพรมิชชันรายอาทิตย์ 1.25 บาท
- ฟรี SMS 500 ครั้ง
- ส่วนเกินครั้งละ 2 บาท
- MMS ครั้งละ 3 บาท
- GPRS นาทีละ 25 สตางค์

เติมเงิน 100 บ.
ใช้ได้ 30 วัน

31 มกราคม 2554

แพ็คเกจ SMS สุดคุ้ม!



ค่าบริการ (บาท/ เดือน)	จำนวน SMS ฟรี (ครั้ง/ เดือน)	อัตราค่าส่ง SMS ตามแพ็คเกจ (บาท/ ครั้ง)	อัตราค่าส่ง SMS นอกแพ็คเกจจำนวนระบุ (บาท/ ครั้ง)	ระยะเวลา การใช้งาน	วันสมัครใช้งาน	
					SMS เบอร์ 93232	ฟรี
30	30	1	3	30 วัน	S4	*4000
50	75	0.67			S5	
75	150	0.50			S6	
10	10	1		7 วัน	S7	



แพ็คเกจ	อัตราค่าบริการ (บาท/เดือน)	อัตราส่วนเกิน
40 SMS	39 บาท	2 บาท / SMS
110 SMS	79 บาท	2 บาท / SMS
200 SMS	139 บาท	2 บาท / SMS

หมายเหตุ :- อัตราค่าบริการดังกล่าวนี้ไม่รวมภาษีมูลค่าเพิ่ม 7%
- ลูกค้าติดต่อแบบอัตโนมัติจะเริ่มต้นได้ตั้งแต่วันที่ - 30 กันยายน 2553 จะขอเวลาปรับสิทธิ์ใช้งาน 12 รอบถัด
- เมื่อสมัครแล้วจะสามารถเริ่มใช้งานได้หลังจากได้รับข้อความ SMS ยืนยัน
- สอบถามรายละเอียดเพิ่มเติม โทร. 1678 dtac call center



โปรโมชันพิเศษ
สำหรับลูกค้า TOT Smart SMS

เลือก	ราคา	ข้อความ	อายุ (วัน)
<input type="checkbox"/>	50	59	90
<input type="checkbox"/>	100	118	90
<input type="checkbox"/>	150	176	90
<input type="checkbox"/>	200	235	90
<input type="checkbox"/>	300	375	180
<input type="checkbox"/>	400	500	270
<input type="checkbox"/>	500	625	270
<input type="checkbox"/>	700	875	270
<input type="checkbox"/>	1,000	1,429	270
<input type="checkbox"/>	1,500	2,143	270
<input type="checkbox"/>	2,000	2,857	270
<input type="checkbox"/>	3,000	4,286	270
<input type="checkbox"/>	4,000	5,714	270
<input type="checkbox"/>	5,000	7,143	270
<input type="checkbox"/>	6,000	9,231	360
<input type="checkbox"/>	7,000	10,769	360
<input type="checkbox"/>	8,000	12,308	360
<input type="checkbox"/>	9,000	13,846	360
<input type="checkbox"/>	10,000	15,385	360

ตาราง 1.1 ตารางการเปรียบเทียบราคา

1.1 วัตถุประสงค์

1. เพื่อเป็นการศึกษาเกี่ยวกับการทำงานของ Wireless CPU ร่วมกับเครื่องคอมพิวเตอร์
2. เพื่อเป็นการศึกษาและพัฒนาทักษะในการใช้โปรแกรม Visual C# และ โปรแกรม MS-SQL Server 2007
3. เพื่อเป็นการนำบอร์ด GSM Module Wireless CPU มาประยุกต์ใช้กับความรู้ที่เรียนมาให้เกิดประโยชน์
4. เพื่อเป็นแนวทางสำหรับบุคคลที่สนใจในการศึกษาต่อไปในอนาคต

1.2 ขอบเขตการทำงาน

1. ใช้โปรแกรม Visual C#, AT-Command และ MS-SQL Server 2007
2. ใช้บอร์ด GSM Module Wireless CPU รุ่น Q 26 (GPRS)
3. จัดทำระบบรักษาความปลอดภัย (ระบบ Log in) และจัดทำระบบการจัดส่ง SMS
4. เขียนโปรแกรมซอฟต์แวร์และทำการทดสอบ เพื่อให้ได้ตามวัตถุประสงค์ที่วางไว้ และให้แสดงรายงานผลการส่ง SMS ได้

1.3 ขั้นตอนการดำเนินงาน

กิจกรรม	พ.ศ. 2553								
	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ธ.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.
1. ศึกษาหาข้อมูลเกี่ยวกับเรื่องที่ต้องการศึกษา และนำมาปรึกษาในกลุ่มสมาชิก และทำการเลือกหัวข้อ	↔								
2. ศึกษาหลักการการทำงานของบอร์ด GSM Module Wireless CPU รุ่น Q 26 (GPRS)		↔							

3.ศึกษาการใช้งานรูปแบบคำสั่งของ AT-Command				←→				
4.ศึกษาและค้นคว้าโปรแกรมซอฟต์แวร์ที่มีความเหมาะสมต่อการใช้งานควบคุม Wireless CPU				←→				
5.ศึกษารูปแบบของคำสั่งและโครงสร้างทางไวยากรณ์ของภาษา Visual C#					←→			
6.ทำการออกแบบและกำหนดขอบเขตของซอฟต์แวร์ ทำการเขียนโปรแกรมซอฟต์แวร์ที่กำหนดไว้ข้างต้น						←→		
7.ทดสอบโปรแกรมซอฟต์แวร์เพื่อให้เป็นไปตามวัตถุประสงค์ที่วางไว้						←→		
8.ทำการตกแต่งและออกแบบกราฟิก รวมถึงตัวติดตั้งโปรแกรมไฟล์ .exe							←→	
9.จัดทำเอกสาร รูปเล่ม การนำเสนอโครงการ								←→

1.4 ผลที่คาดว่าจะได้รับ

1. ได้โปรแกรมซอฟต์แวร์ในการจัดส่ง SMS โดยผ่านบอร์ด GSM Module Wireless CPU
2. เพื่อเป็นการพัฒนาทักษะในการเขียนโปรแกรมซอฟต์แวร์ ด้วยโปรแกรม Visual C#
3. เพื่อเป็นการช่วยลดปัญหาความยุ่งยากของการจัดส่งข้อความสั้นผ่านระบบโทรศัพท์เคลื่อนที่
4. เพื่อเป็นการลดค่าใช้จ่ายส่วนเกินของบริษัท, องค์กร หรือผู้ใช้บริการที่มีขนาดเล็ก

บทที่ 2

ระบบโทรศัพท์เคลื่อนที่ GSM และบริการข่าวสารสั้น

2.1 บทนำ

ในอดีตก่อนที่จะได้มีการพัฒนาระบบโทรคมนาคมทั้งในระบบของโทรศัพท์เคลื่อนที่ GSM และบริการข่าวสารสั้นเราจะเห็นได้ว่า หากต้องการที่จะส่งข่าวสารหรือสิ่งที่มีมนุษย์ต้องการจะสื่อสารให้กับผู้ที่อยู่ห่างไกลให้ได้รับรู้ นั้น การนำส่งข่าวสารดังกล่าวนั้นจะกระทำไปได้ด้วย “ผู้นำสาร (Messenger/Courier)” ซึ่งเป็นผู้ที่จะนำเอาข่าวสารไปส่งแก่ผู้รับ การนำเอาข่าวสารไปส่งนั้น อาจจะกระทำด้วยการเดินเท้าหรือการใช้การคมนาคมอื่นๆ จนกระทั่งมนุษย์ได้พัฒนา “สื่อ” ที่อยู่ตามธรรมชาติมาดัดแปลงให้สามารถเตรียมส่งข่าวสารให้ส่งได้รวดเร็วขึ้นบวกกับระยะทางที่อยู่ห่างไกลของผู้รับด้วยการประยุกต์ใช้การส่งแบบสัญญาณไฟ สัญญาณควัน หรือสัญญาณเสียงต่างๆ แต่ระยะทางที่สื่อสารได้ก็ยังคงมีอยู่อย่างจำกัด สื่อสารได้เพียงในระยะทางที่อยู่ใกล้ๆ เท่านั้น ต่อมา จะเห็นได้ว่าเทคโนโลยีเข้ามามีบทบาท ส่งผลทำให้ได้มีการพัฒนาการสื่อสารและการส่งข่าวสารเป็นอย่างมาก จนเกิดเป็นระบบ โทรศัพท์เคลื่อนที่และบริการข่าวสารสั้น (Short Message Service : SMS)

2.2 ระบบโทรศัพท์เคลื่อนที่ GSM

ในปัจจุบันนี้ระบบ โทรศัพท์เคลื่อนที่ที่กำลังได้รับความนิยมเป็นอย่างมากระบบหนึ่งก็คือ ระบบ GSM (Global System for Mobile communication) ซึ่งเป็นระบบโทรศัพท์แบบดิจิทัล ที่มีประสิทธิภาพสูงกว่าระบบแอนะล็อกที่ใช้อยู่เดิมหลายด้าน ได้แก่ ประสิทธิภาพในการใช้สเปกตรัม โดยสามารถรองรับจำนวนผู้ใช้ได้มากกว่า สามารถทนต่อสัญญาณรบกวนได้ดีกว่า มีความปลอดภัยสูง และยังใช้กำลังในการส่งสัญญาณน้อยกว่าอีกด้วย ระบบโทรศัพท์เคลื่อนที่ GSM ประกอบด้วยระบบย่อยๆ 4 ระบบดังนี้

1. ระบบย่อยสถานีฐาน (Base Station Sub System : BSS) ประกอบด้วย

- สถานีฐานรับ-ส่งสัญญาณ (Base Transceiver station: BTS)
- ตัวควบคุมสถานีฐาน (Base Station Controller: BSC)

2. ระบบย่อยสวิตชิง (Switching Sub System: SSS) ประกอบด้วย

- ศูนย์สวิตชิงเคลื่อนที่ (Mobile Switching Center: MSC)
- ฐานข้อมูลทะเบียนผู้ใช้ (Home Location Register: HLR)
- ฐานข้อมูลผู้มาเยือน (Visitor Location Register: VLR)
- ฐานข้อมูลตรวจสอบความถูกต้องของผู้ใช้ (Authentication Center: AC)
- ฐานข้อมูลเครื่องโทรศัพท์มือถือ (Equipment Identity Register: EIR)
- ศูนย์บริการเสริม เช่น ศูนย์บริการส่งข้อความสั้น (Short Message Service

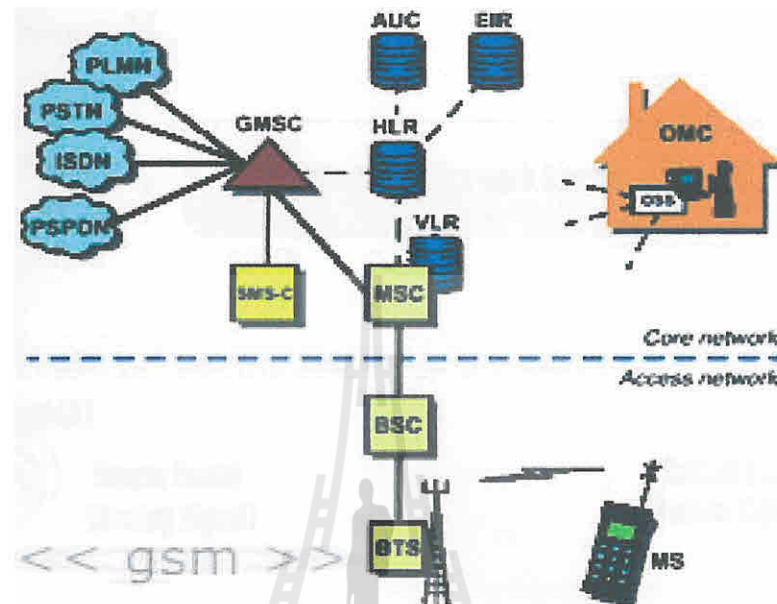
Center: SMS-C) เป็นต้น

3. ศูนย์ปฏิบัติการและบำรุงรักษา (Operation and Maintenance Center: OMC)

- ศูนย์ปฏิบัติการและบำรุงรักษาสำหรับระบบย่อยสถานีฐาน (Operation and Maintenance Center for Base Station Sub System: OMC-B)

4. สถานีเคลื่อนที่หรือ โทรศัพท์เคลื่อนที่ (Mobile Station: MS)

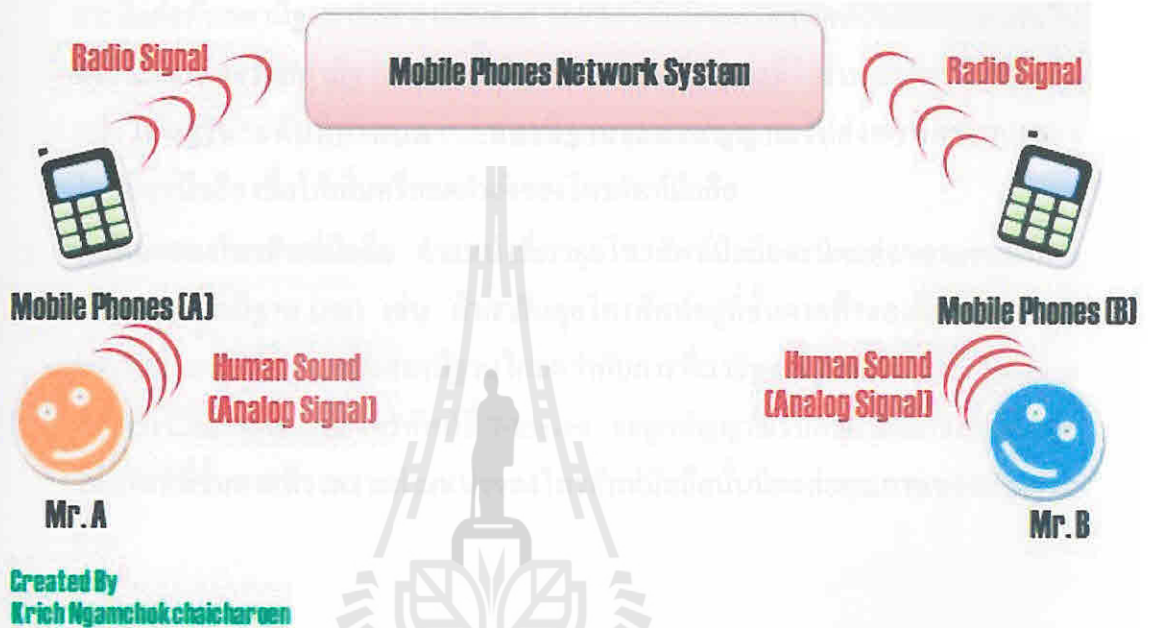
2.3 การใช้โทรศัพท์ในการติดต่อสื่อสาร (Mobile Phone for Human Communications)



รูป 2.1 แสดง รูปแบบการติดต่อสื่อสารแบบ GSM

การติดต่อสื่อสารของโทรศัพท์มือถือนั้น สามารถติดต่อกันได้ทั้งระหว่าง PLMN (Public Land Mobile Network) กับ PSTN (Public Switched Telephone Network) โดยมี Gateway MSC (GMSC) เป็นอุปกรณ์ในการติดต่อประสาน หรือติดต่อภายในระบบเครือข่าย PLMN ด้วยกันเอง ดังนั้น ถ้านาย A ทำการโทรเข้าโทรศัพท์มือถือของนาย B โดยใช้โทรศัพท์ที่บ้านของตัวเอง เมื่อ นาย A เริ่มการโทรออก ข้อมูลจากโทรศัพท์ต้นทางจะถูกนำไปใช้ในการหาที่อยู่ ณ ขณะนั้นของ โทรศัพท์ปลายทาง โดยส่งไปยัง PSTN เพื่อค้นหาเส้นทางในการเชื่อมต่อเครือข่ายปลายทาง (GMSC) และไปที่ HLR (Home Location Register) ส่วนที่เก็บข้อมูล รายละเอียด และเครือข่าย SIM ของเครื่องเอาไว้ และเชื่อมต่อไปยัง VLR (Visitors Location Register) ในแต่ละเครือข่ายว่าอยู่ใน MSC ไหน (VLR จะมีกระบวนการ Location Update เพื่อบันทึกเครื่องที่เข้ามาภายใน Location Area ของชุมสายนั้นๆ) โดยใช้ IMSI (International Mobile Subscriber Identity) รหัสเฉพาะเครื่อง ที่ได้มาจาก HLR เปรียบเทียบ หากพบว่าอยู่ที่ MSC ไหน VLR จะส่ง MSRN (Mobile Station Roaming Number) เลขหมายชั่วคราวที่กำหนดขึ้นมาเป็นเส้นทางให้กับ GMSC ใช้ในการ route ไป

ยัง MSC ปลายทาง เมื่อ route เส้นทางได้ก็จะทำการ paging แจ้งไปยังโทรศัพท์เครื่องนั้นๆ สัญญาณ paging จะถูกส่ง ออกจาก BS (Base Station) ที่ควบคุมบริเวณ Location Area นั้นอยู่ และเมื่อโทรศัพท์ปลายทางรับสายตอบรับกลับมาการเชื่อมต่อก็จะเสร็จสมบูรณ์ ทำให้นาย A สามารถติดต่อสนทนากับนาย B ได้



รูป 2.2 แสดง การสนทนาของ A และ B

เนื่องด้วยเสียงของมนุษย์ถือเป็นสัญญาณแบบอนาล็อก (Analog Signal) เมื่อเราใช้เสียงของเราพูดผ่านโทรศัพท์มือถือ เสียงของเราจะถูกแปลงผสมสัญญาณเสียงกับคลื่นพาหะ (Modulate) เพื่อลดการถูกรบกวนจากสัญญาณอื่น และขยายสัญญาณด้วยเครื่องส่ง เพื่อส่งออกอากาศด้วยคลื่นวิทยุ ไปให้สถานีฐาน (BS) จากนั้นสัญญาณวิทยุจะถูกส่งจากเครื่องข่ายไปสู่บุคคลปลายทางที่เราได้ทำการติดต่อ คลื่นวิทยุจะถูกแปลงกลับมาเป็นสัญญาณเสียงอีกครั้งหนึ่งด้วยเครื่องรับภายในโทรศัพท์ปลายทาง เพื่อให้อีกบุคคลหนึ่งสามารถรับฟังและเกิดความเข้าใจได้

ขอบเขตในการติดต่อสื่อสาร

การติดต่อของโทรศัพท์มือถือกับสถานีฐานต้องอยู่ในขอบเขต หรือพื้นที่สัญญาณที่สถานีฐานส่งไปถึง ซึ่งขอบเขตดังกล่าวมีปัจจัยร่วมด้วยกัน 2 ปัจจัย คือ

1. **กำลังส่งของโทรศัพท์มือถือ** โทรศัพท์มือถือที่มีกำลังส่งต่างกันจะมีผลทำให้ขอบเขตของการติดต่อกับสถานีฐาน (BS) ต่างกันด้วย โดยที่กำลังส่งของโทรศัพท์มือถือจะแปรผันไปตามระยะห่างจากสถานีฐาน ถ้าสถานีฐานวัดระดับสัญญาณที่ได้รับจากโทรศัพท์มือถือแล้วไม่อยู่ในระดับที่กำหนด สถานีฐานจะส่งสัญญาณไปยังส่วนควบคุมของโทรศัพท์มือถือ เพื่อให้เพิ่มหรือลดกำลังของโทรศัพท์มือถือ
2. **ตำแหน่งของโทรศัพท์มือถือ** ตำแหน่งที่เราคุยโทรศัพท์มือถือจะมีผลต่อขอบเขตในการติดต่อกับสถานีฐาน (BS) เช่น ถ้าเรายืนคุยโทรศัพท์อยู่ที่ชั้นคาเฟ่ของตึก CB4 จะมีขอบเขตในการติดต่อไปยังสถานีฐาน ไกลกว่ากับการที่เรายืนคุยโทรศัพท์อยู่ที่ 7-Eleven หน้าตึก CB4 แต่การคุยโทรศัพท์ที่ 7-Eleven จะถูกสัญญาณรบกวนน้อยกว่าการไปคุยโทรศัพท์ที่ชั้นคาเฟ่ เพราะตำแหน่งของโทรศัพท์มือถือนั้นมีผลต่อคุณภาพของสัญญาณด้วย

2.4 บริการส่งข้อความสั้น (Short Message Service: SMS)

บริการส่งข่าวสารสั้นใช้ความสามารถของระบบย่อยในระบบโทรศัพท์เคลื่อนที่ GSM ในการรับข้อความสั้นจากโทรศัพท์เคลื่อนที่ไว้พร้อมกับตรวจสอบจุดหมายปลายทางที่ต้องการจะส่งข่าวสารนั้น เมื่อพบแล้วจึงส่งข้อมูลนั้นไปยังโทรศัพท์เคลื่อนที่ปลายทาง โดยเรียกตัวกลางที่ทำหน้าที่นี้ว่า ศูนย์บริการรับฝากข่าวสาร หรือ Short Message Service Center (SMS-C) การต่อร่วมจะกระทำระหว่างชุมสายโทรศัพท์เคลื่อนที่กับ SMS-C โดยใช้ระบบการสื่อสารแบบ CCS 7 (Common Channel Signaling System No.7)

ข่าวสารสั้นจากโทรศัพท์เคลื่อนที่ GSM ไปเก็บไว้ใน SMS-C แต่ละชุดมีความยาวได้สูงสุด 160 อักขระ (160 Characters)

เมื่อข้อความสั้นถูกส่งไปยัง SMS-C จะประมวลผลข้อความสั้นเหล่านั้น เนื่องจากในตัว
ของข้อความสั้นเองได้รวมหมายเลขของโทรศัพท์เคลื่อนที่ปลายทางอยู่ อุปกรณ์ SMS-C จึงสามารถ
ติดต่อกับ HLR เพื่อตรวจสอบว่าหมายเลขที่ต้องการจะติดต่อนั้นอยู่ ณ ที่ใดในโครงข่าย HLR
จะแจ้งหมายเลขของ VLR ที่โทรศัพท์เคลื่อนที่นั้นลงทะเบียนอยู่ในปัจจุบันกลับไปยัง SMS-C ซึ่ง
SMS-C จะติดต่อไปยัง VLR นั้นๆ เพื่อให้ VLR ติดต่อกับโทรศัพท์เคลื่อนที่ปลายทางนั้นต่อไป
ในกรณีโทรศัพท์เคลื่อนที่ตอบรับการเรียกจาก VLR ศูนย์บริการ SMS-C จะส่งข้อความสั้นนั้นผ่าน
สถานีฐานไปยังโทรศัพท์เคลื่อนที่ปลายทางต่อไป

ในกรณีที่ไม่มี การตอบรับจากโทรศัพท์เคลื่อนที่ หรืออุปกรณ์ HLR พบว่า
โทรศัพท์เคลื่อนที่ปลายทางนั้นอยู่ในสถานะของการปิดเครื่อง HLR จะแจ้งกลับไปยัง SMS-C ให้
ประวิงเวลาการส่งนั้นออกไป เมื่อใดก็ตามที่โทรศัพท์เคลื่อนที่ปรากฏตัวขึ้นในโครงข่าย จะเกิด
กระบวนการ Location Update ขึ้น SMS-C จะเริ่มกระบวนการส่งข้อความสั้นไปยัง
โทรศัพท์เคลื่อนที่นั้นอีกครั้ง

ข้อความสั้นที่โทรศัพท์เคลื่อนที่ปลายทางได้รับจาก SMS-C จะถูกเก็บลงในแผ่น SIM
(Subscriber Identification Module) การรับข้อความสั้นแล้วไม่ลบออกไปเมื่ออ่านเสร็จจะทำให้
พื้นที่สำหรับเก็บข้อความสั้นใน SIM มีโอกาสเต็ม ข้อความสั้นอื่นๆ ที่ไม่สามารถเก็บลงใน SIM จะ
ได้รับการเก็บไว้ในอุปกรณ์ SMS-C ซึ่งผู้ให้บริการโทรศัพท์เคลื่อนที่บางรายได้จำกัดระยะเวลาที่
ข่าวสารสั้นสามารถคงอยู่ได้ใน SMS-C หากเกินกว่าเวลาที่ได้ตั้งไว้ข้อความสั้นเหล่านั้นก็จะถูกลบ
ไปโดยอัตโนมัติ

2.5 หลักการรับส่งข้อความสั้น หรือ SMS ของโทรศัพท์มือถือ

SMS ย่อมาจาก Short Message Service เป็นบริการส่งข้อความสั้นๆ จากโทรศัพท์เคลื่อนที่
ต้นทางผ่านชุมสายไปยังโทรศัพท์เคลื่อนที่ปลายทาง โดยสามารถส่งได้สูงสุด 160 ตัวอักษรต่อครั้ง
ตามข้อกำหนดมาตรฐานขององค์การ ETSI (European Telecommunications Standards Institute)
นอกจากนี้ยังสามารถส่งข้อความไปที่เครื่อง Fax, PC หรือ Internet address อีกด้วย

ระบบ SMS ในระบบเครือข่ายโทรศัพท์เคลื่อนที่รองรับโดยระบบ GSM (Global System for Mobile Communication), TDMA (Time Division Multiple Access) และ CDMA (Code Division Multiple Access)

เมื่อ SMS ถูกส่งจากโทรศัพท์เคลื่อนที่เครื่องหนึ่ง ข้อความนั้นจะถูกส่งไปที่ Short Message Service Center (SMSC) จากนั้นจึงจะส่งไปยังโทรศัพท์เคลื่อนที่เครื่องรับอีกทอดหนึ่ง โดยมีกระบวนการดังนี้

- 1 SMSC จะส่ง SMS Request ไปยัง Home Location Register (HLR) เพื่อหาตำแหน่งของผู้รับ
- 2 เมื่อ HLR ได้รับสัญญาณ Request ก็ส่งสถานะของผู้รับ (Subscriber's status) กลับมายัง SMSC คือ สถานะของเครื่องรับ Inactive หรือ Active, ตำแหน่งของเครื่องรับ

ถ้าสถานะของเครื่องรับเป็น Inactive แล้ว SMSC จะเก็บข้อความไว้ช่วงเวลาหนึ่ง และเมื่อใดที่เครื่องรับมีสถานะ Active แล้ว HLR จะส่ง SMS Notification ไปยัง SMSC และ SMSC ก็จะตอบรับข้อความนั้นไว้ จากนั้น SMSC จะส่งผ่านข้อความในรูปแบบ Short Message Delivery Point-to-Point ไปยังระบบบริการ โดยระบบจะทำการเรียกไปยังเครื่องรับและถ้าเครื่องรับมีการตอบรับกลับมา ข้อความก็จะถูกส่งตามไปและ SMSC จะได้รับการตอบยืนยันว่า ข้อความได้ถูกรับโดยปลายทางเรียบร้อยแล้ว หลังจากนั้นข้อความจะมีสถานะเป็น SENT และจะไม่ถูกส่งอีก

2.6 โหมดของการรับส่งข้อมูล SMS

แบ่งออกเป็น 2 โหมด คือ Text Mode และ PDU Mode (Protocol Description Unit Mode)

การส่งข้อความใน Text Mode นั้นจะเป็นการนำข้อความที่ต้องการส่งมาเข้ารหัสก่อน (โดยตัวเครื่องเอง) แล้วจึงส่งข้อมูลในรูปแบบ PDU Mode อีกครั้งหนึ่ง แต่ในบางเครื่องก็ไม่สนับสนุนการส่งแบบ Text Mode ผ่านทาง AT Command แต่หากเป็น PDU Mode จะสามารถส่งได้ นอกจากเครื่องจะไม่ต้องทำอาชีพการเปลี่ยนแปลงข้อมูลอีกชั้น

2.7 รูปแบบในการส่งข้อมูลในรูปแบบ SMS ผ่านทาง AT Command

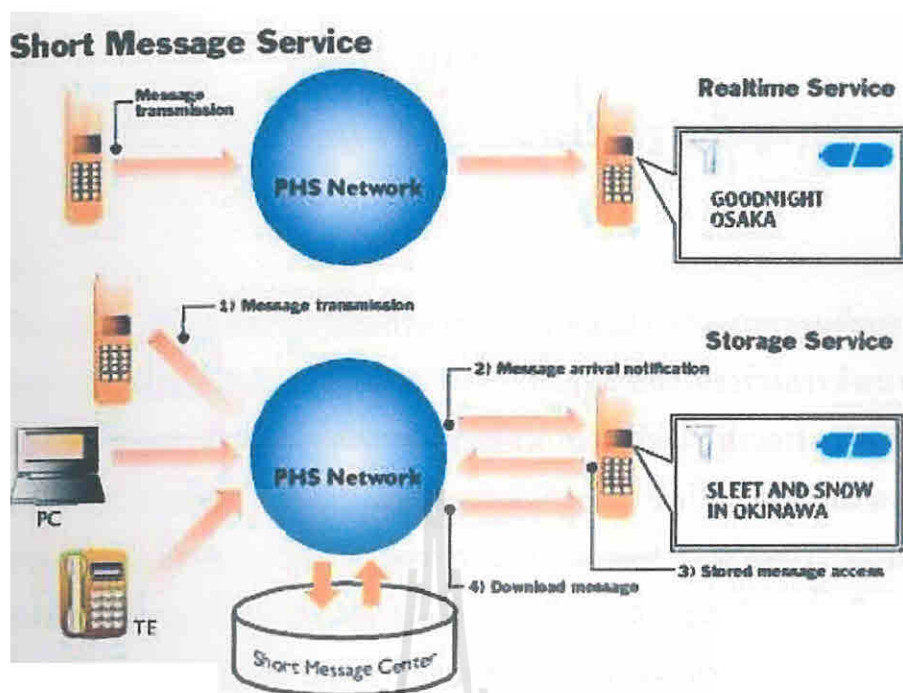
มี 2 รูปแบบ คือ Text Mode และ PDU Mode

Text Mode เป็นการส่งข้อมูลในรูปแบบของตัวอักษรได้โดยตรง ซึ่งตัวเครื่องส่วนใหญ่ไม่รองรับการส่งข้อมูลรูปแบบนี้ผ่านทาง AT Command จึงไม่สามารถใช้งานได้สมบูรณ์

PDU Mode PDU ย่อมาจาก PACKET DATA UNIT เป็นรูปแบบการส่งข้อความ SMS อีกรูปแบบหนึ่งที่ต้องมีการเข้ารหัสข้อมูลที่สลับซับซ้อนแต่ตัวเครื่องจะสามารถรับรู้ได้ทุกเครื่องที่รับคำสั่ง AT Command ได้

2.8 ข้อดีของการใช้งานบริการส่งข่าวสารสั้น ในระบบโทรศัพท์เคลื่อนที่ GSM

1. มีความเชื่อถือได้สูง เนื่องจากข่าวสารสั้นที่ถูกส่งไปจะไม่สูญหายไปจนกว่าจะถึงปลายทางในช่วงระยะเวลาหนึ่ง
2. ครอบคลุมพื้นที่การใช้งานได้เป็นบริเวณกว้าง เนื่องจากสามารถทำงานได้ในทุกพื้นที่ที่มีโครงข่ายของระบบ GSM อยู่
3. ค่าใช้จ่ายค่อนข้างถูก มีการลงทุนน้อย ค่าบริการต่อครั้งไม่แพงจนเกินไป



รูป 2.3 แสดง ขอบเขตของการส่ง SMS

	ข้อดี	ข้อด้อย
การส่งข่าวสารโดยผู้นำสาร (Messenger/Courier)	สามารถ "ส่ง" ข่าวสารได้มาก (ความจุข่าวสารมีค่าสูง) ระยะทางของการ "ส่ง" ขึ้นอยู่กับความสามารถของผู้ส่ง และระบบคมนาคมร่วมสมัย	เวลาที่ใช้ในการส่งไม่แน่นอน อาจล่าช้าหรือสูญหายได้
การ "สื่อ" ข่าวสารด้วย สิ่งประดิษฐ์ธรรมชาติ	"สื่อ" ข่าวสารได้น้อย (ความจุของข่าวสารนั้นขึ้นอยู่กับเทคนิคและความสามารถของผู้ "สื่อ")	ระยะทางที่ "สื่อ" ได้มีระยะจำกัด อาจถูกรบกวน ด้วยสิ่งแวดล้อม หรือทัศนวิสัย

ตาราง 2.1 ข้อดี - ข้อด้อยของการส่งข่าวสาร

บทที่ 3

โปรแกรม Microsoft Visual C#.Net และฐานข้อมูล SQL

3.1 บทนำ

ในบทที่ 3 นี้เราจะกล่าวถึงโปรแกรม Microsoft Visual C#.Net และฐานข้อมูล SQL ก่อนอื่นจะขอกล่าวถึงที่มาของ Net Framework ซึ่งในที่นี้จะหมายถึงโครงสร้างการพัฒนาโปรแกรมคอมพิวเตอร์ ที่ถูกออกแบบมาเพื่อให้อำนวยความสะดวกในการพัฒนาโปรแกรมสมัยใหม่ ที่ใช้งานในระบบเครือข่าย (Internet, Intranet, Mobile Devices, ฯลฯ) ซึ่งเป็นส่วนช่วยอย่างมากที่ทำให้ระบบหลาย ๆ ระบบทำงานต่อเชื่อมกันได้อย่างราบรื่น Microsoft จึงได้พัฒนารูปแบบการพัฒนา ระบบคอมพิวเตอร์ขึ้นมา ซึ่งเรียกว่า .Net Framework นั่นเอง ส่วนฐานข้อมูลนั้นเป็นการจัด เก็บข้อมูลอย่างเป็นระบบ ทำให้ผู้ใช้สามารถใช้ข้อมูลที่เกี่ยวข้องในระบบงานต่าง ๆ ร่วมกันได้ โดยที่ จะไม่เกิดความซ้ำซ้อนของข้อมูล และยังสามารถหลีกเลี่ยงความขัดแย้งของข้อมูลด้วย อีกทั้งข้อมูล ในระบบก็จะต้องเชื่อถือได้ และเป็นมาตรฐานเดียวกัน โดยจะมีการกำหนดระบบความปลอดภัย ของข้อมูลขึ้น โปรแกรมฐานข้อมูลนั้นมีอยู่หลากหลาย ทางคณะผู้จัดทำได้เลือกใช้คือ โปรแกรม SQL ซึ่งเป็นโปรแกรมฐานข้อมูลที่มีโครงสร้างของภาษาที่เข้าใจง่าย ไม่ซับซ้อน มีประสิทธิภาพ การทำงานสูง สามารถทำงานที่ซับซ้อนได้โดยใช้คำสั่งเพียงไม่กี่คำสั่ง โปรแกรม SQL จึงเหมาะที่ จะใช้กับระบบฐานข้อมูลเชิงสัมพันธ์ และเป็นภาษาหนึ่งที่มีผู้นิยมใช้กันมาก

3.2 ทำความรู้จักกับภาษา Visual C#

Visual C# เป็นภาษาที่ถูกออกแบบมาเพื่อรองรับการทำงานในยุค .NET โดยมีแนวภาษา เป็นแบบการเขียนโปรแกรมเชิงวัตถุสมัยใหม่ (Object-oriented programming, OOP) ซึ่งมีความ สามารถเชิงวัตถุใกล้เคียงกับ C++ แต่มีความง่ายในการเขียนโปรแกรมเหมือนกับ Visual Basic

C# เป็นภาษาใหม่ที่ทาง ไมโครซอฟต์ได้พัฒนาขึ้นมาพร้อมกับ โปรแกรมชุด Visual Studio.NET เป็นภาษาที่พัฒนาขึ้นมาเพื่อรองรับการทำงานบน .NET Platform โดยมีแนวทางของ การเขียน โปรแกรมเชิงวัตถุสมัยใหม่หรือเรียกสั้นๆ ว่า Modern OOP (Modern Objected Oriented Programming)

ในการเขียน โปรแกรมด้วย ภาษา C# นั้นได้รับอิทธิพลมาจากภาษา C++ จึงมีรูปแบบคำสั่ง ที่คล้ายคลึงกัน และคล้ายกับ JAVA เพราะ JAVA ได้รับอิทธิพลมาจากภาษา C++ โดยที่คำสั่งนั้น ถูกกำหนดโดยบล็อก ที่ครอบด้วยเครื่องหมาย { กับ }

ภายในบล็อกจะประกอบด้วยคำสั่ง ซึ่งแต่ละคำสั่งแยกกัน หรือสิ้นสุดลง ด้วยเครื่องหมาย ;
ซึ่งคำสั่งหนึ่งๆ ไม่จำเป็นต้องเขียนจบในบรรทัดเดียวกันก็ได้ เช่น

```
private void button1_Click(object sender, EventArgs e)
{
    MessageBox.Show(textBox1.Text,"สวัสดีครับ");
}
private void button2_Click(object sender, EventArgs e)
{
    DialogResult dr;
    dr = MessageBox.Show("คุณต้องการจบการทำงานหรือไม่", "Exit ?",
    MessageBoxButtons.YesNo, MessageBoxIcon.Question);
    if (dr == DialogResult.Yes)
    {
        MessageBox.Show("จบการทำงานครับ", "End", MessageBoxButtons.OK,
    MessageBoxIcon.Warning);
        this.Close();
    }
    else
        MessageBox.Show("คุณยังคงทำงานต่อไป", "Continue", MessageBoxButtons.OK,
    MessageBoxIcon.Information);
}
```

3.3 ข้อสังเกตในการเขียนโปรแกรมด้วยภาษา Visual C#.NET

การเขียนประโยคคำสั่ง เพื่อควบคุมการทำงานของโปรแกรมด้วยภาษา Visual C#.NET จำเป็นต้องเขียนให้ถูกต้องตามโครงสร้าง และไวยากรณ์ของภาษาอย่างเคร่งครัด เช่น คำสำคัญ เครื่องหมาย เป็นต้น หากผิดไปจากโครงสร้าง และไวยากรณ์ก็จะเกิดข้อผิดพลาด ไม่สามารถสั่งการให้คอมพิวเตอร์ทำงานได้ ในเบื้องต้นมีข้อสังเกตที่สำคัญดังนี้

- 1 คำสั่งต่างๆ จะเป็นตัวพิมพ์ตัวเล็กทั้งหมด ยกเว้นการเรียกใช้กลุ่มออบเจ็กต์ต่างๆ เท่านั้นที่อาจมีตัวพิมพ์ตัวใหญ่ปะปนอยู่ด้วย เช่น listBox 1, textBox1, MessageBox เป็นต้น

- 2 ตัวพิมพ์เล็ก ตัวพิมพ์ใหญ่มีความหมายต่างกัน เป็นคนละตัวกัน เช่น ตัวแปรชื่อ studentName เป็นคนละตัวกับ StudentName เป็นต้น
- 3 เมื่อเขียนคำสั่งจบประโยค ไม่ว่าจะประโยคจะสั้น หรือยาวมากกว่าหนึ่งบรรทัด ต้องเขียนเครื่องหมาย ; (เซมิโคลอน) ไว้ท้ายประโยคเสมอ
- 4 ชนิดของข้อมูลมีความสำคัญมาก โปรแกรมเมอร์จำเป็นต้องระบุให้ชัดเจน ตัวเลขที่อยู่ในสถานะข้อความ กับตัวเลขที่อยู่ในสถานะจำนวนเต็ม มีความต่างกัน เช่น ตัวเลขที่อยู่ในสถานะข้อความจะไม่สามารถเรียกไปใช้ในการคำนวณได้ เป็นต้น
- 5 ช่องว่างที่เกิดจากการย่อหน้า หรือขึ้นบรรทัดใหม่ มีผลต่อความเป็นระเบียบ อ่านง่าย ของชุดคำสั่งเท่านั้น ไม่มีผลต่อการทำงาน

3.4 บล็อกของคำสั่ง (Block)

ภาษา Visual C#.NET ใช้เครื่องหมาย { (ปีกกาเปิด) และ เครื่องหมาย } (ปีกกาปิด) เป็นสัญลักษณ์บล็อก หรือขอบเขตของคำสั่ง โดยเครื่องหมายปีกกาเปิดทำหน้าที่เปิดคำสั่ง และปีกกาปิดทำหน้าที่ปิดคำสั่ง ชุดของคำสั่งจะทำงานอยู่ในบล็อกเท่านั้น และ บล็อกของคำสั่งสามารถมีได้หลายๆบล็อก ซ้อนกัน ตัวอย่างเช่น

```
Private void button1_Click ()
{
    ประโยคคำสั่ง A ;
    If (เงื่อนไข)
    {
        ประโยคคำสั่ง B ;
    }
}
```

3.5 ชนิดของข้อมูล (Data Type)

การเขียนโปรแกรมคอมพิวเตอร์ โปรแกรมเมอร์จำเป็นต้องมีความรู้ ความเข้าใจถึงชนิดของข้อมูลต่างๆ ก่อนเสมอ เนื่องจาก เมื่อโปรแกรมทำงาน ต้องใช้หน่วยความจำเก็บข้อมูล ถ้าโปรแกรมเมอร์ประกาศประเภทของข้อมูลในตัวแปรผิดประเภท หรือไม่เหมาะสมกับประเภทของการใช้งาน อาจทำให้เกิดผลเสีย เช่น

- o โปรแกรมทำงานผิดพลาด

- โปรแกรมใช้ทรัพยากรของเครื่องคอมพิวเตอร์มากไป
 - อาจทำให้หน่วยความจำรั่ว
- ดังนั้นการทำความเข้าใจถึงชนิดของข้อมูล จึงเป็นสิ่งจำเป็น

ชนิดของข้อมูล	ขอบเขต	ขนาด	รายละเอียด
bool	True หรือ false	1 ไบต์	เก็บค่าทางตรรกะ จริง หรือเท็จ
char		2 ไบต์	เก็บตัวอักษร
string		ไม่ระบุ	เก็บข้อความ
int	- 2,147,483,648 ถึง 2,147,483,647	4 ไบต์	เก็บเลขจำนวนเต็ม
unit	0 ถึง 4,294,967,295	4 ไบต์	เก็บเลขจำนวนเต็มบวก
long	- 9,223,372,036,854,775,808 ถึง 9,223,372,036,854,775,807	8 ไบต์	เก็บเลขจำนวนเต็ม
ulong	0 ถึง 18,446,744,073,709,551,615	8 ไบต์	เก็บเลขจำนวนเต็มบวก
short	-32,768 ถึง 32,767	2 ไบต์	เก็บเลขจำนวนเต็ม
ushort	0 ถึง 65,535	2 ไบต์	เก็บเลขจำนวนเต็มบวก
float		4 ไบต์	เก็บเลขจำนวนจริง ทศนิยม 7 ตำแหน่ง
double		8 ไบต์	เก็บเลขจำนวนจริง ทศนิยม 15 ตำแหน่ง

ตาราง 3.1 ตารางแสดงประเภทของข้อมูล

3.6 ตัวแปร (Variable)

ตัวแปรหมายถึงพื้นที่ว่างในหน่วยความจำของคอมพิวเตอร์ สำหรับเก็บข้อมูล เพื่อเรียกมาใช้งาน หรือ ทำงานในการเขียน โปรแกรม

การสร้างตัวแปรหมายถึงการจองพื้นที่ของหน่วยความจำ การสร้างตัวแปร โปรแกรมเมอร์จะต้องระบุชนิดของข้อมูลให้ชัดเจน และต้องตั้งชื่อตัวแปรให้ถูกต้อง และไม่ซ้ำกัน การตั้งชื่อตัวแปร มีกฎ ดังนี้

- ต้องขึ้นต้นด้วยตัวอักษร

- ห้ามเว้นวรรค
- ห้ามใช้สัญลักษณ์อื่น นอกจาก _ (under score)
- อักษรตัวใหญ่ กับอักษรตัวเล็กมีความหมายต่างกัน
- ไม่ตรงกับคำสงวน

รูปแบบไวยากรณ์ของการสร้างตัวแปรมีดังนี้

```
DataType VariableName ;
```

โดย DataType หมายถึงชนิดของข้อมูล เช่น int , float , double , char , string เป็นต้น

VariableName หมายถึงชื่อของตัวแปร ซึ่งต้องตั้งชื่อตามกฎ ดังที่กล่าวแล้วข้างต้น ตัวอย่างเช่น

```
string employeeName ; // หมายถึงการสร้างตัวแปรเก็บข้อมูลข้อความ ชื่อ
```

employeeName

```
int age ; // หมายถึงการสร้างตัวแปรเก็บข้อมูลเลขจำนวนเต็ม ชื่อ age
```

```
float salary ; // หมายถึงการสร้างตัวแปรเก็บข้อมูลเลขจำนวนจริง ชื่อ salary
```

การใส่ค่าให้กับตัวแปรทำได้โดยใช้เครื่องหมายดำเนินการใส่ค่าให้ตัวแปร ซึ่งในภาษา Visual C#.NET ใช้เครื่องหมายเท่ากับ (=) ตัวอย่างเช่น เมื่อต้องการใส่ค่าให้กับตัวแปรข้างต้น ทำได้โดยเรียกชื่อตัวแปรที่ต้องการใส่ค่า โดยไม่ต้องระบุชนิดของข้อมูลอีก ตามด้วยเครื่องหมายดำเนินการ และตามด้วย ค่าที่ต้องการใส่ ตัวอย่างเช่น

```
employeeName = "Somchai Rugdee" ;
```

```
age = 25 ;
```

```
salary = 25400.57 ;
```

3.7 เครื่องหมายดำเนินการ (Operator)

เครื่องหมายดำเนินการหมายถึงเครื่องหมายที่เขียนระหว่าง ตัวแปร ทำหน้าที่ดำเนินการในลักษณะต่างๆ กันออกไป โดยจัดกลุ่มเครื่องหมายดำเนินการได้ดังนี้

3.7.1 เครื่องหมายดำเนินการทางคณิตศาสตร์ (Mathematic Operation)

Operator	ตัวอย่าง	ความหมาย
+	a = b+c;	a เท่ากับผลลัพท์ของ b บวกกับ c
-	a = b-c;	a เท่ากับผลลัพท์ของ b ลบกับ c
*	a = b*c;	a เท่ากับผลลัพท์ของ b คูณกับ c
/	a = b/c;	a เท่ากับผลลัพท์ของ b หารกับ c

%	a = b%c;	a เท่ากับผลลัพธ์ของเศษของ b หารกับ c
++	a = ++b;	a เท่ากับผลลัพธ์ของ b+1 b ถูกเพิ่มค่าอีก 1
--	a = --b;	a เท่ากับผลลัพธ์ของ b-1 b ถูกลดค่าอีก 1

ตาราง 3.2 ตารางแสดงเครื่องหมายดำเนินการทางคณิตศาสตร์

3.7.2 เครื่องหมายดำเนินการกำหนดค่าให้กับตัวแปร (Assignment Operator)

การกำหนดค่าให้กับตัวแปร ดำเนินการ โดยนำค่าทางด้านขวาของเครื่องหมายดำเนินการ

ใส่ให้กับตัวแปรทางด้านซ้าย

Operator	ตัวอย่าง	ความหมาย
=	a = b;	a เท่ากับผลลัพธ์ของ b
+=	a += b;	a เท่ากับผลลัพธ์ของ a บวก b
-=	a -= b;	a เท่ากับผลลัพธ์ของ a ลบ b
*=	a *= b;	a เท่ากับผลลัพธ์ของ a คูณ b
/=	a /= b;	a เท่ากับผลลัพธ์ของ a หาร b
%=	a %= b;	a เท่ากับผลลัพธ์ของเศษของ a หาร b

ตาราง 3.3 ตารางแสดงเครื่องหมายดำเนินการทางคณิตศาสตร์

3.7.3 เครื่องหมายดำเนินการเปรียบเทียบค่าของตัวแปร (Rational Operator)

เครื่องหมายดำเนินการเปรียบเทียบค่าของตัวแปร จะมีผลลัพธ์เพียงสองอย่างเท่านั้น คือ

ถูก (True) หรือผิด (false)

<	น้อยกว่า
>	มากกว่า
==	เท่ากัน
!=	ไม่เท่ากัน
<=	น้อยกว่าหรือเท่ากัน
>=	มากกว่าหรือเท่ากัน

3.8 การเขียนหมายเหตุ (Comment)

หมายเหตุหมายถึงบันทึกที่โปรแกรมเมอร์เขียนแทรกไว้ในโปรแกรม ไม่มีผลกับการทำงานของโปรแกรม เพราะหมายเหตุไม่ใช่ชุดคำสั่งให้โปรแกรมทำงาน การเขียนหมายเหตุมีเหตุผลสำคัญ 2 ประการ คือ

1) อธิบายการทำงานของชุดคำสั่ง ช่วยให้โปรแกรมเมอร์ หรือผู้อื่นเข้าใจการทำงานของชุดคำสั่งได้ในภายหลัง

2) กำหนดให้ชุดคำสั่งนั้น ไม่มีผลต่อการทำงาน อาจทำเพื่อทดสอบชุดคำสั่ง หรือทำการตรวจค่าของตัวแปร

การเขียนหมายเหตุสามารถทำได้ 2 ลักษณะ คือ

1. ใช้เครื่องหมาย // นำหน้าบรรทัดที่ต้องการทำหมายเหตุ ส่งผลทำให้ชุดคำสั่งบรรทัดนั้นไม่ทำงาน หรือมีผลทำให้โปรแกรมเมอร์สามารถพิมพ์ข้อความบันทึกหมายเหตุได้ด้วยภาษาเขียนปกติโดยไม่ทำให้เกิดข้อผิดพลาด

2. ใช้เครื่องหมาย /* */ ครอบระหว่างบรรทัด ในกรณีที่ต้องการทำหมายเหตุหลายๆบรรทัด

ตัวอย่างการเขียนหมายเหตุ

// Program Name : Fahrenheit Convert to degree Celsius and Kelvins

// Programmer Name : Mr.Somchai Rugdee

using system

3.9 รู้จักภาษา SQL

SQL ย่อมาจาก Structured Query Language เป็นภาษาที่ใช้ร่วมกับระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (RDBMS: Relational Database Management System)

3.9.1 คำสั่ง Select

เป็นคำสั่งที่ใช้สอบถาม หรือคัดเลือกข้อมูล (Query) จากหนึ่งตาราง หรือหลายตารางที่มีความสัมพันธ์กัน โดยกำหนดเงื่อนไขเพื่อค้นหาข้อมูลที่ต้องการ รวมทั้งสามารถเรียงลำดับ และจัดกลุ่มข้อมูล

รูปแบบ

Select [predicate] * |fieldname1,fieldname2,.....

From tablename

Where criteria

[Order By *fieldname1* [ASC | Desc] , *fieldname2* [ASC | Desc],...]

อธิบาย

predicate หมายถึง คำสั่งย่อย TOP, DISTINCT ใช้กำหนดจำนวน Record หรือรูปแบบการแสดงผล

fieldname1,fieldname2,.. หมายถึง ชื่อฟิลด์ที่จะเลือกนำมาแสดงข้อมูล

tablename หมายถึง ชื่อตารางข้อมูลที่จะเลือกนำมาแสดงข้อมูล

criteria หมายถึง เงื่อนไขที่ใช้ Query จากตาราง

ตัวอย่างการใช้คำสั่ง Selectแบบไม่มีเงื่อนไข

Personal_ID	Civil_ID	MOD_ID	Status	Prefix_ID	Name	Surname
5	3100500376534	1132300771	0	12	กิตติคุณ	ทองวีดิ
7		NULL	0	6	ฉวีราพร	ทองวีดิ
9	3600101025288	1155100883	0	9	ชัยณรงค์	ธนาคุณ
10	3659900154028	1122300422	0	12	ปรีชา	เหิมวีไล
12	3520101513551	1181301139	0	11	อุทัย	ชัชชนะ
13	3190900304549	1286101336	0	12	ทวีศักดิ์	หำยวงศ์
14	3650101148321	1353800482	0	17	สมเกียรติ	เกษเทศ
15	3650101159446	1210500626	0	19	สมนึก	สุราแก้ว
16	3640400384875	1416300438	0	20	สุทิน	ทองเอ็ม
17	3659900640895	1273800674	0	17	ฉัตรชัย	ฉิมเอ็ม
18	3650100361578	1213800594	0	17	ประยุทธ	บุรินทร์
19	3659900101269	1173800017	0	17	แพตล	บัวผืน
20	3102002560149	1291300955	0	12	สาทร	เดชะวงศ์
21	1509900911516	NULL	0	26	ศิริกัลยา	เดชะวงศ์
22	3129900268273	NULL	0	6	มนต์ฉาย	เดชะวงศ์
23	3139900057531	1302900674	0	12	อรรอง	ทิพาปกรณ์
24		NULL	0	6	วรากุล	ทิพาปกรณ์
25		NULL	0	26	ชนากานต์	ทิพาปกรณ์
26		NULL	0	25	ปกรณ์	ทิพาปกรณ์
27	3570100673236	1281201116	0	12	วรเทพ	บุญณะ
28	3571200153405	1401201242	0	20	ฉัฐกันท์	ทิวาศา
29	3650100644880	1153700027	0	35	ถวัล	สิงห์
30	3520100385311	1335101396	0	13	ธงชัย	วรรณसार
31	5471200091914	1361201696	0	20	ณรงค์	รักขัน
32	5570700032262	1271200675	0	17	มานะ	คำเงิน
33	5570700032262	1271200675	0	17	มานะ	คำเงิน

รูป 3.1 แสดงตัวอย่างข้อมูลจาก Table Personal

3.9.1.1 แสดงข้อมูลจากทุก field ของตาราง Personal

คำสั่ง *Select * from Personal*

The screenshot shows a SQL query window with the following query: `select * from personal`. The results pane displays a table with 16 rows of data. The columns are: Personal_ID, Civil_ID, MOD_ID, Status, Prefix_ID, Name, Surname, Sex, BirthDate, and MaritalStatus. The status of all records is 0, and the marital status is either 2 or NULL.

Personal_ID	Civil_ID	MOD_ID	Status	Prefix_ID	Name	Surname	Sex	BirthDate	MaritalStatus
6	3100500376534	1132300771	0	12	กิตติคุณ	ทองวิจิต	1	1952-10-03 00:00:00.000	2
7		NULL	0	6	ณัฐพร	ทองวิจิต	2	2007-07-17 00:00:00.000	NULL
9	3600101025288	1155100883	0	9	ชัยณรงค์	รณางค์	1	1953-01-10 00:00:00.000	2
10	3659900154028	1122300422	0	12	ปรีชา	เทียมวิไล	1	1950-06-06 00:00:00.000	2
12	3520101513551	1181301139	0	11	จุฑิยา	ชัยชนะ	1	1957-06-16 00:00:00.000	2
13	3190900304549	1286101336	0	12	ทวีศักดิ์	พจ่างวงศ์	1	1965-12-04 00:00:00.000	2
14	3650101148321	1353800482	0	17	สมเกียรติ	เกษุเทศ	1	1974-12-03 00:00:00.000	1
15	3650101159446	1210500626	0	19	สมนึก	จุรานแก้ว	1	1957-06-23 00:00:00.000	2
16	3640400384875	1416300438	0	20	สุทิน	ทองเริ่ม	1	1977-01-01 00:00:00.000	2
17	3659900640895	1273800674	0	17	ฉัตรชัย	ฉิมเยี่ยม	1	1963-01-04 00:00:00.000	2
18	3650100361578	1213800594	0	17	ประสุทร	รุ่งเรือง	1	1957-03-04 00:00:00.000	2
19	3659900101269	1173800017	0	17	นพพล	บัวผัน	1	1953-07-18 00:00:00.000	2
20	3102002560149	1291300955	0	12	สาทร	เตชะวงศ์	1	1954-09-23 00:00:00.000	2
21	1509900911516	NULL	0	26	ศิริกัญญา	เตชะวงศ์	2	1990-09-08 00:00:00.000	NULL
22	3129900268273	NULL	0	6	มณีฉาย	เตชะวงศ์	2	1943-08-31 00:00:00.000	NULL
23	3139900057531	1302900674	0	12	ยรรยง	ทิพาปก...	1	1969-10-25 00:00:00.000	2

Query executed successfully. drdo (3.0 RTM) sa (55) DRD02 00:00:01 9423 rows

รูป 3.2 แสดงตัวอย่างข้อมูลจากคำสั่ง *Select * from Personal*

3.9.1.2 ใช้คำสั่ง TOP [n] แสดงข้อมูลเฉพาะ 5 Record แรกของตาราง Personal

คำสั่ง *Select Top 5 * from Personal*

The screenshot shows a SQL query window with the following query: `select Top 5 * from personal`. The results pane displays the following data:

	Personal_ID	Civil_ID	MOD_ID	Status	Prefix_ID	Name	Surname	Sex	BirthDate	MaritalStatus
1	6	3100500376534	1132300771	0	12	กิตติคุณ	ทลงวิธิต	1	1952-10-03 00:00:00.000	2
2	7		NULL	0	6	พัฏฐาพร	ทลงวิธิต	2	2007-07-17 00:00:00.000	NULL
3	9	3600101025288	1155100883	0	9	ชัยณรงค์	รนาคุณ	1	1953-01-10 00:00:00.000	2
4	10	3659900154028	1122300422	0	12	ปรีชา	เจียมวิไล	1	1950-06-06 00:00:00.000	2
5	12	3520101513551	1181301139	0	11	สุทัย	ชัยชนะ	1	1957-06-16 00:00:00.000	2

Query executed successfully. drdo (9.0 RTM) sa (55) DRD02 00:00:00 5 rows

รูป 3.3 แสดงตัวอย่างข้อมูลจากคำสั่ง *Select Top 5 * from Personal*

3.9.1.3 แสดงข้อมูลเฉพาะ Field Name, Surname และ MOD_ID และให้เชื่อมข้อมูล Name และ Surname โดยใช้เครื่องหมาย + และตั้งชื่อ Field นี้ว่า “ชื่อ” โดยกำหนดไว้หลังคำสั่ง as
คำสั่ง *Select Name + ' ' + Surname as ชื่อ, MOD_ID from personal*

	ชื่อ	MOD_ID
1	กิตติคุณ ทอกรวิธิต	1132300771
2	ฉวีธราพร ทอกรวิธิต	NULL
3	ชัยณรงค์ ธนาธรม	1155100883
4	ปวีณา เทียมศรีใจ	1122300422
5	สุทธิชัย ชัยชนะ	1181301139
6	ทวีศักดิ์ พงษ์วงษ์	1286101336
7	สมเกียรติ เกตุเทศ	1353800482
8	สมนึก อจาแก้ว	1210500626
9	สุทิน ทองเริ่ม	1416300438
10	ฉัตรชัย ฉิมเริ่ม	1273800674
11	ประยุทธ ทุ่งพุ่ม	1213800594
12	นพดล บัวสัน	1173800017
13	สาทร เตชะวงศ์	1291300955
14	ศิริกัญญา เตชะวงศ์	NULL
15	มณีฉาย เตชะวงศ์	NULL
16	ยรรยง กิฬาปกรณ์	1302900674
17	วรากุล กิฬาปกรณ์	NULL

รูป 3.4 แสดงตัวอย่างข้อมูลจากคำสั่ง *Select Name + ' ' + Surname as ชื่อ, MOD_ID from personal*

บทที่ 4

ข้อมูลของฮาร์ดแวร์ในโครงการ

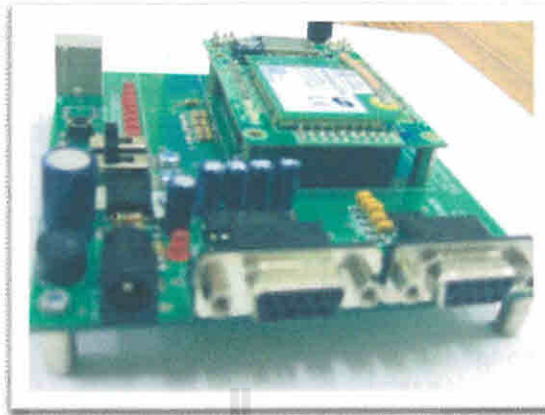
4.1 บทนำ

ในบทที่ 4 จะกล่าวถึงส่วนประกอบของข้อมูลฮาร์ดแวร์เป็นหลัก จากโครงการนี้เราจะมีส่วนประกอบของฮาร์ดแวร์คือ ชุดอุปกรณ์บอร์ด GSM Module Wireless CPU และ SIM Card โทรศัพท์เคลื่อนที่ บอร์ด GSM Module Wireless CPU เป็นชุดอุปกรณ์ที่มีขายอยู่ตามท้องตลาดโดยทั่วไปอยู่แล้ว ในเรื่องของราคาอุปกรณ์นั้นก็ขึ้นอยู่กับการที่เราจะเลือกใช้งานว่าจะนำชุดบอร์ดนั้นมาใช้งานเกี่ยวกับอะไร และทางผู้จัดทำได้ทำการแสดงให้เห็นถึงส่วนประกอบที่อยู่ภายในของส่วนที่ต้องใช้งานจริง ส่วน SIM Card โทรศัพท์เคลื่อนที่นั้น ย่อมเป็นที่รู้จักกันดีในยุคปัจจุบัน เพราะมีการใช้งานอย่างแพร่หลาย ส่วนการทำงานในส่วนของซอฟต์แวร์นั้นเราจะกล่าวในบทต่อไป

4.2 ภาพรวมของอุปกรณ์ต้นแบบ



รูป 4.1 ภาพมุมมองบอร์ดจากด้านบน



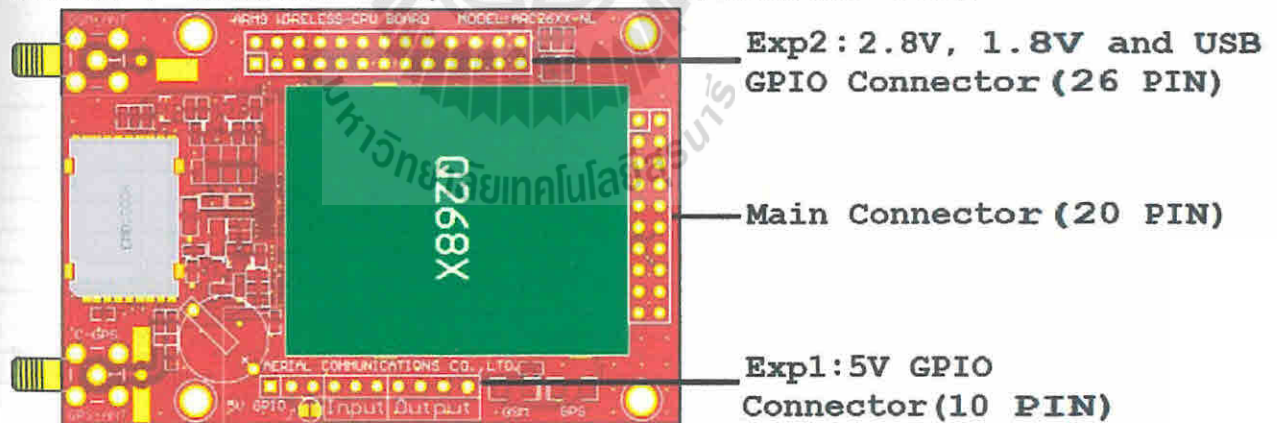
รูป 4.2 ภาพมุมมองบอร์ดจากด้านข้าง

4.3 ชุดบอร์ด GSM Module Wireless CPU

PIN Descriptions

Product : ARM9 WIRELESS-CPU BOARD

MODEL : ARC2686-NLO/ARC2687-NLO/ARC2686-CGPS



MAIN CONNECTOR (20PIN)

Pin Number	Assignment	I/O	I/O Type	Descriptions
1	BOOT			Reserved use by Aerial (Pull down with 100KOhm)
2	RESET	I	CMOS	Reset GSM Module (Active High)
3	TX	I	5V	UART-Transmit (TTL)
4	RTS	I	5V	UART-Request to send (TTL)

5	RI	O	5V	UART-Ring Indicator (TTL)
6	DCD	O	5V	UART-Data Carrier Detect (TTL)
7	RX	O	5V	UART- Receive (TTL)
8	CTS	O	5V	UART-Clear to send (TTL)
9	SPK P	O	Analog	Speaker Positive Output
10	SPK N	O	Analog	Speaker Negative Output
11	MIC P	I	Analog	Microphone Positive Input
12	MIC N	I	Analog	Microphone Negative Input
13	PWR	I	CMOS	Power On/Off Module (Logic High > 1 sec)
14	VCC_2V8	O	2V8	+2.8v Digital Supply (Use for check module in power on state)
15	DTR	I	5V	UART-DSR (TTL)
16	DSR		5V	UART-DSR (TTL)
17	+5V	I	Power Supply	Power Supply Input (High Current)
18	+5V	I	Power Supply	Power Supply Input (High Current)
19	GND		Supply	
20	GND		Supply	

EXP1 : 5V GPIO CONNECTOR (10PIN)

Pin Number	Assignment		I/O	I/O Type	Descriptions	
	NLO	C-GPS			NLO	C-GPS
1	N/A	GPS FIX	O	Open Drain	Not Available	GPS Fixed indicator
2	N/A	WD_CHECK	I	CMOS	Not Available	Pull down to disable watchdog
3	N/A	N/A	-	-	Not Available	
4		GPIO44	I	5V	Digital Input	
5		GPIO24	I	5V	Digital Input	
6		GPIO21	I	5V	Digital Input	
7		GPIO11	O	5V	Digital Output	
8		GPIO10	O	5V	Digital Output	
9		FLASH_LED	O	Open Drain	GSM/GPRS LED Status	
10		GPIO9	O	5V	Digital Output	

ข้อควรระวัง!

I/O ชุดนี้จะถูกเชื่อมต่อเข้า โดยตรงกับ ARM CPU ซึ่งมีระดับของสัญญาณที่เชื่อมต่อตามที่กำหนดไว้ที่

ช่อง

I/O Type การเชื่อมต่อด้วยระดับแรงดันไม่ถูกต้อง อาจทำให้ Wireless CPU เสียหายได้

4.4 โครงสร้าง SIM Card โทรศัพท์เคลื่อนที่

นับตั้งแต่โทรศัพท์มือถือระบบจีเอสเอ็ม หรือระบบที่เรารู้จักกันดีในทุกวันนี้ ส่วนประกอบที่สำคัญที่ทำให้เครื่องโทรศัพท์เคลื่อนที่สามารรถโทรออกได้ก็คือ ซิมการ์ด (SIM Card) ซึ่งซิมการ์ดขนาดมาตรฐานที่เราซื้อหามาใช้ หรือบางเจ้าแจกให้ทดลองใช้กันฟรีๆ ซิมการ์ดนั้นมีขนาดเท่ากันทุกค่าย วัสดุที่ใช้ทำเป็นพลาสติกเหนียว ขึ้นรูปออกมาเป็นแผ่นขนาดเท่านามบัตร หรือบัตรเอทีเอ็ม โดยมีความยาว 85.5 มม. กว้าง 53.9 มม. และมีความหนาเพียง 0.85 มม.



รูป 4.3 ลักษณะบรรจุภัณฑ์

ในอดีตนั้นจะเห็นได้ว่าโทรศัพท์เคลื่อนที่ในระบบจีเอสเอ็มรุ่นแรกๆที่ถือกำเนิดมาจะใส่ซิมการ์ดเข้าไปเข้าไปทั้งใบ เพราะสมัยนั้นตัวเครื่องยังมีขนาดใหญ่ แต่สมัยนี้ต้องแกะตัวซิมการ์ดขนาดเล็กออกมาจากแผ่น ซึ่งแผงวงจรทั้งหมดนี้มีขนาดที่เล็กมาก แต่ก็ยังมีขอบบางส่วนเป็นพลาสติก มีขนาด

มม. กว้าง 25 มม. สมัยก่อนถือว่ามีขนาดเล็กมาก แต่พอมาถึงสมัยนี้แล้วถือว่าขนาดค่อนข้างใหญ่ เมื่อเทียบกับขนาดของการ์ดหน่วยความจำแบบไมโครเอสดีที่ใช้กันทั่วไป



รูป 4.4 ลักษณะ SIM Card

ปัจจุบันซิมการ์ดของผู้ให้บริการบางรายมีขนาดกรอบเพียงครึ่งเดียว แต่ตัวซิมการ์ดยังมีเท่าเดิม อาจจะด้วยเหตุผลที่ต้องการทำให้แพ็คเกจใช้กระดาษน้อยลง ใช้พลาสติกน้อยลง และช่วยลดโลกร้อนไปด้วยในตัว

4.4.1 หน่วยความจำของซิมการ์ด

ความจุของซิมการ์ดนั้นขึ้นอยู่กับรุ่นที่ผู้ให้บริการนำเข้ามาจำหน่าย แต่ส่วนมากแล้วจะมีความจุ 16 และ 32 กิโลไบต์ สังเกตได้จากด้านหลังซิมการ์ดจะมีเลข 16k หรือ 32k กำกับอยู่ ซึ่งบางรุ่นก็ไม่มีบอกให้ทราบ สำหรับพื้นที่หน่วยความจำหลักๆ แล้วก็มีไว้สำหรับบรรจุข้อมูลสมุดโทรศัพท์ ข้อความ

เอสเอ็มเอส ซึ่งบางรุ่นก็มีความจุแตกต่างกันไปอีกเช่นกัน มีตั้งแต่ 200-1000 รายชื่อ และเก็บข้อความเอสเอ็มเอสได้อีกตั้งแต่ 15-30 ข้อความ โดยสามารถตรวจสอบหน่วยความจำได้จากเมนูสมุดโทรศัพท์ หรือข้อความ



รูป 4.5 การตรวจสอบสถานะความจำของ SIM Card

4.4.2 ระบบรักษาความปลอดภัยของซิมการ์ด

สามารถเปิดใช้งานได้อย่างง่ายๆ ในเมนูตั้งค่า แล้วเลือกไปที่ตั้งค่า PIN โดยกำหนดรหัสเป็นหมายเลข 4 หลักได้ตามต้องการ โดยรหัสเริ่มต้นของซิมการ์ดแต่ละค่ายก็จะไม่เหมือนกัน อย่างเช่นของ GSM Advance จะใช้รหัส 1234, DTAC ใช้รหัส 1800 และ True move ใช้รหัส 0000



รูป 4.6 การตั้งค่าการใช้งานรหัส PIN

เมื่อเปิดใช้งานรหัส PIN แล้ว ทุกครั้งที่เปิดเครื่อง ตัวเครื่องจะถามรหัสก่อนเสมอ ซึ่งสามารถมีโอกาใส่รหัสให้ถูกต้องเพียง 3 ครั้งเท่านั้น แต่ถ้ากรหัสผิดเกิน 3 ครั้งแล้วก็จะจำเป็นต้องใช้รหัส PUK Code ปลดล็อก หากเป็นบางรุ่นจะมีรหัสติดมากับกรอบแผ่นซิมการ์ด แต่หากไม่มีก็ต้องโทรถามคอลล์เซ็นเตอร์ทางเดียว โดยบอกหมายเลขที่ปรากฏอยู่บนซิมการ์ด 19 หลัก อาจจะมาก หรือน้อยกว่านั้นขึ้นอยู่กับบางรุ่น ก็จะได้รหัส 8 หลักมาปลดล็อกอีกที



รูป 4.7 รหัส PUK CODE ที่มาพร้อม SIM Card

แต่หากยังค้นหุ้ร้ังใส่รหัส PUK Code แบบผิดๆ อีก 10 ครั้ง ซิมการ์ดจะล็อคตัวเองทันที ไม่สามารถใช้งานได้อีกต่อไป ซึ่งบางคนบอกว่าซิมการ์ดเสีย แต่จริงๆ แล้วไม่ได้เสีย เพียงแค่ซิมการ์ดล็อคตัวเองเพื่อไม่ให้ผู้อื่นเข้าถึงข้อมูลในซิมการ์ดได้

4.4.3 หมายเลขโทรศัพท์ของคุณไม่ได้อยู่ในซิมการ์ด

การใช้งานจริงเมื่อเปิดโทรศัพท์ขึ้นมา ตัวเครื่องจะส่งหมายเลขซิมการ์ดไปที่ระบบ และระบบก็จะตรวจสอบทันทีว่าหมายเลขซิมการ์ดที่ส่งเข้ามานั้นเป็นหมายเลขอะไร ว่ากันง่ายๆ คือหมายเลขโทรศัพท์ของคุณอยู่ที่ระบบนั่นเอง เพราะไม่ว่าจะเปลี่ยนซิมการ์ดใบไหน เจ้าหน้าที่ก็จะกรอกข้อมูลหมายเลขซิมการ์ดใบใหม่เข้าระบบทุกครั้ง และคุณก็ใช้หมายเลขโทรศัพท์เบอร์เดิม ไม่จำเป็นต้องเปลี่ยนหมายเลขโทรศัพท์ตามไปด้วย

4.4.4 ซิมการ์ดไม่จำเป็นต้องเสียบกับโทรศัพท์มือถือเสมอไป

สมัยก่อนมีเพียง โทรศัพท์มือถือเท่านั้นที่สามารถใช้งานร่วมกันได้ แต่ปัจจุบันซิมการ์ดสามารถเสียบเข้ากับแอร์การ์ด (Air card) เพื่อเชื่อมต่อกับคอมพิวเตอร์เพื่อใช้งานอินเทอร์เน็ตจีพีอาร์เอส/เอ็ดจี หรือ 3G ผ่านพอร์ตยูเอสบีซีก็ได้ บางรุ่นก็เอาไว้ใช้ต่ออินเทอร์เน็ตเพียงอย่างเดียว บางรุ่นก็มีชุดหูฟังไว้เสียบกับแอร์การ์ดเพื่อใช้โทรศัพท์ได้ด้วย



รูป 4.8 การใช้งาน SIM Card นอกเหนือจากโทรศัพท์เคลื่อนที่

หรือหากเป็นเน็ตบุ๊ก หรือ โน้ตบุ๊กบางรุ่นที่มีช่องเสียบซิมการ์ดก็นำมาเสียบเข้ากับตัวเครื่องได้ทันที ไม่จำเป็นต้องใช้แอร์การ์ด ก็สะดวกไม่น้อยเลยทีเดียว นอกจากนี้ยังมีการ์ดรีดเดอร์ (Card reader) บางรุ่นที่นอกจากจะอ่าน หรือเขียนข้อมูลลงการ์ดหน่วยความจำแล้ว ยังมีช่องใส่ซิมการ์ดสำหรับจัดการข้อมูลทั้งรายชื่อ เบอร์โทรศัพท์ และเปิดดูข้อความเอสเอ็มเอสได้จากบนหน้าจอคอมพิวเตอร์ ถ้าใช้เครื่องพิมพ์เบอร์โทร โดยใช้ปุ่มโทรศัพท์ที่พิมพ์ผ่านคอมพิวเตอร์ สะดวกดีเหมือนกันนะ

4.4.5 ซิมการ์ดหายหรือเสียทำยังไงดี

ไม่ว่าซิมการ์ดของคุณจะสูญหาย หรือเสียจนไม่สามารถใช้งานได้ ก็ไม่จำเป็นต้องกลัวว่าจะไม่ได้หมายเลขเดิม เพราะเจ้าหน้าที่จะใส่หมายเลขซิมการ์ดเข้าไปในระบบโดยใช้หมายเลขเดิมได้เหมือนไม่มีอะไรเกิดขึ้น แต่อาจจะต้องเสียค่าข้อมูลในซิมการ์ด ไม่ว่าจะเป็นเบอร์โทรศัพท์เพื่อนๆ หรือเบอร์ลูกค้า ข้อความที่หวานใจส่งให้ ถ้าซิมการ์ดของหาย อันดับแรกเลยต้องรีบโทรไปคอลล์เซ็นเตอร์เพื่อระงับการใช้งานชั่วคราว เพราะซิมการ์ดที่หายไปอาจจะมีผู้หวังดีนำไปใส่เครื่องอื่น โทรให้ หรือที่แย่กว่านั้นคือ โทรออกต่างประเทศที่มีค่าบริการมหาโหด อาจจะต้องแฉ่งความเพื่อนำไปบันทึกแฉ่งความ ไปยื่นกับเจ้าหน้าที่ศูนย์บริการเพื่อขอทำซิมการ์ดใบใหม่



รูป 4.9 รายละเอียดเกี่ยวกับการใช้งาน

4.4.6 วิธีดูแลรักษาซิมการ์ด

การดูแล และรักษาซิมการ์ดให้ใช้ไปได้ยาวนานๆ คือให้เสียบเอาไว้ในเครื่องตลอดเวลาที่ดีที่สุด ไม่
 ต้องทำอะไรให้มากมายนัก เพราะซิมการ์ดที่อยู่ใน โทรศัพท์จะโดนความชื้นเล็กน้อยตลอดเวลา แต่หาก
 ต้องถอดเข้าถอดออกเพื่อนำไปใช้กับเครื่องอื่นอยู่บ่อยๆ ความชื้นอาจจะทำให้หน้าสัมผัสของซิมการ์ดที่
 เป็นทองเหลืองเกิดออกไซด์ หรือสนิม สังกัดได้ไม่ยากเพราะจะมีคราบเกาะติด วิธีทำความสะอาดก็หา
 ขยลบดินสอมาลบออก

บทที่ 5

การออกแบบระบบและพัฒนาโปรแกรม

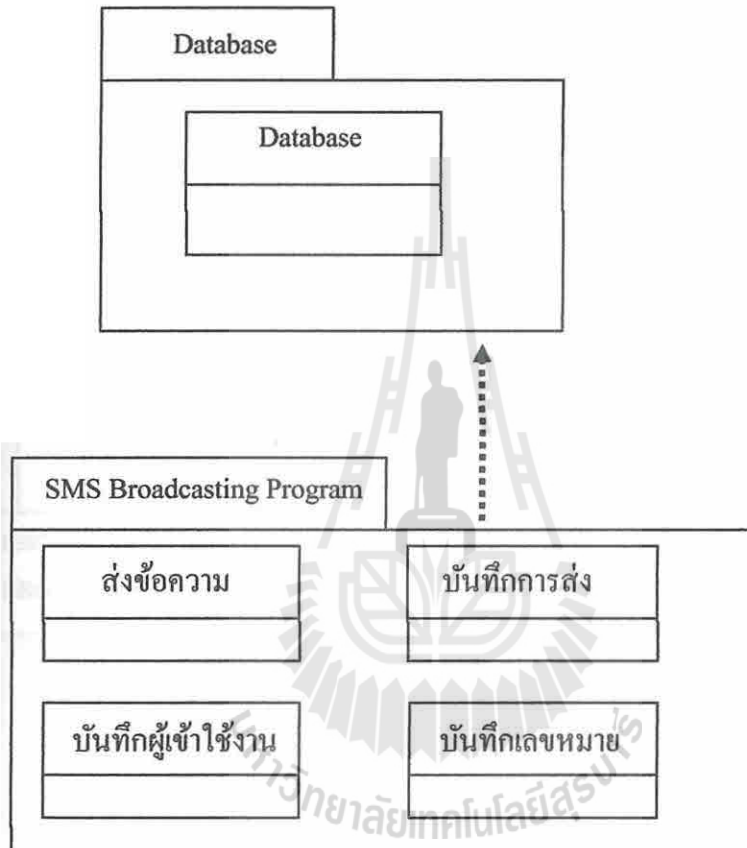
5.1 บทนำ

การออกแบบระบบและพัฒนาโปรแกรม ถือว่าเป็นหัวใจหลักและเป็นส่วนสำคัญอย่างยิ่ง อีกหัวข้อหนึ่งของตัวโปรแกรมซอฟต์แวร์ในโครงการนี้ เพราะเนื่องจากการออกแบบที่ดีของระบบนั้นจะส่งผลให้มีการใช้งานที่ดี คู่มาต่อการลงทุนประหยัดเวลาในด้านการพัฒนาโปรแกรม เพื่อป้องกันไม่ทำให้เกิดปัญหาต่างๆในส่วนต่อไป การออกแบบระบบจึงถือว่าเป็นส่วนพื้นฐานที่มีความสำคัญอย่างมาก เมื่อเราทำการออกแบบระบบให้มีประสิทธิภาพที่ดีและพร้อมที่จะนำไปใช้งานแล้ว ต่อไปจะเป็นส่วนของการพัฒนาโปรแกรม เมื่อเราทำการออกแบบที่ดีก็จะส่งผลให้มีการพัฒนาที่เร็วและดีตามไปด้วย อย่างไรก็ตามการออกแบบระบบและพัฒนาโปรแกรมที่มีลักษณะเป็นระบบระเบียบจะช่วยส่งผลที่ดีต่อผู้ที่จะนำซอฟต์แวร์นี้ไปทำการพัฒนาต่อ หรือช่วยผู้ที่ต้องนำซอฟต์แวร์นี้ไปศึกษาเพื่อเป็นตัวอย่างต่อไป

5.2 ภาพรวมการออกแบบ แสดงด้วยสถาปัตยกรรมของระบบ

5.2.1 สถาปัตยกรรมของระบบเชิงตรรกะ

สถาปัตยกรรมเชิงตรรกะของระบบจะเน้นในด้านการจัดกลุ่มข้อมูลในฐานข้อมูลให้เป็นหมวดหมู่ให้เกิดความเหมาะสม และไม่เกิดความซ้ำซ้อน โดยมีความสัมพันธ์ดังรูป



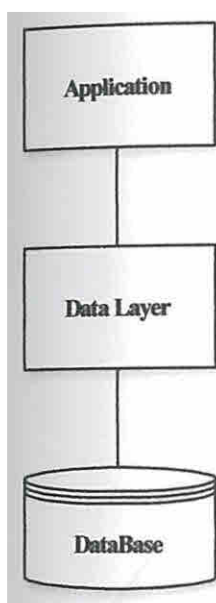
รูป 5.1 แสดงสถาปัตยกรรมของระบบเชิงตรรกะ

5.2.2 สถาปัตยกรรมของระบบเชิงกายภาพ

การทำงานจะแบ่งภาพรวมเป็น 3 Layer คือ

Presentation Layer

ในระดับชั้นนี้จะเป็นส่วนในการแสดงผลการทำงานและรอรับคำสั่งจากผู้ใช้ ซึ่งเป็นส่วนที่ติดต่อกับผู้ใช้งาน โดยตรงผ่านทางคอมพิวเตอร์ โดยมีการติดต่อผ่าน GUI ที่ได้ออกแบบไว้



Application Layer

ในระดับชั้นนี้จะทำการควบคุมการทำงานให้เป็นไปตามเงื่อนไข หลังจากที่ได้รับคำสั่งของจากชั้น Presentation Layer หรือทำการนำข้อมูลที่ได้รับมาจาก Data Link Layer มาประมวลผลเพื่อที่จะทำงานต่อไป

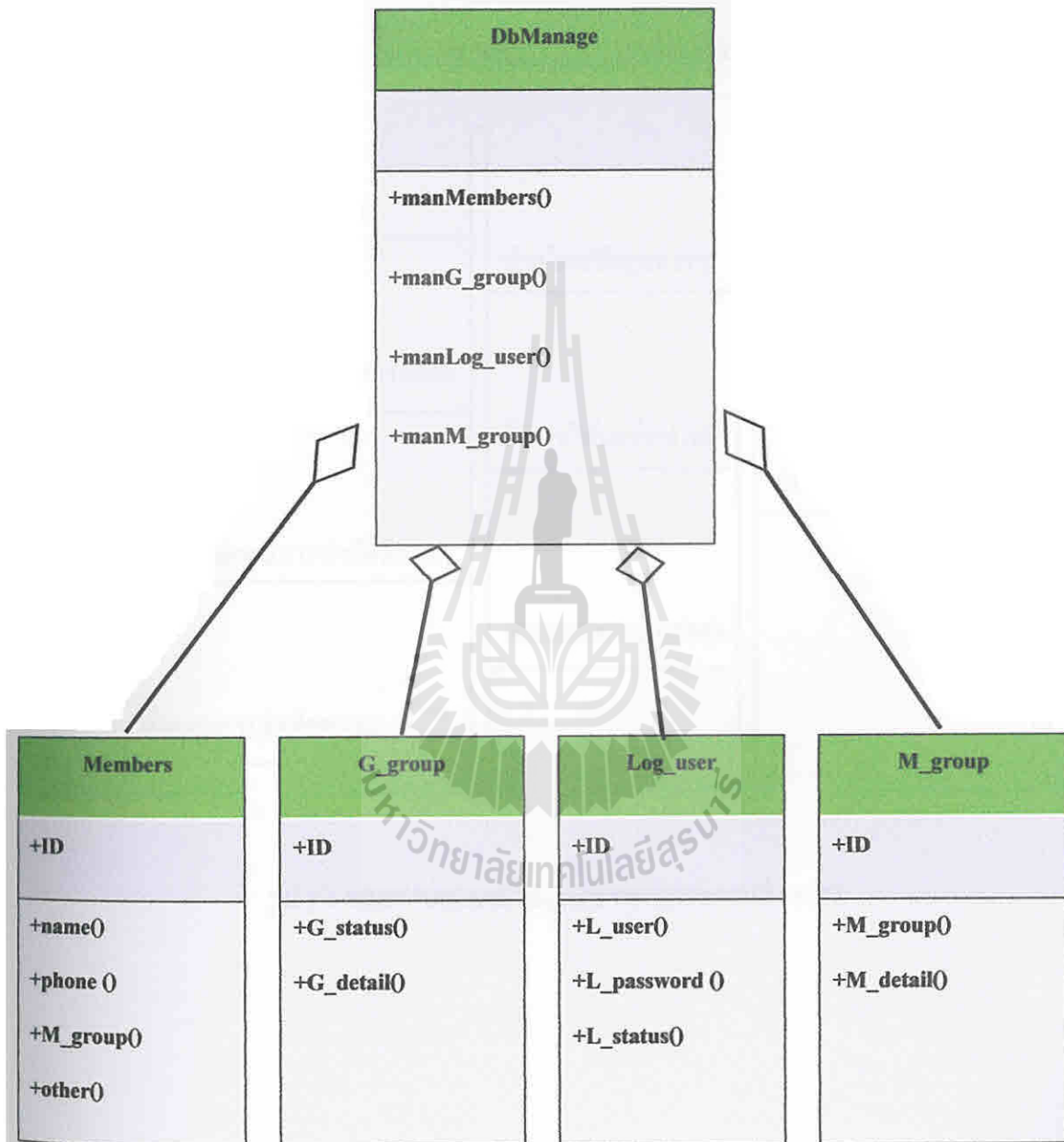
Data Link Layer

จะจัดการในส่วนการติดต่อ และการทำงานกับข้อมูลภายนอก เช่นการติดต่อกับ ฐานข้อมูล(Database) และการติดต่อกับบอร์ด wireless CPU ผ่านทางพอร์ทอนุกรม RS-232 ซึ่งในชั้นนี้จะทำการรับส่งข้อมูลตามการทำงานในชั้น Application Layer ที่ได้รับขอมมา

รูป 5.2 แสดงสถาปัตยกรรมของระบบเชิงกายภาพ

5.3 คลาสออกแบบ (Design Class)

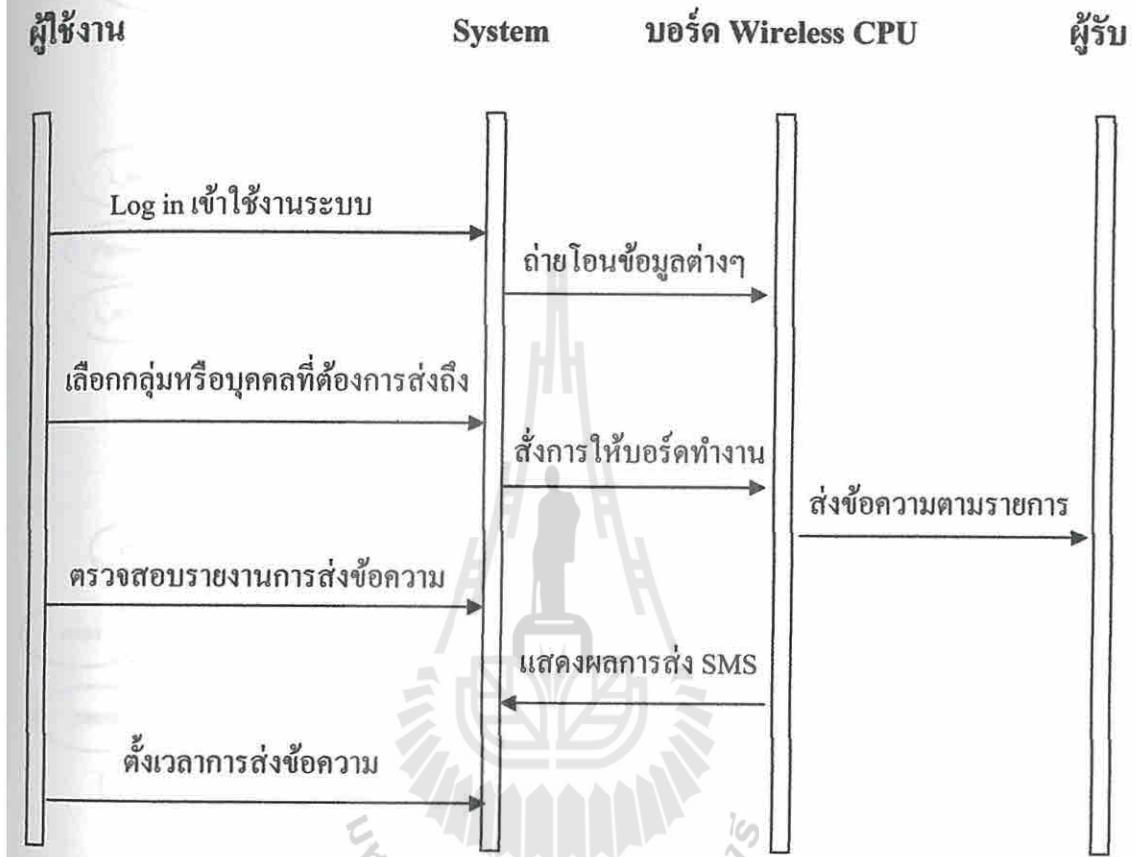
โปรแกรมจัดการระบบฐานข้อมูล แต่ละคลาสมีรายละเอียดดังแสดงในรูปที่



รูป 5.3 แสดง Class Diagram ของการจัดการระบบฐานข้อมูล

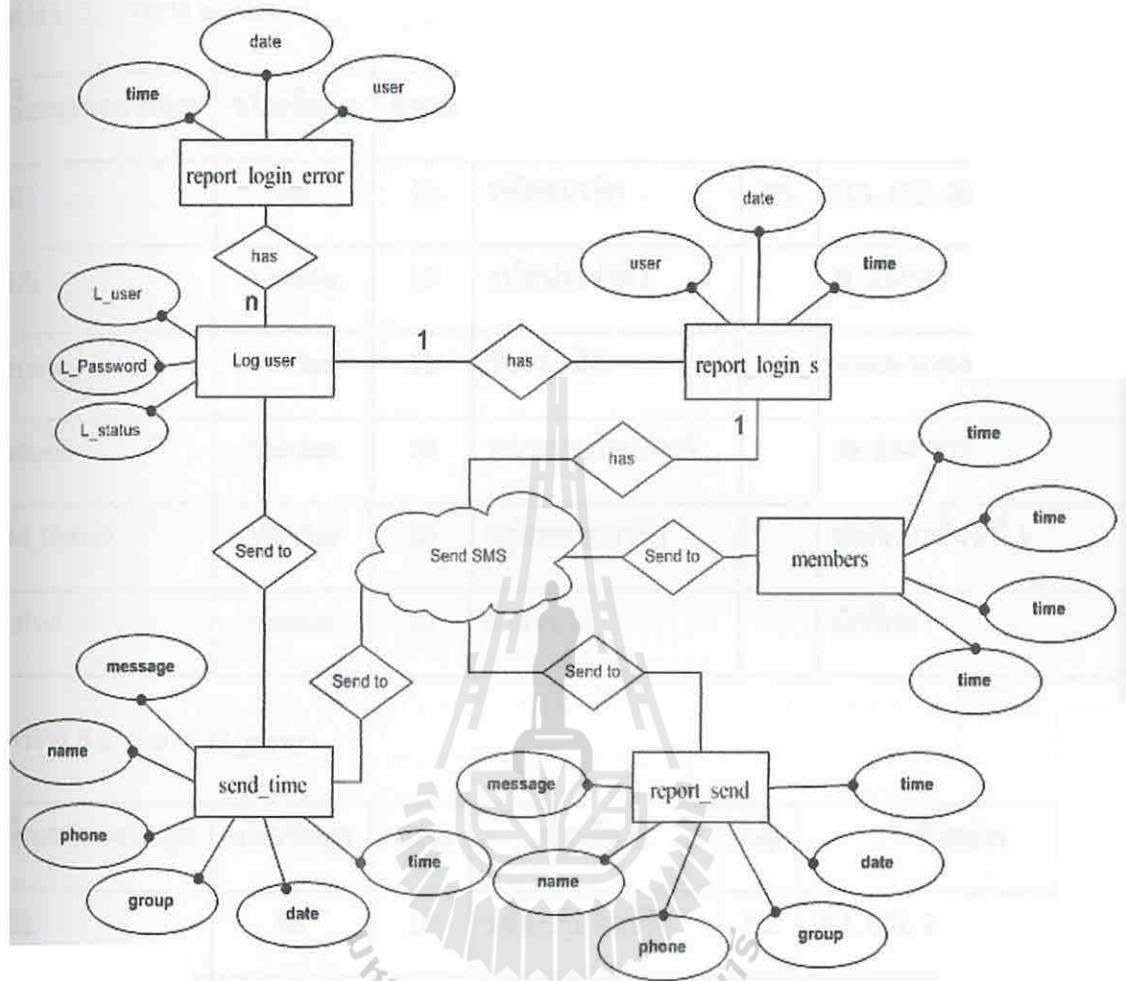
5.4 แผนภาพแสดงพฤติกรรมเชิงพลวัต

Sequence diagram



รูป 5.4 แสดง Sequence diagram ของโปรแกรมส่ง SMS

5.5 แผนผังแสดงความสัมพันธ์



รูป 5.5 แสดงแผนผังแสดงความสัมพันธ์ของโปรแกรมส่ง SMS

5.6 พจนานุกรมข้อมูล

ตาราง 5.1 ตาราง members

ชื่อเขตของข้อมูล	ชนิดข้อมูล	ขนาด	คำอธิบาย	Key	ตัวอย่าง
<u>ID</u>	Int	10	รหัสสมาชิก	PK	001, 002, 003
Ids	Varchar	15	รหัสประจำตัว		B1234567
Name	Varchar	50	ชื่อสมาชิก		นายเอ นายสกุลบี
Phone	Varchar	50	หมายเลขโทรศัพท์		0812345678
M_Group	Varchar	50	กลุ่มของสมาชิก		นักศึกษาชั้นปีที่ 1
other	Varchar	50	คำอธิบาย		นักศึกษา

ตาราง 5.2 ตาราง G_group

ชื่อเขตของข้อมูล	ชนิดข้อมูล	ขนาด	คำอธิบาย	Key	ตัวอย่าง
<u>ID</u>	Int	10	รหัสกลุ่มสมาชิก	PK	001, 002, 003
G_Status	Varchar	50	ชื่อสถานะกลุ่ม		ผู้ดูแลระบบ, บุคคลทั่วไป
G_detail	Varchar	50	คำอธิบาย		ดูแลระบบ

ตาราง 5.3 ตาราง M_group

ชื่อเขตของข้อมูล	ชนิดข้อมูล	ขนาด	คำอธิบาย	Key	ตัวอย่าง
<u>ID</u>	Int	10	รหัสกลุ่มที่ส่ง	PK	001, 002, 003
M_group	Varchar	50	ชื่อกลุ่ม		นักศึกษาชั้นปีที่ 1
M_detail	Varchar	50	คำอธิบาย		นักศึกษารหัส B53xxx

ตาราง 5.4 ตาราง Log_user

ชื่อเขตของข้อมูล	ชนิดข้อมูล	ขนาด	คำอธิบาย	Key	ตัวอย่าง
ID	Int	10	รหัสผู้เข้าใช้งาน	PK	001, 002, 003
L_User	Varchar	50	ชื่อผู้เข้าใช้งาน		admin
L_Password	Varchar	50	รหัสเข้าใช้งาน		Ab123
L_status	Varchar	50	สถานะผู้เข้าใช้งาน		ผู้ดูแลระบบ

ตาราง 5.5 ตาราง report_login_error

ชื่อเขตของข้อมูล	ชนิดข้อมูล	ขนาด	คำอธิบาย	Key	ตัวอย่าง
id	Int	10	ลำดับที่เข้าใช้	PK	001, 002, 003
user	Varchar	50	ชื่อผู้ใช้งาน		adtest
date	Varchar	50	วันที่ใช้งาน		6 มิถุนายน 2553
time	Varchar	50	เวลาที่ใช้งาน		10:26:17

ตาราง 5.6 ตาราง report_login_s

ชื่อเขตของข้อมูล	ชนิดข้อมูล	ขนาด	คำอธิบาย	Key	ตัวอย่าง
id	Int	10	ลำดับที่เข้าใช้	PK	001, 002, 003
user	Varchar	50	ชื่อผู้ใช้งาน		admin
date	Varchar	50	วันที่ใช้งาน		6 มิถุนายน 2553
time	Varchar	50	เวลาที่ใช้งาน		17:43:38

ตาราง 5.7 ตาราง report_send

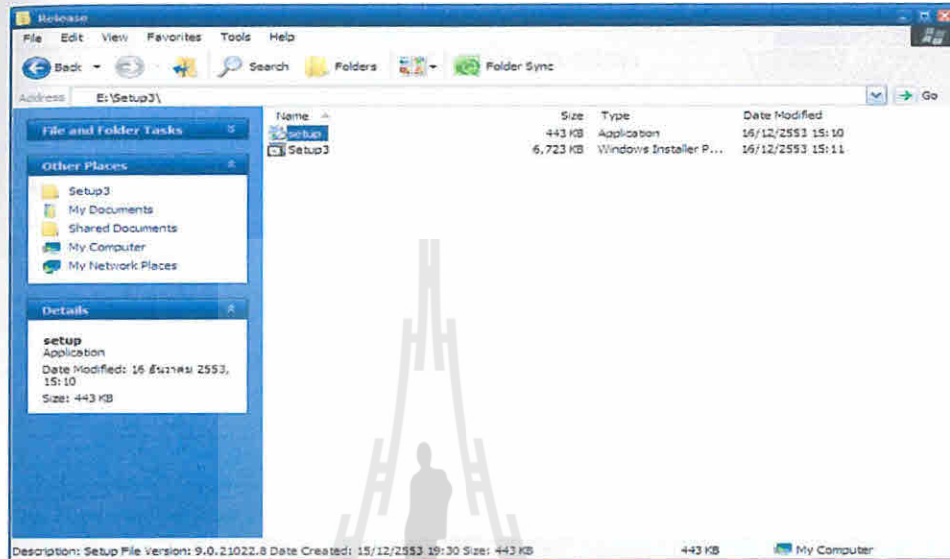
ชื่อเขตของข้อมูล	ชนิดข้อมูล	ขนาด	คำอธิบาย	Key	ตัวอย่าง
id	Int	10	ลำดับที่ส่ง	PK	001, 002, 003
message	Varchar	160	ข้อความที่ส่ง		ทดสอบส่งข้อความ
name	Varchar	50	ชื่อผู้รับ		นายเอ นายสกุลบี
phone	Varchar	50	เบอร์โทรศัพท์		0812345678
Group	Varchar	50	กลุ่มของผู้รับ		ทั่วไป
date	datetime		วันที่ส่ง		25/10/2553 18:33:10
time	Varchar	50	เวลาที่ส่ง		18:33:10

ตาราง 5.8 ตาราง send_time

ชื่อเขตของข้อมูล	ชนิดข้อมูล	ขนาด	คำอธิบาย	Key	ตัวอย่าง
id	Int	10	ลำดับที่ส่ง	PK	001, 002, 003
message	Varchar	160	ข้อความที่ส่ง		ทดสอบส่งข้อความ
name	Varchar	50	ชื่อผู้รับ		นายเอ นายสกุลบี
phone_num	Varchar	50	เบอร์โทรศัพท์		0812345678
Group	Varchar	50	กลุ่มของผู้รับ		ทั่วไป
date	datetime		วันที่ส่ง		3 พฤศจิกายน 2553
time	Varchar	50	เวลาที่ส่ง		18:33

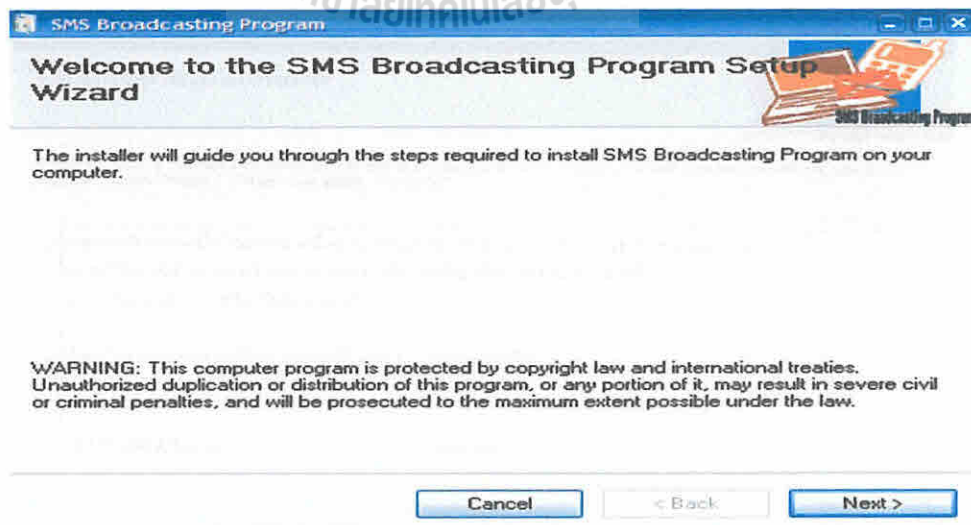
5.7 การติดตั้งโปรแกรมในการใช้งาน

5.7.1 เมื่อจะทำการติดตั้งใส่แผ่น CD ROM แล้วทำเลือกคลิกที่ 



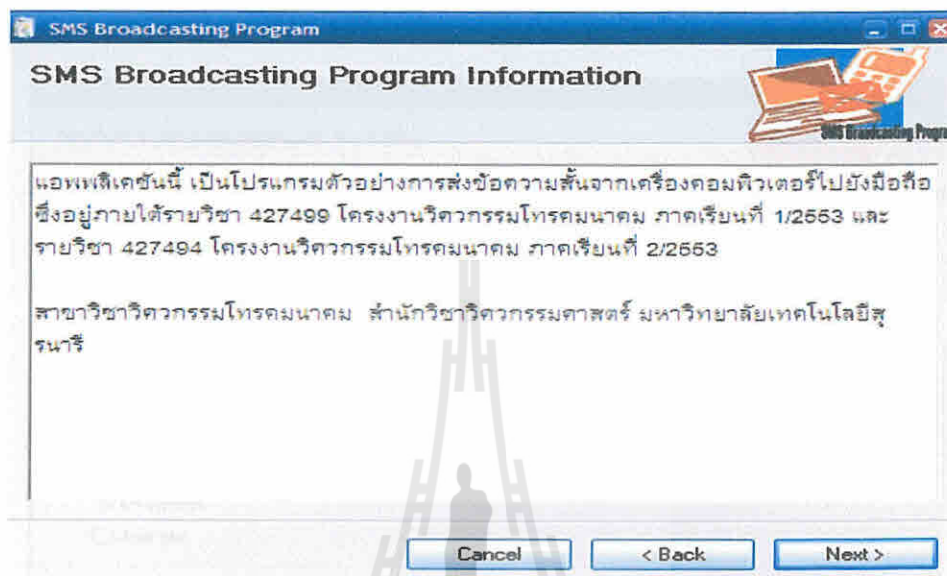
รูป 5.6 แสดงตัวติดตั้งโปรแกรมเมื่อทำการใส่แผ่น CD ROM

5.7.2 เมื่อทำการดับเบิลคลิกที่  แล้วจะพบว่าจะเห็นหน้าต่างรับดั่งภาพข้างล่างดังนั้นขั้นตอนต่อไปคือให้ทำการคลิกที่ปุ่ม Next >



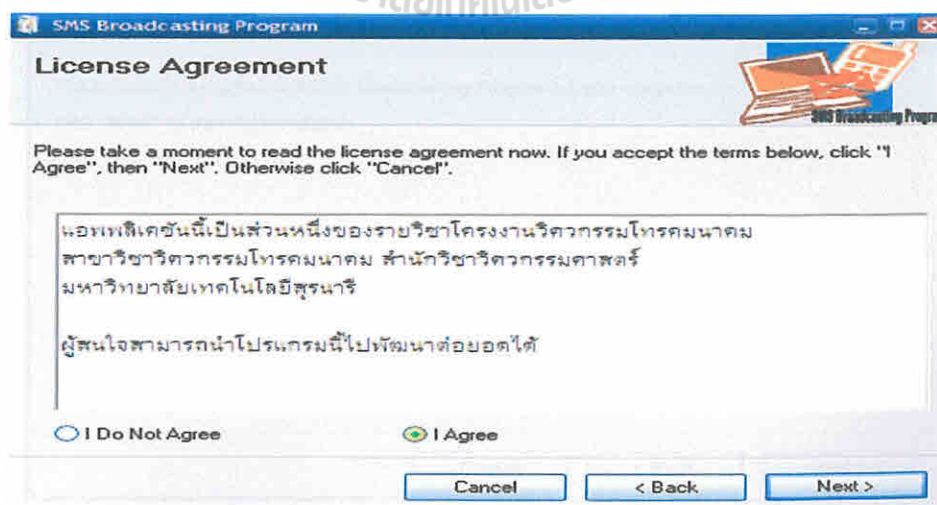
รูป 5.7 หน้าจอแสดงข้อความต้อนรับการติดตั้งโปรแกรม SMS Broadcasting Program

5.7.3 จากขั้นตอนที่แล้วเมื่อเราทำการคลิกที่ปุ่ม Next > แล้วจะปรากฏหน้าจอแสดงถึงรายละเอียดของโปรแกรมดังแสดงได้ดังภาพด้านล่าง เมื่อทำความเข้าใจกับข้อความแล้วก็ทำการคลิกที่ปุ่ม Next> ต่อไป



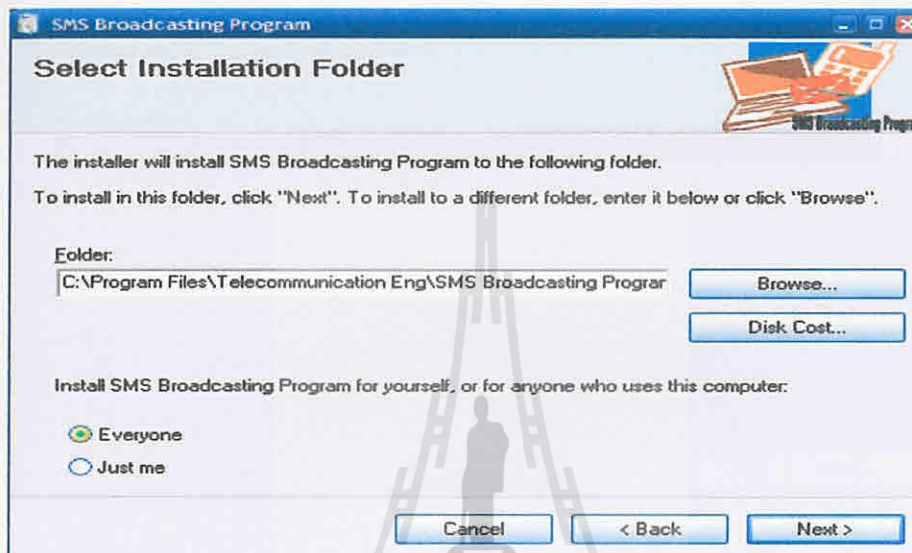
รูป 5.8 แสดงข้อมูลของโปรแกรม

5.7.4 จากขั้นตอนที่แล้วเมื่อเราทำการคลิกที่ปุ่ม Next > แล้วจะพบกับหน้าจอแสดงถึง License ของโปรแกรมเมื่อทำความเข้าใจกับข้อความแล้วก็ให้ทำการคลิกเลือกที่ข้อความ I Agree เพื่อแสดงการยอมรับเงื่อนไขของโปรแกรมจากนั้นก็ให้ทำการคลิกที่ปุ่ม Next > ต่อไป



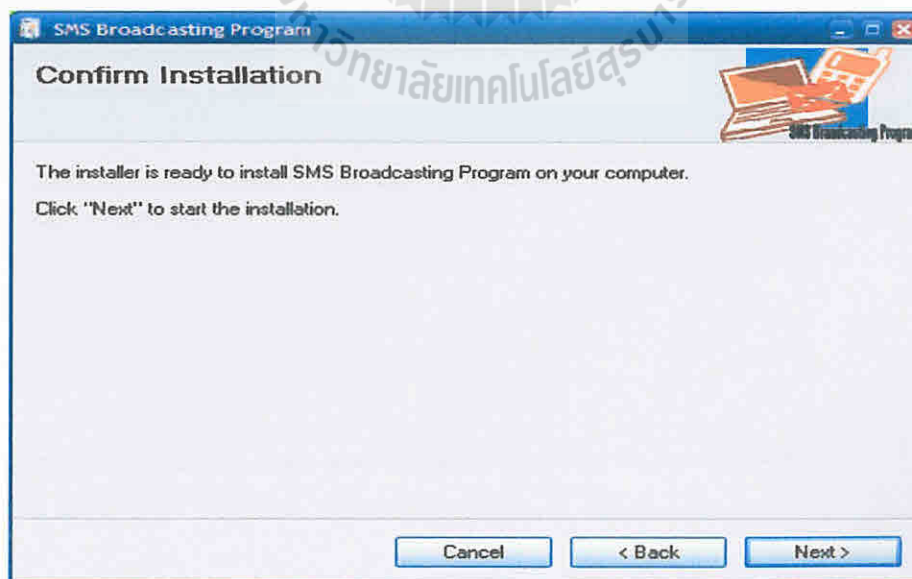
รูป 5.9 แสดงถึง License ของโปรแกรม

5.7.5 จากขั้นตอนที่แล้วเมื่อเราทำการคลิกที่ปุ่ม Next > แล้วจะปรากฏหน้าจอภาพข้างล่างนี้ โดยให้เราทำการคลิกที่ปุ่ม Browse... เพื่อทำการเลือกว่าจะให้ระบบทำการลงโปรแกรมตรงไหนเมื่อทำการเลือกเสร็จแล้วให้ทำการคลิกที่ข้อความ Everyone เมื่อทำการดังกล่าวข้างต้นเสร็จแล้วให้ทำการคลิกที่ปุ่ม Next > เพื่อเริ่มขั้นตอนต่อไป



รูป 5.10 แสดงหน้าจอการเลือกที่ตั้งของ โปรแกรมเมื่อทำการ Installation โปรแกรม

5.7. ขั้นตอนต่อไปให้ทำการคลิกที่ปุ่ม Next อีกครั้งเพื่อทำการยืนยันการลง โปรแกรม



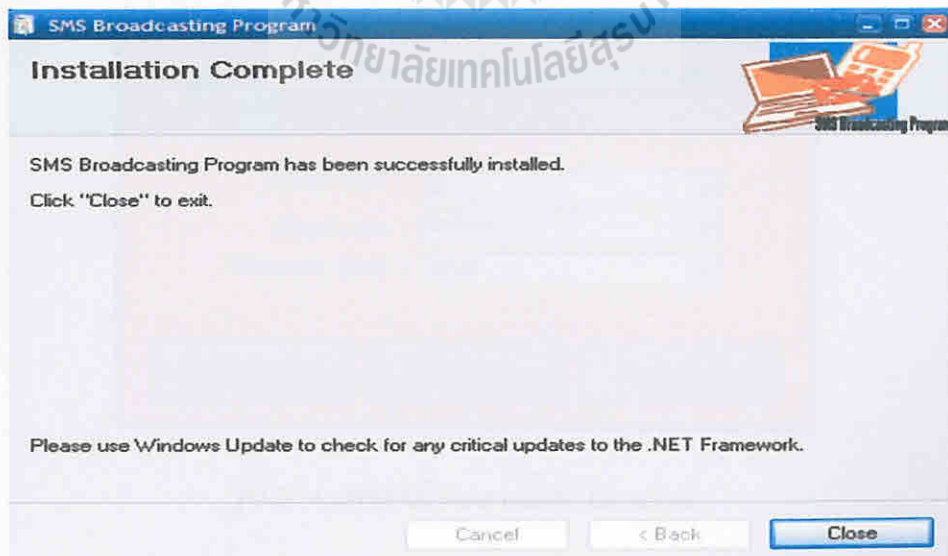
รูป 5.11 แสดงหน้าจอการตอบรับการลง โปรแกรมลงเครื่องผู้ใช้งาน

5.7.7 เมื่อทำการคลิกที่ปุ่ม Next> ในขั้นตอนต่อไปดังแสดงตามภาพข้างล่างนี้จะเห็นได้ว่าระบบจะทำการโหลดตัวโปรแกรมเพื่อติดตั้ง



รูป 5.12 แสดงการ โหลดตัวโปรแกรมเพื่อทำการติดตั้ง

5.7.8 ขั้นตอนสุดท้ายเมื่อทำการติดตั้งโปรแกรมเสร็จสิ้นแล้วจะปรากฏหน้าจอดังภาพแสดงดังข้างล่าง



รูป 5.13 แสดงหน้าจอเมื่อทำการติดตั้งโปรแกรมเสร็จแล้ว

5.8 ส่วนต้นแบบของโครงร่างของส่วนเชื่อมโยงกับผู้ใช้

5.8.1 หน้าจอหลักของโปรแกรม

หน้าจอหลักของโปรแกรม แบ่งออกเป็น 5 ส่วนคือ

1. การตั้งค่าโปรแกรม
2. หน้าจอแสดงหน้าหลักของโปรแกรม
3. หน้าจัดการการส่งข้อความ SMS
4. หน้าจัดการรายชื่อติดต่อ
5. หน้ากำหนดผู้เข้าใช้งาน
6. หน้ากำหนดกลุ่มที่ใช้ในการติดต่อ
7. หน้ารายงานการส่งข้อความ
8. หน้ารายงานการเข้าใช้ระบบ

5.8.2 หน้าจอการตั้งค่าโปรแกรม

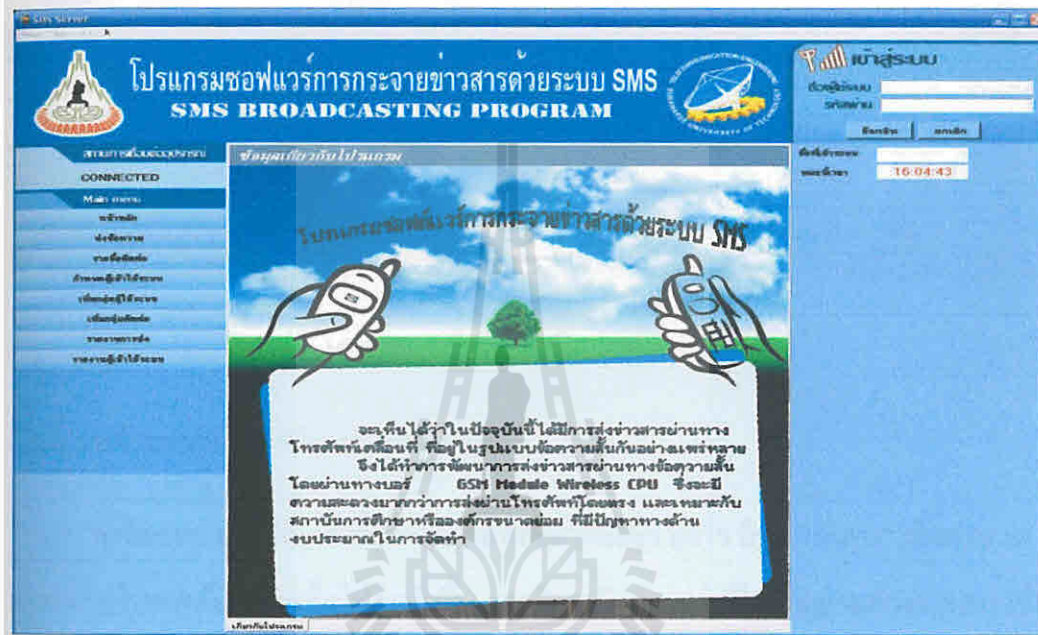
หน้าจอของการตั้งค่าโปรแกรมสำหรับการปรับแต่งโปรแกรมให้ทำงานได้ถูกต้อง ตรงกับความต้องการของผู้ใช้ และการตั้งค่าต่างๆให้ตรงกับอุปกรณ์ เช่น การกำหนดพอร์ทที่ใช้งานจริง



รูป 5.14 แสดงหน้าจอการตั้งค่าโปรแกรม

5.8.3 หน้าจอแสดงหน้าหลักของโปรแกรม

ลักษณะของหน้าจอแสดงหน้าหลักของโปรแกรมก่อนเริ่มทำการใช้งาน ทางด้านซ้ายมือ จะบอกถึงรายละเอียดและหัวข้อต่างๆ ส่วนทางด้านขวามือ จะบอกถึงการเข้าสู่ระบบ Log In เพื่อที่จะได้เริ่มทำงานในส่วนต่อไป

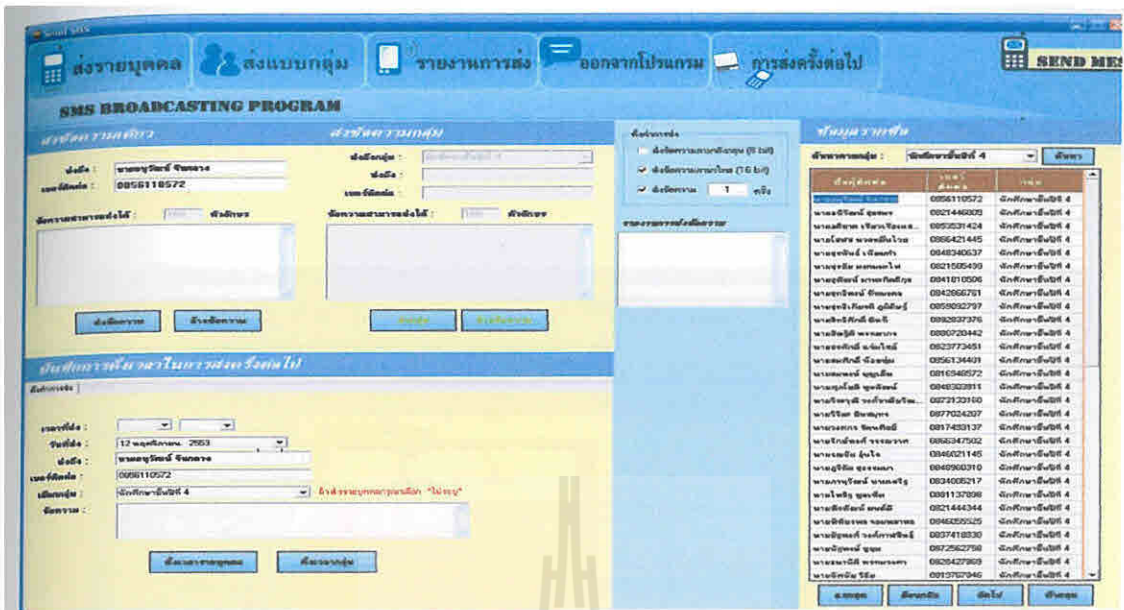


รูป 5.15 แสดงหน้าจอแสดงหน้าหลักของโปรแกรม

5.8.4 หน้าจอแสดงการจัดการการส่งข้อความ SMS

หน้าจอแสดงการจัดการการส่งข้อความซึ่งจะแสดงส่วนประกอบอยู่ 3 ส่วนคือ

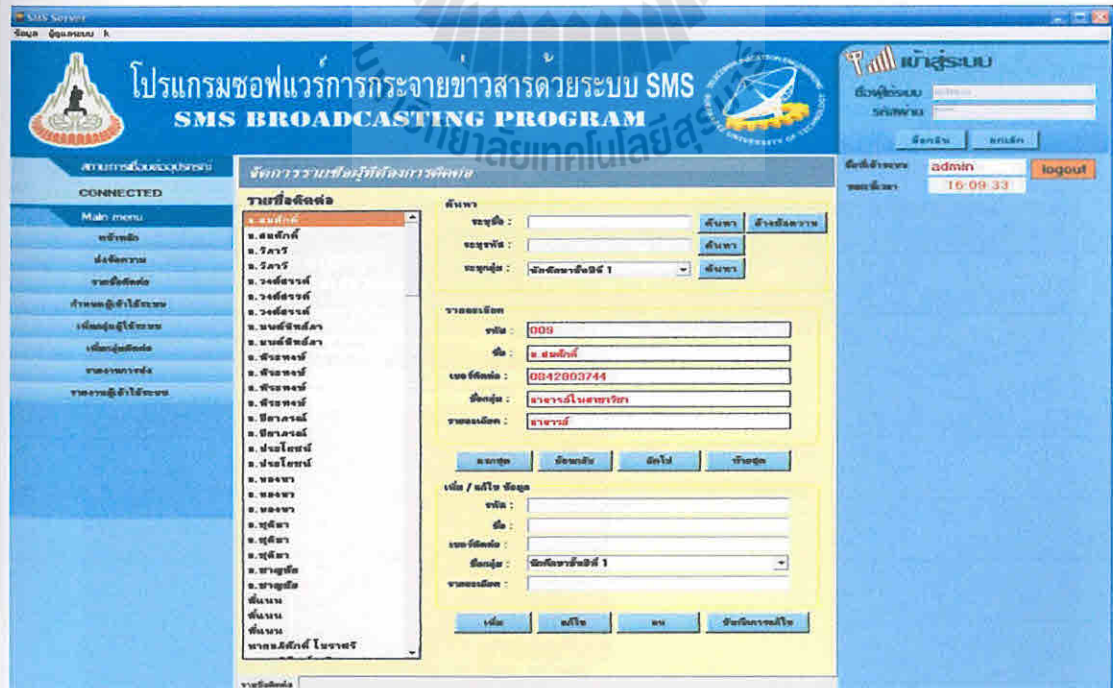
1. ส่วนของการจัดส่งข้อความรายบุคคล
2. ส่วนของการจัดส่งข้อความแบบกลุ่ม
3. ส่วนของการตั้งเวลาในการจัดส่งข้อความ



รูป 5.16 แสดงหน้าจอการจัดการการส่งข้อความ

5.8.5 หน้าจอแสดงการจัดการรายชื่อติดต่อ

ทำจัดการรายชื่อติดต่อเพื่อการติดต่อที่สะดวกและรวดเร็ว มีปุ่ม “ค้นหา” เพื่ออำนวยความสะดวกในการค้นหารายชื่อให้ง่ายขึ้น และมีการจัดสรรแบ่งเป็นกลุ่มตามลักษณะการใช้งาน



รูป 5.17 แสดงหน้าจอการจัดการรายชื่อติดต่อ

5.8.6 หน้าจอแสดงการกำหนดผู้ใช้งาน

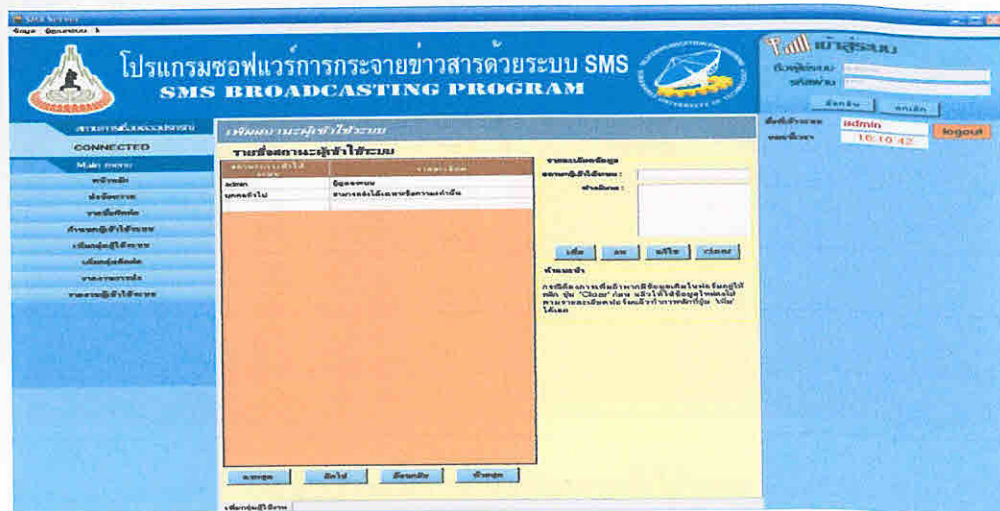
หน้าจอแสดงการกำหนดผู้ใช้งานเป็นส่วนที่ตั้งค่าและกำหนดชื่อและรหัสผ่านของผู้ที่จะใช้งาน



รูป 5.18 แสดงหน้าจอการกำหนดผู้ใช้งาน

5.8.7 หน้าจอแสดงการกำหนดกลุ่มที่ใช้ในการติดต่อ

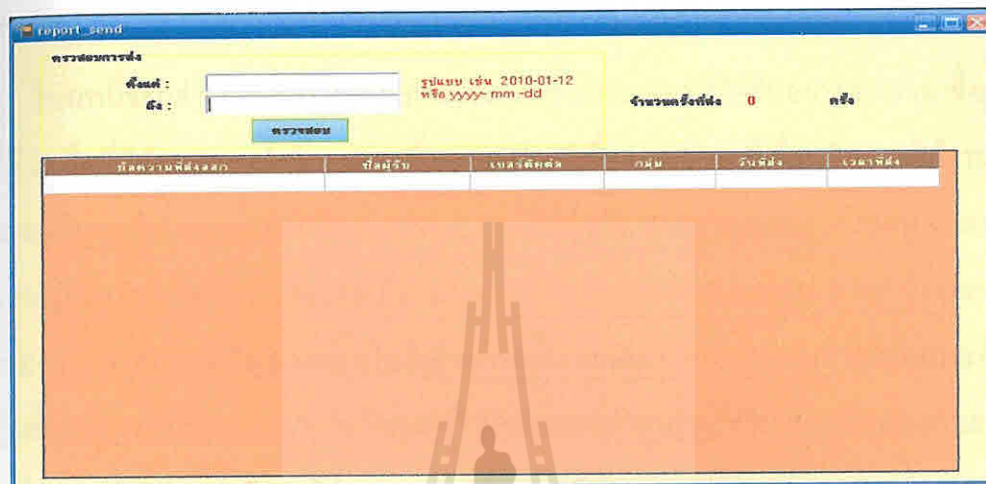
หน้าจอแสดงการกำหนดกลุ่มที่ใช้ในการติดต่อ เป็นการตั้งค่าและตั้งชื่อกลุ่มเพื่อใช้ในการติดต่อที่สะดวก รวดเร็วมากยิ่งขึ้น



รูป 5.19 แสดงหน้าจอการกำหนดกลุ่มที่ใช้ในการติดต่อ

5.8.8 หน้าจอแสดงหน้ารายงานการส่งข้อความ

หน้าจอแสดงหน้ารายงานการส่งข้อความ เป็นการบอกรายละเอียดของข้อความที่เคหะทำการจัดส่งมาแล้ว



รูป 5.20 แสดงหน้าจอรายงานการส่งข้อความ

5.8.9 หน้าจอแสดงหน้ารายงานการเข้าใช้ระบบ

หน้าจอแสดงหน้ารายงานการเข้าใช้ระบบ มีไว้เพื่อทำการตรวจสอบว่ามีผู้ใดเข้ามาใช้งานในระบบ เมื่อใด เวลาใด

ส่งถึง	เบอร์โทรศัพท์	เวลาที่ส่ง
test	4 ตุลาคม 2553	14:49:22
admin	4 ตุลาคม 2553	14:50:23
test	4 ตุลาคม 2553	15:01:34
admin	4 ตุลาคม 2553	15:01:58
admin	4 ตุลาคม 2553	15:05:19
admin	4 ตุลาคม 2553	15:09:14
admin	4 ตุลาคม 2553	15:12:15
admin	4 ตุลาคม 2553	15:13:28
admin	4 ตุลาคม 2553	15:37:21
admin	7 ตุลาคม 2553	13:36:39
admin	7 ตุลาคม 2553	13:49:19
admin	8 ตุลาคม 2553	13:18:20
admin	8 ตุลาคม 2553	13:33:03
admin	8 ตุลาคม 2553	13:35:08
admin	8 ตุลาคม 2553	13:47:12
admin	8 ตุลาคม 2553	13:49:06
admin	8 ตุลาคม 2553	13:49:36
admin	11 ตุลาคม 2553	13:29:48
admin	11 ตุลาคม 2553	13:31:00
admin	11 ตุลาคม 2553	13:36:06
admin	11 ตุลาคม 2553	13:40:57
admin	11 ตุลาคม 2553	13:44:43
admin	11 ตุลาคม 2553	13:47:05
admin	11 ตุลาคม 2553	13:52:14
admin	11 ตุลาคม 2553	14:02:50
admin	11 ตุลาคม 2553	14:16:09
admin	11 ตุลาคม 2553	14:20:33
admin	11 ตุลาคม 2553	14:30:39

ส่งถึง	เบอร์โทรศัพท์	เวลาที่ส่ง
admin	11 ตุลาคม 2553	14:01
admin	11 ตุลาคม 2553	14:01
test	14 ตุลาคม 2553	14:00
test	20 ตุลาคม 2553	18:46
admin	11 ตุลาคม 2553	8:55

รูป 5.21 แสดงหน้ารายงานการเข้าใช้ระบบ

บทที่ 6

สรุปผลและข้อเสนอแนะ

6.1 บทนำ

ในบทนี้จะกล่าวถึงผลสรุปของโครงการและข้อเสนอแนะที่ได้รับจากโครงการ ซึ่งเราจะเห็นได้ว่า สิ่งที่ได้จากการทำโครงการนี้ เราจะต้องมีพื้นฐานความรู้เกี่ยวกับการใช้งานของโปรแกรมต่างๆ เช่น Microsoft Office, Visual C#, โปรแกรมฐานข้อมูล SQL เป็นต้น ถ้าหากพื้นฐานความรู้ดังกล่าวยังคงมีไม่เพียงพอ ก็จะส่งผลกระทบต่อโครงการโดยตรง อาจทำให้เกิดความล่าช้าของการทำงาน แต่ปัญหาเหล่านั้นก็เป็นผลดีทำให้เกิดการกระตุ้นและตื่นตัวในการค้นหาความรู้เพิ่มเติม ทำให้มีความเข้าใจในเรื่องการใช้โปรแกรมต่างๆ มากยิ่งขึ้น และข้อเสนอแนะต่างๆ จะเป็นส่วนช่วยให้เกิดการพัฒนาโปรแกรมซอฟต์แวร์ต่อไป

6.2 ปัญหาและแนวทางในการแก้ไขปัญหา

ในการทำโครงการโปรแกรมซอฟต์แวร์การกระจายข่าวสารด้วยระบบ SMS (SMS Broadcasting Program) จะเห็นได้ว่าการพบปัญหาได้บ่อยๆ ดังจะแสดงไว้ในตาราง 6.1 ซึ่งประกอบด้วยตัวปัญหาที่พบ สาเหตุของปัญหา และรวมถึงวิธีการแก้ไขปัญหา

ปัญหาที่พบในขณะดำเนินงาน	สาเหตุและวิธีการแก้ไข
1. เกิดข้อผิดพลาดในการเขียนโปรแกรมในช่วงแรก	สาเหตุ เนื่องจากทางผู้จัดทำยังไม่มี ความชำนาญในการเขียนโปรแกรมด้วยภาษา C# วิธีการแก้ไข ทางผู้จัดทำได้ทำการศึกษาหาความรู้เพิ่มเติมของการเขียนโปรแกรมด้วยภาษา C# จากหนังสือ และInternet

<p>2. เกิดข้อผิดพลาดในการสร้างฐานข้อมูลที่จะนำมาใช้เป็นหน่วยความจำในการเรียกใช้ในการส่งข้อมูล</p>	<p>สาเหตุ ทางผู้จัดทำไม่ทราบว่าจะใช้โปรแกรมใดมาเป็นตัวช่วยในการเขียนโปรแกรม</p> <p>วิธีการแก้ไข ทางผู้จัดทำได้ทำการศึกษาจึงได้ทราบว่าจะต้องใช้โปรแกรม SQL เข้ามาช่วย</p>
<p>3. ความผิดพลาดที่เกิดจากตัวอุปกรณ์ของทางบริษัทเกิดมีปัญหา</p>	<p>สาเหตุ ในการที่จะนำเอา ชุด CPU Wireless ของทางบริษัทมาประยุกต์ใช้กับ โครงงานนั้น เราจะต้องมีการติดตั้งหรือเพิ่มรายละเอียดเข้าไปหลายโปรแกรม จึงไม่ทราบว่าเกิดปัญหาอะไร ณ จุดไหน ที่ทำให้ CPU Wireless เกิดการทำงานผิดปกติ</p> <p>วิธีการแก้ไข ได้ Board ตัวใหม่มาจากทางบริษัท แล้วทำการโหลดตัวโปรแกรมเข้าไปใหม่</p>
<p>4. เมื่อนำโปรแกรมไปติดตั้งยังเครื่องคอมพิวเตอร์เกิดปัญหาระหว่างการติดตั้ง</p>	<p>สาเหตุ เพราะตัวผู้ติดตั้งเกิดการผิดพลาด พิมพ์โค้ดโปรแกรมตกหล่น</p> <p>วิธีการแก้ไข กลับไปทำการตรวจสอบโค้ดโปรแกรม และทำการเพิ่ม โค้ด โปรแกรมให้ครบถ้วน</p>
<p>5. Delay ทำงานช้าเกินไป และไม่สามารถทำงานได้ตามที่กำหนด (Hardware)</p>	<p>สาเหตุ เนื่องจาก delay ทำงานได้ช้า กว่าที่กำหนดไว้ จึงทำให้ไม่ตรงตามวัตถุประสงค์ที่วางไว้เพราะจะเกิดการล่าช้าในการส่งข้อความสั้น</p> <p>วิธีการแก้ไข ทำการลดจำนวนตัวอักษรที่ทำการส่ง จาก 70 ตัวอักษรเป็น 50 ตัวอักษรและใช้การเช็คเวลาเครื่อง</p>

ตาราง 6.1 ปัญหาและสาเหตุที่พบในขณะดำเนินงานและวิธีการแก้ไข

6.3 ข้อเสนอแนะ

6.3.1 ก่อนการติดตั้งโปรแกรมควรวัดศึกษาวิธีการติดตั้งอย่างละเอียดเพราะอาจเกิดการผิดพลาดในการติดตั้ง

6.3.2 การใช้โปรแกรมตรวจสอบการตั้งค่าระบบของโปรแกรมให้เหมาะสม เช่น การตั้งค่าของ port และ buad rate

6.3.3 ในการส่งข้อความต้องคำนึงถึงจำนวนตัวอักษรด้วย สำหรับภาษาไทยนั้นจะทำการส่งได้ไม่เกิน 50 ตัวอักษร ส่วนภาษาอังกฤษจะไม่เกิน 160 ตัวอักษร

6.3.4 อุปกรณ์ที่ใช้ในการทำโครงงานบางชิ้นมีขนาดเล็กและเปราะบาง ผู้ใช้งานควรใช้งานด้วยความระมัดระวังเป็นอย่างมาก

6.4 แนวทางในการพัฒนาต่อไป

เนื่องจากโครงงาน โปรแกรมซอฟต์แวร์การกระจายข่าวสารด้วยระบบ SMS (SMS Broadcasting Program) นี้สามารถนำไปปรับและพัฒนาให้ใช้ได้กับองค์กรที่มีขนาดเล็ก เพื่อใช้ในการกระจายข่าวสารได้ซึ่งทำได้สะดวก และประหยัด เพราะเราสามารถเลือกราคาที่เหมาะสมกับองค์กรของเราได้ และไม่จำเป็นต้องไปเช่าระบบ SMS Gateway ที่มีความสิ้นเปลืองเรื่องค่าใช้จ่ายมากเกินไป จึงไม่เหมาะสำหรับองค์กรขนาดเล็ก

6.5 บทสรุป

โครงการโปรแกรมซอฟต์แวร์การกระจายข่าวสารด้วยระบบ SMS (SMS Broadcasting Program) มีส่วนประกอบหลักดังนี้

1. ระบบฐานข้อมูลที่ใช้

- Microsoft SQL Server 2007

2. ภาษาที่ใช้ในการพัฒนา

- ภาษา C#
- คำสั่ง AT-Command
- การส่งข้อความในโหมด PDU

3. Hardware ที่ใช้ในการพัฒนา

- Personal Computer หรือ Notebook
- บอร์ด GSM Module Wireless CPU
- SIM Card โทรศัพท์เคลื่อนที่

4. Software ที่ใช้ในการพัฒนา

- Microsoft Visual Studio .net 2008
- Adobe Photoshop CS3
- Microsoft Office 2007

Input Specification

1. รับข้อมูลผู้เข้าใช้ระบบผ่านทางคีย์บอร์ด
2. รับข้อมูลกลุ่มหรือข้อมูลบุคคลที่ต้องการส่งข้อความสั้นถึงผ่านทางคีย์บอร์ด
3. รับข้อมูลข้อความสั้น SMS ผ่านทางคีย์บอร์ด
4. รับคำสั่งส่งข้อความสั้น SMS โดยผู้ดูแลระบบหรือบุคคลที่ถูกกำหนดไว้

Output Specification

1. บันทึกข้อมูลกลุ่ม ข้อมูลผู้ที่ต้องการส่งข้อความสั้น ไปถึง และผู้ที่สามารถเข้าใช้งานระบบลงในฐานข้อมูล
2. ส่ง SMS จากฐานข้อมูลไปยังกลุ่มหรือตัวบุคคลที่ระบุไว้โดยการติดต่อผ่านทางบอร์ด GSM Wireless CPU ด้วยคำสั่ง AT-Command ผ่านทางชุดโปรแกรมซอฟต์แวร์ที่เราพัฒนาขึ้น
3. บันทึกการส่งข้อความสั้นไว้ในฐานข้อมูล
4. แสดงผลการส่งข้อความของข้อความสั้นในแต่ละข้อความเก็บไว้ในฐานข้อมูล

Functional Specification

1. สามารถจัดการข้อมูลกลุ่ม หรือข้อมูลตัวบุคคลและข้อความสั้น ได้
2. สามารถเชื่อมต่อคอมพิวเตอร์ส่วนบุคคลกับบอร์ด GSM Wireless CPU ได้
3. สามารถสั่งให้บอร์ด GSM Wireless CPU ส่งข้อความสั้นไปยังเครื่องโทรศัพท์เคลื่อนที่ที่ระบุไว้ได้

4. สามารถส่งข้อความสั้น ไปยัง โทรศัพท์เคลื่อนที่ทั้งแบบส่งเดี่ยวและส่งเป็นกลุ่มได้

5. บันทึกการส่งข้อความสั้นลงไว้ในฐานข้อมูล และแสดงผลได้

ระบบ SMS Service Package	ระบบ SMS Gateway	ระบบที่เราพัฒนาขึ้น
-ราคาอุปกรณ์สูง -ไม่สามารถตั้งเวลาในการส่งข้อความล่วงหน้าได้ -ไม่สามารถส่งในรูปแบบกลุ่มหรือหลายคนๆ ในเวลาเดียวกันได้ -บางเครือข่ายจำกัดการส่งข้อความเฉพาะเครือข่าย	-สิ้นเปลืองงบประมาณในการเช่าระบบ -จำกัด User ในการเข้าใช้ระบบหรือ ต้องเพิ่มค่าใช้จ่ายเมื่อต้องการเพิ่มจำนวน User -อัตราค่าใช้จ่ายผูกติดกับจำนวนข้อความที่ต้องการส่งหรือ ต้องเช่าเหมาเป็นปี	-ราคาอุปกรณ์ค่อนข้างต่ำเมื่อเทียบกับระบบอื่น -ไม่มีปัญหาเรื่องเครือข่ายในการส่งข้อความสั้น -สามารถตรวจสอบได้ทันที -ค่าใช้จ่ายในการส่ง SMS ค่อนข้างถูก

ตาราง 6.2 ตารางการเปรียบเทียบการส่ง SMS



ตารางสรุปผลการส่งข้อความด้วยระบบ SMS

ว/ค/ป	ข้อความ	จำนวนข้อความที่ส่ง	จำนวนผู้ที่ได้รับ
18/10/2553	ประชุมสาขาวิชา เวลา 13.00น. ณ ห้องE01	235	235
20/10/2553	ป.คณะกรรมการพัฒนาหลักสูตรบัณฑิตศึกษา	8	8
20/10/2553	เวลา 13.00น. เป็นต้นไป ณ ห้องประชุม E01	8	8
03/11/2553	4/11/53 ประชุมหัวหน้าสาขาฯเรื่องรับ นศ. โควต้าปี 54	2	2
03/11/2553	เวลา 12.00น. เป็นต้นไป ณ ห้องประชุมวิชาการ 1	2	2
09/11/2553	12/11/53นศ.ที่เรียนรายวิชาDATACOM,DIGITALCOMให้	100	100
09/11/2553	ไปทำความสะอาดลานย่ำขึ้นรถที่ส่วนกิจฯรถออกเวลา09.00	100	100
11/11/2553	12/11/53 นศ.ที่เรียนรายวิชาDATACOM,DIGITALCOM ให้	100	100
11/11/2553	ไปทำความสะอาดลานย่ำขึ้นรถที่ส่วนกิจฯรถออกเวลา09.00	100	100
12/11/2553	พุ่มนี้ขอเชิญร่วมงานคืนสู่เหย้า โทรคม#9 เวลา18.00น	250	250
12/11/2553	พุ่มนี้ขอเชิญร่วมงานคืนสู่เหย้า โทรคมฯ ครั้งที่9	250	250
12/11/2553	ณ สุรสัมมนาการเวลา 18.30น.และร่วมนำของใช้มาประมูลใน	250	250
12/11/2553	งานเพื่อนำเงินเข้ากองทุนศิษย์เก่า	250	250
16/11/2553	19/11/53 หัวหน้าสาขา โทรคมฯ เรียกพบนศ.ทุกชั้นปี	235	235
16/11/2553	ที่ห้อง B2101 เวลา13.00น. เป็นต้นไป	235	235

ประวัติผู้เขียน



นายณัฐพงศ์ แซ่หู่ เกิดเมื่อวันที่ 29 กุมภาพันธ์ พ.ศ. 2531 ภูมิลำเนาอยู่ที่ ตำบลแสนพัน อำเภอดงหลวง จังหวัดนครพนม สำเร็จการศึกษาระดับมัธยมปลายจากโรงเรียนจุฬาราชวิทยาลัย มุกดาหาร อำเภอเมืองจังหวัดมุกดาหาร เมื่อปี พ.ศ. 2549 ปัจจุบันเป็นนักศึกษาชั้นปีที่ 4 สาขาวิชาวิศวกรรมโทรคมนาคม สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี



นายณัฐวุฒิ ภูงามเงิน เกิดเมื่อวันที่ 25 พฤษภาคม พ.ศ. 2531 ภูมิลำเนาอยู่ที่ ตำบลบัวงาม อำเภอบุนนาค จังหวัดอุบลราชธานี สำเร็จการศึกษาระดับมัธยมปลายจากโรงเรียนบุนนาควิทยาคาร อำเภอบุนนาค จังหวัดอุบลราชธานี เมื่อปี พ.ศ. 2550 ปัจจุบันเป็นนักศึกษาชั้นปีที่ 4 สาขาวิชาวิศวกรรมโทรคมนาคม สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี



นางสาวชฎารัตน์ หาดทวยกาญจน์ เกิดเมื่อวันที่ 18 กันยายน พ.ศ. 2531 ภูมิลำเนาอยู่ที่ ตำบลธาตุเชิงชุม อำเภอเมือง จังหวัดสกลนคร สำเร็จการศึกษาระดับมัธยมปลายจากโรงเรียนสกลราชวิทยานุกูล อำเภอเมืองจังหวัดสกลนคร เมื่อปี พ.ศ. 2549 ปัจจุบันเป็นนักศึกษาชั้นปีที่ 4 สาขาวิชาวิศวกรรมโทรคมนาคม สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี



ภาคผนวก ก

ตัวอย่างโปรแกรมในการเชื่อมต่อและตั้งค่าต่างๆ ภายในโปรแกรม

```

using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using GsmComm.GsmCommunication;
namespace SMS
{
    public class frmConnection : System.Windows.Forms.Form{
    public frmConnection(){
        InitializeComponent();
    }
    protected override void Dispose( bool disposing ){
        if( disposing ){
            if(components != null){
                components.Dispose();
            }
        }
        base.Dispose( disposing );
    }
    #region Windows Form Designer generated code
    private void InitializeComponent(){
        System.ComponentModel.ComponentResourceManager resources = new
        System.ComponentModel.ComponentResourceManager(typeof(frmConnection));
    }
    #endregion
}

```

ใช้ในการอ้างอิง namespace


```

public void SetData(int port, int baudRate, int timeout){
    this.port = port;
    this.baudRate = baudRate;
    this.timeout = timeout;
}
// กำหนดตัวเลือก Text ในการเลือกพอร์ต , บอร์ดเรท,
// ไทม์เอาต์
}

public void GetData(out int port, out int baudRate, out int timeout){
    port = this.port; // กำหนดชื่อ Textbox ของพอร์ต
    baudRate = this.baudRate; // กำหนดชื่อ Textbox ของบอร์ดเรท
    timeout = this.timeout; // กำหนดชื่อ Textbox ของ ไทม์เอาต์
}

private bool EnterNewSettings(){
    int newPort; // กำหนดค่าใน Textbox (newPort) จากตัวอักษรให้เป็นตัวเลข
    int newBaudRate; // กำหนดค่าใน Textbox (newBaudRate) จากตัวอักษรให้เป็นตัวเลข
    int newTimeout; // กำหนดค่าใน Textbox (newTimeout) จากตัวอักษรให้เป็นตัวเลข
    try {
        // ทำการแปลงค่าใน Textbox (newPort) จากตัวอักษรให้เป็นตัวเลข
        newPort = int.Parse(cboPort.Text);
    }
    catch(Exception){
        // ทำการแจ้งเตือนว่าไม่พบพรีตที่ท่านเลือกไว้ใน Textbox
        MessageBox.Show(this, "Invalid port number.", "Error", MessageBoxButtons.OK,
        MessageBoxIcon.Exclamation);
        cboPort.Focus();
        return false;
    }

    try {
        // ทำการแปลงค่าใน Textbox (newBaudRate) จากตัวอักษรให้เป็นตัวเลข
        newBaudRate = int.Parse(cboBaudRate.Text);
    }
    catch(Exception){

```

```

// ทำการแจ้งเตือนว่าไม่พบค่าบอรรตที่ท่านเลือกไว้ใน Textbox
MessageBox.Show(this, "Invalid baud rate.", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
cboBaudRate.Focus();
return false;
}

try {
// ทำการแปลงค่าใน Textbox (cboTimeout) จากชนิดตัวอักษรให้เป็นตัวเลข
newTimeout = int.Parse(cboTimeout.Text);
}
catch(Exception){
// ทำการแจ้งเตือนว่าไม่พบค่าไทม์เอาต์ที่ท่านเลือกไว้ใน Textbox
MessageBox.Show(this, "Invalid timeout value.", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
cboTimeout.Focus();
return false;
}
//ทำการติดตั้งพอร์ต, บอรรต และ ไทม์เอาต์ที่เลือกไว้
SetData(newPort,newBaudRate,newTimeout);
return true;
}

//กำหนดค่าการกระทำเมื่อทำการคลิกที่ปุ่ม btnOK
private void btnOK_Click(object sender, System.EventArgs e){
if (!EnterNewSettings())
DialogResult = DialogResult.None;
}

```

```

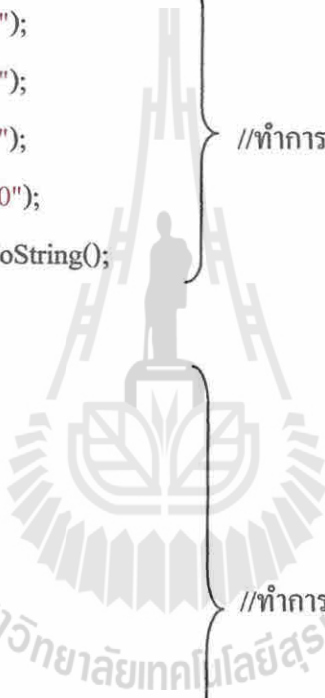
private void frmConnection_Load(object sender, System.EventArgs e) {
    cboPort.Items.Add("1");
    cboPort.Items.Add("2");
    cboPort.Items.Add("3");
    cboPort.Items.Add("4");
    cboPort.Text = port.ToString();

    cboBaudRate.Items.Add("9600");
    cboBaudRate.Items.Add("19200");
    cboBaudRate.Items.Add("38400");
    cboBaudRate.Items.Add("57600");
    cboBaudRate.Items.Add("115200");
    cboBaudRate.Text = baudRate.ToString();

    cboTimeout.Items.Add("150");
    cboTimeout.Items.Add("300");
    cboTimeout.Items.Add("600");
    cboTimeout.Items.Add("900");
    cboTimeout.Items.Add("1200");
    cboTimeout.Items.Add("1500");
    cboTimeout.Items.Add("1800");
    cboTimeout.Items.Add("2000");
    cboTimeout.Text = timeout.ToString();
}

// การกระทำของ โปรแกรมเมื่อทำการคลิกปุ่มตรวจสอบ
private void btnTest_Click(object sender, System.EventArgs e) {
    if (!EnterNewSettings())
        return;
    Cursor.Current = Cursors.WaitCursor;
}

```



```

// ทำการติดตั้งค่าของ port, baudRate และ timeout ตามลำดับ
GsmCommMain comm = new GsmCommMain(port, baudRate, timeout);

try
{
    comm.Open();
    while (!comm.IsConnected())
    {
        Cursor.Current = Cursors.Default;
        // ทำการแจ้งเตือนผู้ใช้งานกรณีที่โปรแกรมหาอุปกรณ์ที่ผู้ใช้เลือกไม่เจอ
        if (MessageBox.Show(this, "ไม่พบอุปกรณ์ กรุณาตรวจสอบอีกครั้ง", "Connection setup",
            MessageBoxButtons.RetryCancel, MessageBoxIcon.Exclamation) == DialogResult.Cancel)
        {
            comm.Close();
            return;
        }
        Cursor.Current = Cursors.WaitCursor;
    }
    comm.Close();
}
catch (Exception ex)
{
    // ทำการแจ้งเตือนผู้ใช้งานกรณีที่โปรแกรมไม่สามารถติดต่อกับอุปกรณ์ได้
    MessageBox.Show(this, "เกิดความผิดพลาดในการติดต่อ: " + ex.Message, "Connection setup",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
    return;
}
// ทำการแจ้งเตือนผู้ใช้งานกรณีที่โปรแกรมทำการติดต่อกับอุปกรณ์ได้สำเร็จ
MessageBox.Show(this, "การตั้งค่าอุปกรณ์เสร็จสมบูรณ์แล้ว", "Connection setup",
    MessageBoxButtons.OK, MessageBoxIcon.Information);
}
}
}

```

ภาคผนวก ข

ตัวอย่างโปรแกรมในการส่งข้อความสั้น หรือ SMS

```

using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using GsmComm.PduConverter; // ใช้ในการอ้างอิง namespace
using GsmComm.GsmCommunication;
using System.Data.SqlClient;
using System.Text;
using System.Data;
namespace SMS
{
    public class Send : System.Windows.Forms.Form
    {
        DataSet ds = new DataSet();
        // ทำการเรียกใช้ฐานข้อมูล SQL ชื่อฐานข้อมูลคือ phone_p
        SqlConnection cs = new SqlConnection("Data Source=TRUEFASTER; Initial Catalog=phone_p;
        Integrated Security=TRUE");
        // ทำการกำหนดเคต้าเซตของระบบ
        SqlDataAdapter da = new SqlDataAdapter();
        // ทำการตั้งชื่อตัวเชื่อมโยงฐานข้อมูล
        BindingSource tbInNamesBS = new BindingSource();
        public Send() {
            InitializeComponent();
        }
        protected override void Dispose( bool disposing ) {

```

```

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

    base.Dispose( disposing );
}

#region Windows Form Designer generated code
private void InitializeComponent() {
    this.components = new System.ComponentModel.Container();
    System.ComponentModel.ComponentResourceManager resources = new
    System.ComponentModel.ComponentResourceManager(typeof(Send));
    System.Windows.Forms.DataGridViewCellStyle dataGridViewCellStyle1 = new
    System.Windows.Forms.DataGridViewCellStyle();
}
#endregion

// กำหนดการกระทำเมื่อทำการเปลี่ยนแปลงข้อมูลของ txt_text_remaining
private void textBox1_TextChanged(object sender, System.EventArgs e){
    // กำหนดค่าการเปลี่ยนแปลงเมื่อมีการพิมพ์ตัวอักษร โดยค่านั้นจะแสดงตรง textBox1
    int remaining=int.Parse(txt_text_remaining.Text.Trim());
    // ทำการลดค่าใน textBox1 ทีละค่าเมื่อมีการป้อนหรือกระทำการเปลี่ยนค่าใน txt_text_remaining
    remaining-=1;
    // แสดงจำนวนตัวอักษรใน textBox1
    txt_text_remaining.Text=remaining.ToString();
}

```

```

private void BtnClear_Click(object sender, System.EventArgs e) {
    txt_message.Text="";
    textBox6.Text = "";
    txt_destination_numbers.Text = "";
    txt_message.Focus();
    txt_text_remaining.Text = "160";
}
// กำหนดการกระทำเมื่อมีการคลิกที่ปุ่ม btnSendMessage
private void btnSendMessage_Click(object sender, System.EventArgs e) {
    Cursor.Current = Cursors.WaitCursor;
    Try {
        // ทำการเปลี่ยนค่าของตัวอักษรให้อยู่ใน โหมด pdu
        SmsSubmitPdu pdu;
        // ตรวจสอบค่าที่ chkAlert ว่าได้มีการทำเครื่องหมายถูกหรือไม่
        bool alert = chkAlert.Checked;
        // ตรวจสอบค่าที่ chkUnicode ว่าได้มีการทำเครื่องหมายถูกหรือไม่
        bool unicode = chkUnicode.Checked;
        // ตรวจสอบว่าถ้าไม่ได้ทำการทำเครื่องหมายถูกตรงที่ alert และไม่ได้ทำเครื่องหมายถูกตรงที่
        unicode ถ้าตรงตามเงื่อนไขข้างต้นให้ทำตามคำสั่งข้างล่างนี้
        if (!alert && !unicode) {
            pdu = new SmsSubmitPdu(txt_message.Text, txt_destination_numbers.Text, "");
        }
        Else
        {
            byte dcs;
            // ตรวจสอบว่าถ้าไม่ได้ทำการทำเครื่องหมายถูกตรงที่ alert และได้ทำเครื่องหมายถูกตรงที่ unicode
            ถ้าตรงตามเงื่อนไขข้างต้นให้ทำตามคำสั่งข้างล่างนี้
            if (!alert && unicode)
                dcs = DataCodingScheme.NoClass_16Bit;
        }
    }
}

```

```

// ตรวจสอบว่าถ้าได้ทำการทำเครื่องหมายถูกต้องที่ alert และได้ทำเครื่องหมายถูกต้องที่ unicode ถ้า
ตรงตามเงื่อนไขข้างต้นให้ทำตามคำสั่งข้างล่างนี้
else if (alert && !unicode)
    dcs = DataCodingScheme.Class0_7Bit;
else if (alert && unicode)
    dcs = DataCodingScheme.Class0_16Bit;
else
    dcs = DataCodingScheme.NoClass_7Bit;
// เมื่อทำการเปลี่ยนค่าต่างๆ ตามข้างต้นแล้วก็ให้ทำการเปลี่ยนค่าของข้อความให้อยู่ใน โหมด pdu
จากค่าที่อยู่ใน dcs
pdu = new SmsSubmitPdu(txt_message.Text, txt_destination_numbers.Text, "", dcs);
}
// ตรวจสอบการส่งว่าได้มีการกำหนดจำนวนการส่งข้อความไว้กี่ครั้ง
int times = chkMultipleTimes.Checked ? int.Parse(txtSendTimes.Text) : 1;
// ทำการวนลูปตามจำนวนที่กำหนดไว้
for (int i=0;i<times;i++){
    CommSetting.comm.SendMessage(pdu);
    Output("ข้อความที่ {0} จาก {1} ครั้ง", i+1, times);
    Output("");
    // นำค่าต่างๆ ที่ได้ทำการส่งไปทั้งหมดไปเก็บไว้ที่ฐานข้อมูลเพื่อทำการตรวจสอบรายงานการส่ง
    อีกครั้ง
    send_db();
    // ทำการแจ้งเตือนว่าได้ทำการส่งข้อความเสร็จสมบูรณ์แล้ว
    MessageBox.Show("ระบบได้ทำการส่งข้อความเรียบร้อยแล้ว");
}
}
catch(Exception ex)
{
    MessageBox.Show(ex.Message);
}
}

```



```

Cursor.Current = Cursors.Default;
}
// ทำการแจ้งเตือนความผิดพลาดเมื่อไม่สามารถทำการส่งข้อความสั้นได้และรายงานถึงสาเหตุที่ทำให้
// การส่งข้อความสั้นไม่ได้ไปด้วย
private void Output(string text) {
    if (this.txtOutput.InvokeRequired) {
        SetTextCallback stc = new SetTextCallback(Output);
        this.Invoke(stc, new object[] { text });
    }
    else {
        txtOutput.AppendText(text);
        txtOutput.AppendText("\r\n");
    }
}

private void Send_Load(object sender, System.EventArgs e) {
    this.t_send_groupTableAdapter.Fill(this.phone_pDataSet12.t_send_group);
    this.m_groupTableAdapter.Fill(this.phone_pDataSet7.M_group);
    this.m_groupTableAdapter.Fill(this.phone_pDataSet7.M_group);
    this.m_membersTableAdapter.Fill(this.phone_pDataSet3.M_members);
    // ทำการตรวจสอบว่าที่ช่องของ chkMultipleTimes ว่าได้ทำเครื่องหมายถูกไว้หรือไม่
    chkMultipleTimes.Checked=true;
    // ทำการตรวจสอบว่าที่ช่องของ chkUnicode ว่าได้ทำเครื่องหมายถูกไว้หรือไม่
    chkUnicode.Checked = true;
    comboBox1.Show();
    label6.Show();
    this.tabControl1.TabPages.Remove(tabPage1);
    this.tabControl1.TabPages.Remove(tabPage2);
    textBox25.Hide();
}

```

```

// ทำการติดต่อกับฐานข้อมูล phone_p โดยเลือกติดต่อกับตาราง members
da.SelectCommand = new SqlCommand("SELECT * FROM members Order by Name desc", cs);
// ทำการเคลียร์ค่าต่างๆ ในตารางข้อมูลทิ้งทั้งหมด
ds.Clear();
// ทำการตรวจสอบข้อมูลที่ได้อิงมาจากตาราง members ในฐานข้อมูล phone_p
da.Fill(ds);
// ทำการดึงข้อมูลจากฐานข้อมูล phone_p จากตาราง members มาที่ละแถว โดยนำมาแสดงที่ตาราง
dg ที่ละบรรทัดจนข้อมูลในฐานข้อมูลครบตามที่มีการบันทึกไว้ในตาราง members
dg.DataSource = ds.Tables[0];
textBox12.Text = comboBox5.Text;
textBox13.Text = comboBox3.Text;
textBox18.Text = textBox21.Text;
// ตั้งค่าให้เลือกชั่วโมง
comboBox6.Items.Add("0");
comboBox6.Items.Add("1");
comboBox6.Items.Add("2");
comboBox6.Items.Add("3");
comboBox6.Items.Add("4");
comboBox6.Items.Add("5");
comboBox6.Items.Add("6");
comboBox6.Items.Add("7");
comboBox6.Items.Add("8");
comboBox6.Items.Add("9");
comboBox6.Items.Add("10");
comboBox6.Items.Add("11");
comboBox6.Items.Add("12");
comboBox6.Items.Add("13");
comboBox6.Items.Add("14");
comboBox6.Items.Add("15");
comboBox6.Items.Add("16");

```

```
comboBox6.Items.Add("17");
comboBox6.Items.Add("18");
comboBox6.Items.Add("19");
comboBox6.Items.Add("20");
comboBox6.Items.Add("21");
comboBox6.Items.Add("22");
comboBox6.Items.Add("23");
//ตั้งค่าให้เลือกนาฬิกา
comboBox4.Items.Add("00");
comboBox4.Items.Add("01");
comboBox4.Items.Add("02");
comboBox4.Items.Add("03");
comboBox4.Items.Add("04");
comboBox4.Items.Add("05");
comboBox4.Items.Add("06");
comboBox4.Items.Add("07");
comboBox4.Items.Add("08");
comboBox4.Items.Add("09");
comboBox4.Items.Add("10");
comboBox4.Items.Add("11");
comboBox4.Items.Add("12");
comboBox4.Items.Add("13");
comboBox4.Items.Add("14");
comboBox4.Items.Add("15");
comboBox4.Items.Add("16");
comboBox4.Items.Add("17");
comboBox4.Items.Add("18");
comboBox4.Items.Add("19");
comboBox4.Items.Add("20");
comboBox4.Items.Add("21");
```



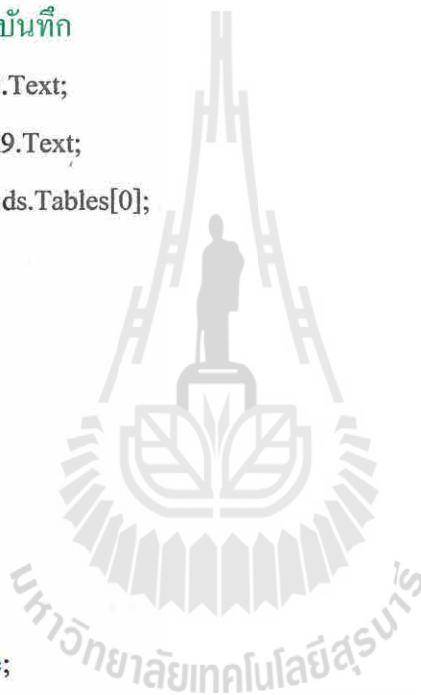
```
comboBox4.Items.Add("22");  
comboBox4.Items.Add("23");  
comboBox4.Items.Add("24");  
comboBox4.Items.Add("25");  
comboBox4.Items.Add("26");  
comboBox4.Items.Add("27");  
comboBox4.Items.Add("28");  
comboBox4.Items.Add("29");  
comboBox4.Items.Add("30");  
comboBox4.Items.Add("31");  
comboBox4.Items.Add("32");  
comboBox4.Items.Add("33");  
comboBox4.Items.Add("34");  
comboBox4.Items.Add("35");  
comboBox4.Items.Add("36");  
comboBox4.Items.Add("37");  
comboBox4.Items.Add("38");  
comboBox4.Items.Add("39");  
comboBox4.Items.Add("40");  
comboBox4.Items.Add("41");  
comboBox4.Items.Add("42");  
comboBox4.Items.Add("43");  
comboBox4.Items.Add("44");  
comboBox4.Items.Add("45");  
comboBox4.Items.Add("46");  
comboBox4.Items.Add("47");  
comboBox4.Items.Add("48");  
comboBox4.Items.Add("49");  
comboBox4.Items.Add("50");  
comboBox4.Items.Add("51");
```



```

comboBox4.Items.Add("52");
comboBox4.Items.Add("53");
comboBox4.Items.Add("54");
comboBox4.Items.Add("55");
comboBox4.Items.Add("56");
comboBox4.Items.Add("57");
comboBox4.Items.Add("58");
comboBox4.Items.Add("59");
//โหลดชื่อกับเบอร์โทรตรงบันทึก
textBox24.Text = textBox7.Text;
    textBox19.Text = textBox9.Text;
tblNamesBS.DataSource = ds.Tables[0];
ClearIt();
records();
FormatDg();
//แสดงเวลาปัจจุบัน
timer1.Interval = 1000;
timer1.Enabled = true;
timer1.Start();
comboBox1.Enabled = true;
//=====
comboBox1.Enabled = false;
textBox7.Enabled = false;
textBox8.Enabled = false;
textBox9.Enabled = false;
button2.Enabled = false;
button3.Enabled = false;
}

```



```

private void Output(string text, params object[] args){
    string msg = string.Format(text, args);
    Output(msg);
}

// กำหนดการกระทำเมื่อได้มีการคลิกที่ปุ่ม btnFrist โดยผลการกระทำที่เกิดขึ้นก็จะแสดงข้อมูลใน
ฐานข้อมูลในแถวแรกสุดออกมา
private void btnFrist_Click(object sender, EventArgs e)    {
    tblNamesBS.MoveFirst();
    dgUpdate();
    records();
}

// กำหนดการกระทำเมื่อได้มีการคลิกที่ปุ่ม btnPrevious โดยผลการกระทำที่เกิดขึ้นก็จะแสดง
ข้อมูลในฐานข้อมูลในแถวก่อนหน้าออกมา
private void btnPrevious_Click(object sender, EventArgs e)    {
    tblNamesBS.MovePrevious();
    dgUpdate();
    records();
}

// กำหนดการกระทำเมื่อได้มีการคลิกที่ปุ่ม btnNext โดยผลการกระทำที่เกิดขึ้นก็จะแสดงข้อมูลใน
ฐานข้อมูลในแถวถัดไปออกมา
private void btnNext_Click(object sender, EventArgs e)    {
    tblNamesBS.MoveNext();
    dgUpdate();
    records();
}

// กำหนดการกระทำเมื่อได้มีการคลิกที่ปุ่ม btnLast โดยผลการกระทำที่เกิดขึ้นก็จะแสดงข้อมูลใน
ฐานข้อมูลในแถวท้ายสุดออกมา
private void btnLast_Click(object sender, EventArgs e)    {
    tblNamesBS.MoveLast();
    dgUpdate();
}

```

```

        records();
    }

    private void dgUpdate()    {
        dg.ClearSelection();
        dg.Rows[tblNamesBS.Position].Selected = true;
        records();
    }

    private void records()    {
        label21.Text = "ลำดับที่ " + (tblNamesBS.Position) + " จาก " + ((tblNamesBS.Count - 1));
    }
    // กำหนดคอลัมน์ต่างๆ ในตาราง dg
    private void FormatDg()    {
        if (dg.RowCount > 0) {
            dg.Columns[0].HeaderText = "ชื่อผู้ติดต่อ";
            dg.Columns[1].HeaderText = "เบอร์ติดต่อ";
            dg.Columns[2].HeaderText = "กลุ่ม";
            dg.Columns[0].Width = 135;
            dg.Columns[1].Width = 80;
            dg.Columns[2].Width = 100;
        }
    }

    // กำหนดการเปลี่ยนค่าใน txt_destination_numbers.Text
    private void Cleartt()    {
        txt_destination_numbers.Text = "";
    }

    // กำหนดการกระทำต่างๆ เมื่อได้มีการคลิกที่ปุ่ม button1
    private void button1_Click(object sender, EventArgs e)    {
        comboBox1.Show();
        Cursor.Current = Cursors.WaitCursor;

```

```

// ทำการตรวจสอบค่าใน txt_destination_numbers ว่าได้มีการพิมพ์เบอร์โทรศัพท์ไว้หรือไม่
txt_destination_numbers.DataBindings.Add(new Binding("Text", tblNamesBS, "Phone"));
textBox1.DataBindings.Add(new Binding("Text", tblNamesBS, "Name"));
records();
FormatDg();
}

private void dg_CellClick(object sender, DataGridViewCellEventArgs e) {
if (e.RowIndex == -1) {
return;
}
else {
int rowIndex = e.RowIndex;
if (textBox7.Enabled == false)
{
textBox6.Text = dg.Rows[rowIndex].Cells[0].Value.ToString();
txt_destination_numbers.Text = dg.Rows[rowIndex].Cells[1].Value.ToString();
textBox7.Text = "";
textBox9.Text = "";
}
else {
textBox7.Text = dg.Rows[rowIndex].Cells[0].Value.ToString();
textBox9.Text = dg.Rows[rowIndex].Cells[1].Value.ToString();
textBox6.Text = "";
txt_destination_numbers.Text = "";
}
}
}
}

```



```

// ทำการกำหนดค่าต่าง ๆ เมื่อมีการคลิกที่ปุ่ม button2
private void button2_Click(object sender, EventArgs e) {
    for (int i=0; i<((tblNamesBS.Count - 1)); i++) {
        if (tblNamesBS.Position <= (tblNamesBS.Count - 1)) {
            Cursor.Current = Cursors.WaitCursor;
        }
    }
    // ทำการแปลงค่าตัวอักษรให้อยู่ในโหมด pdf
    SmsSubmitPdu pdu;
    bool alert = chkAlert.Checked;
    bool unicode = chkUnicode.Checked;
    if (!alert && !unicode) {
        pdu = new SmsSubmitPdu(textBox8.Text, textBox9.Text, "");
    }
    else {
        byte dcs;
        if (!alert && unicode)
            dcs = DataCodingScheme.NoClass_16Bit;
        else if (alert && !unicode)
            dcs = DataCodingScheme.Class0_7Bit;
        else if (alert && unicode)
            dcs = DataCodingScheme.Class0_16Bit;
        else
            dcs = DataCodingScheme.NoClass_7Bit; // should never occur here
        pdu = new SmsSubmitPdu(textBox8.Text, textBox9.Text, "", dcs);
    }
    int times = chkMultipleTimes.Checked ? int.Parse(txtSendTimes.Text) : 1;
    for (int s = 0; s < times; s++) {
        CommSetting.comm.SendMessage(pdu);
        Output("ข้อความที่ {0} จาก {1} ครั้ง", s + 1, times);
        Output("");
    }
}

```

```

send_db_r();
    }
}
catch (Exception ex)    {
    MessageBox.Show(ex.Message);
}
Cursor.Current = Cursors.Default;
btnNext_Click(sender, e);
}
else {    send_db_r();
        button2_Click(sender, e);
    }
}
}
private void timer1_Tick_1(object sender, EventArgs e)    {
    ll4.Text = DateTime.Now.ToLongTimeString();
}
private void send_db()    {
// ทำการเชื่อมต่อกับฐานข้อมูล phone_p
SqlConnection cs = new SqlConnection("Data Source=TRUEFASTER; Initial Catalog=phone_p;
Integrated Security=TRUE");
SqlDataAdapter da = new SqlDataAdapter();
// ทำการเพิ่มข้อมูลต่างๆ ลงในฐานข้อมูล phone_p ตรงตาราง report_send
da.InsertCommand = new SqlCommand("INSERT INTO report_send VALUES(@message,
@name, @phone, @group, @date, @time)", cs);
// ทำการเพิ่มข้อมูล message ในตาราง report_send
da.InsertCommand.Parameters.Add("@message", SqlDbType.VarChar).Value =
txt_message.Text;

```

```

// ทำการเพิ่มข้อมูล name ในตาราง report_send
da.InsertCommand.Parameters.Add("@name", SqlDbType.VarChar).Value = textBox6.Text;
// ทำการเพิ่มข้อมูล phone ในตาราง report_send
da.InsertCommand.Parameters.Add("@phone", SqlDbType.VarChar).Value =
txt_destination_numbers.Text;
// ทำการเพิ่มข้อมูล group ในตาราง report_send
da.InsertCommand.Parameters.Add("@group", SqlDbType.VarChar).Value = comboBox1.Text;
// ทำการเพิ่มข้อมูล date ในตาราง report_send
da.InsertCommand.Parameters.Add("@date", SqlDbType.DateTime).Value =
DateTime.Now.ToString();
// ทำการเพิ่มข้อมูล time ในตาราง report_send
da.InsertCommand.Parameters.Add("@time", SqlDbType.VarChar).Value = ll4.Text;
cs.Open();
da.InsertCommand.ExecuteNonQuery();
cs.Close();
}
// ทำการ เพิ่มข้อมูลลงในตาราง report_send ในฐานะข้อมูล phone_p
private void send_db_r() {
// ทำการเชื่อมต่อกับฐานข้อมูล phone_p
SqlConnection cs = new SqlConnection("Data Source=TRUEFASTER; Initial Catalog=phone_p;
Integrated Security=TRUE");
SqlDataAdapter da = new SqlDataAdapter();
// ทำการเพิ่มข้อมูลต่างๆ ลงในฐานะข้อมูล phone_p ตรงตาราง report_send
da.InsertCommand = new SqlCommand("INSERT INTO report_send VALUES(@message,
@name, @phone, @group, @date, @time)", cs);
// ทำการเพิ่มข้อมูล message ในตาราง report_send
da.InsertCommand.Parameters.Add("@message", SqlDbType.VarChar).Value = textBox8.Text;
// ทำการเพิ่มข้อมูล name ในตาราง report_send
da.InsertCommand.Parameters.Add("@name", SqlDbType.VarChar).Value = textBox7.Text;

```

```

// ทำการเพิ่มข้อมูล phone ในตาราง report_send
da.InsertCommand.Parameters.Add("@phone", SqlDbType.VarChar).Value = textBox9.Text;
// ทำการเพิ่มข้อมูล group ในตาราง report_send
da.InsertCommand.Parameters.Add("@group", SqlDbType.VarChar).Value = comboBox1.Text;
// ทำการเพิ่มข้อมูล date ในตาราง report_send
da.InsertCommand.Parameters.Add("@date", SqlDbType.DateTime).Value =
DateTime.Now.ToString();
// ทำการเพิ่มข้อมูล time ในตาราง report_send
da.InsertCommand.Parameters.Add("@time", SqlDbType.VarChar).Value = ll4.Text;
cs.Open();
da.InsertCommand.ExecuteNonQuery();
cs.Close();
}
private void dg_CellDoubleClick(object sender, DataGridViewCellEventArgs e) {
if (e.RowIndex == -1) {
return;
}
else {
int rowIndex = e.RowIndex;
textBox6.Text = dg.Rows[rowIndex].Cells[0].Value.ToString();
txt_destination_numbers.Text = dg.Rows[rowIndex].Cells[1].Value.ToString();
}
}
// ทำการเพิ่มข้อมูลลงในตาราง t_send_group
da.InsertCommand = new SqlCommand("INSERT INTO t_send_group
VALUES(@phone_num)", cs);
// ทำการเพิ่มข้อมูล phone_num ในตาราง t_send_group
da.InsertCommand.Parameters.Add("@phone_num", SqlDbType.VarChar).Value =
txt_destination_numbers.Text;
cs.Open();
da.InsertCommand.ExecuteNonQuery();

```

```

cs.Close();
}

// แสดงการกระทำต่างๆ เมื่อมีการคลิกที่ปุ่ม button4

private void button4_Click(object sender, EventArgs e) {
    txt_text_remaining.Text = "160";

    //=====

    comboBox1.Enabled = false;
    textBox7.Enabled = false;
    textBox8.Enabled = false;
    textBox9.Enabled = false;
    button2.Enabled = false;
    button3.Enabled = false;

    //=====

    textBox6.Enabled = true;
    txt_destination_numbers.Enabled = true;
    txt_message.Enabled = true;
    btnSendMessage.Enabled = true;
    BtnClear.Enabled = true;

    //=====

    textBox7.Text = "";
    textBox8.Text = "";
    textBox9.Text = "";
}

// ทำการปิด โปรแกรมเมื่อ ได้มีการคลิกที่ปุ่ม button7

private void button7_Click(object sender, EventArgs e) {
    this.Close();
}

```

```

// กำหนดค่าต่างๆ เมื่อได้มีการคลิกที่ปุ่ม button5
private void button5_Click(object sender, EventArgs e) {
    txt_text_remaining.Text = "160";
    //=====
    comboBox1.Enabled = true;
    textBox7.Enabled = true;
    textBox8.Enabled = true;
    textBox9.Enabled = true;
    button2.Enabled = true;
    button3.Enabled = true;
    textBox6.Enabled = false;
    txt_destination_numbers.Enabled = false;
    txt_message.Enabled = false;
    btnSendMessage.Enabled = false;
    BtnClear.Enabled = false;
    //=====
    textBox6.Text = "";
    txt_destination_numbers.Text = "";
    txt_message.Text = "";
    tblNamesBS.DataSource = ds.Tables[0];
    textBox9.DataBindings.Add(new Binding("Text", tblNamesBS, "Phone"));
    textBox7.DataBindings.Add(new Binding("Text", tblNamesBS, "Name"));
    textBox24.Text = textBox7.Text;
    textBox19.Text = textBox9.Text;
}

private void button1_Click_1(object sender, EventArgs e) {
    da.SelectCommand = new SqlCommand("SELECT * FROM members Where M_Group= " +
    comboBox2.Text + "Order by Name desc", cs);
    ds.Clear();
    da.Fill(ds);
}

```

```

dg.DataSource = ds.Tables[0];
tblNamesBS.DataSource = ds.Tables[0];
ClearIt();
records();
FormatDg();
textBox24.Text = textBox7.Text;
textBox19.Text = textBox9.Text;
}
private void button8_Click(object sender, EventArgs e) {
    if (textBox12.Text == "" || textBox13.Text == "" || textBox16.Text == "") {
    }
    else {
        textBox14.Text = DateTime.Now.TimeOfDay.Hours.ToString();
        textBox15.Text = DateTime.Now.TimeOfDay.Minutes.ToString();
        textBox17.Text = DateTime.Now.TimeOfDay.Milliseconds.ToString();
        int xe = int.Parse(textBox14.Text);
        int ye = int.Parse(textBox15.Text);
        int ze = int.Parse(textBox17.Text);
        int xee = int.Parse(textBox12.Text);
        int yee = int.Parse(textBox13.Text);
        int zee = int.Parse(textBox16.Text);
        int aa = xee - xe;
        int ab = yee - ye;
        int Ac = ze - zee;
        label47.Text = aa.ToString();
        label25.Text = ab.ToString();
        label27.Text = Ac.ToString();
        float cc = (((aa * 60 * 60) + (ab * 60)) * 10) - Ac;
        textBox11.Text = cc.ToString();
    }
}

```

```

}

private void button10_Click_1(object sender, EventArgs e)    {
    btnCountDown_Click(sender, e);
}

private void button6_Click(object sender, EventArgs e)    {
    report_send send_sms = new report_send();
    send_sms.Show();
}

private void save_time()    {
    SqlConnection cs = new SqlConnection("Data Source=TRUEFASTER; Initial Catalog=phone_p;
    Integrated Security=TRUE");
    SqlDataAdapter da = new SqlDataAdapter();
    da.InsertCommand = new SqlCommand("INSERT INTO send_time VALUES(@message,
    @phone_num, @send_to, @time, @date, @group)", cs);
    da.InsertCommand.Parameters.Add("@message", SqlDbType.VarChar).Value = textBox23.Text;
    da.InsertCommand.Parameters.Add("@phone_num", SqlDbType.VarChar).Value =
    textBox19.Text;
    da.InsertCommand.Parameters.Add("@send_to", SqlDbType.VarChar).Value = textBox24.Text;
    da.InsertCommand.Parameters.Add("@time", SqlDbType.VarChar).Value = textBox25.Text;
    da.InsertCommand.Parameters.Add("@date", SqlDbType.VarChar).Value =
    dateTimePicker2.Text;
    da.InsertCommand.Parameters.Add("@group", SqlDbType.VarChar).Value = comboBox7.Text;
    cs.Open();
    da.InsertCommand.ExecuteNonQuery();
    cs.Close();
}

private void button9_Click_1(object sender, EventArgs e)    {
    textBox24.Text = textBox7.Text;
    textBox19.Text = textBox9.Text;
    if (comboBox7.Text == "ไม่มีรายชื่อ")    {

```



```

textBox25.Text = "" + comboBox6.Text + ":" + comboBox4.Text + ":" + label7.Text + "";
save_time();
}

else {

textBox25.Text = "" + comboBox6.Text + ":" + comboBox4.Text + ":" + label7.Text + "";
save_time();

btnNext_Click(sender, e);
// ทำการตรวจสอบค่าและทำการวนลูปเพื่อเปลี่ยนค่าข้อมูลตามที่เรากำหนด
for (int i = 0; i < ((tblNamesBS.Count - 1)); i++) {
if (tblNamesBS.Position <= (tblNamesBS.Count - 1)) {
textBox25.Text = "" + comboBox6.Text + ":" + comboBox4.Text + ":" + label7.Text + "";
textBox24.Text = textBox7.Text;
textBox19.Text = textBox9.Text;
if (int.Parse(label7.Text) <= 60) {
int xet = int.Parse(label7.Text);
int aat = xet + 10;
label7.Text = aat.ToString();
}

if (label7.Text == "1") {
label7.Text = "01";
textBox25.Text = "" + comboBox6.Text + ":" + comboBox4.Text + ":" + label7.Text + "";
}

if (label7.Text == "2") {
label7.Text = "02";
textBox25.Text = "" + comboBox6.Text + ":" + comboBox4.Text + ":" + label7.Text + "";
}

if (label7.Text == "3") {
label7.Text = "03";
textBox25.Text = "" + comboBox6.Text + ":" + comboBox4.Text + ":" + label7.Text + "";
}
}
}

```

```
if (label7.Text == "4") {
label7.Text = "04";
textBox25.Text = "" + comboBox6.Text + ":" + comboBox4.Text + ":" + label7.Text + "";
}
if (label7.Text == "5") {
label7.Text = "05";
textBox25.Text = "" + comboBox6.Text + ":" + comboBox4.Text + ":" + label7.Text + "";
}
if (label7.Text == "6") {
label7.Text = "06";
textBox25.Text = "" + comboBox6.Text + ":" + comboBox4.Text + ":" + label7.Text + "";
}
if (label7.Text == "7") {
label7.Text = "07";
textBox25.Text = "" + comboBox6.Text + ":" + comboBox4.Text + ":" + label7.Text + "";
}
if (label7.Text == "8") {
label7.Text = "08";
textBox25.Text = "" + comboBox6.Text + ":" + comboBox4.Text + ":" + label7.Text + "";
}
if (label7.Text == "9") {
label7.Text = "09";
textBox25.Text = "" + comboBox6.Text + ":" + comboBox4.Text + ":" + label7.Text + "";
}
else{
textBox25.Text = "" + comboBox6.Text + ":" + comboBox4.Text + ":" + label7.Text + "";
}
}
if (int.Parse(label7.Text) > 9) {
```

```

label7.Text = "00";

int xe = int.Parse(comboBox4.Text);

int aa = xe + 01;

comboBox4.Text = aa.ToString();
}

if (comboBox4.Text == "0") {
comboBox4.Text = "00";

textBox25.Text = "" + comboBox6.Text + ":" + comboBox4.Text + ":" + label7.Text + "";
}

if (comboBox4.Text == "1") {
comboBox4.Text = "01";

textBox25.Text = "" + comboBox6.Text + ":" + comboBox4.Text + ":" + label7.Text + "";
}

if (comboBox4.Text == "2") {
comboBox4.Text = "02";

textBox25.Text = "" + comboBox6.Text + ":" + comboBox4.Text + ":" + label7.Text + "";
}

if (comboBox4.Text == "3") {
comboBox4.Text = "03";

textBox25.Text = "" + comboBox6.Text + ":" + comboBox4.Text + ":" + label7.Text + "";
}

if (comboBox4.Text == "4") {
comboBox4.Text = "04";

textBox25.Text = "" + comboBox6.Text + ":" + comboBox4.Text + ":" + label7.Text + "";
}

if (comboBox4.Text == "5") {
comboBox4.Text = "05";

textBox25.Text = "" + comboBox6.Text + ":" + comboBox4.Text + ":" + label7.Text + "";
}

if (comboBox4.Text == "6") {

```

```

label7.Text = "06";
textBox25.Text = "" + comboBox6.Text + ":" + comboBox4.Text + ":" + label7.Text + "";
}
if (comboBox4.Text == "7") {
comboBox4.Text = "07";
textBox25.Text = "" + comboBox6.Text + ":" + comboBox4.Text + ":" + label7.Text + "";
}
if (comboBox4.Text == "8") {
comboBox4.Text = "08";
textBox25.Text = "" + comboBox6.Text + ":" + comboBox4.Text + ":" + label7.Text + "";
}
if (comboBox4.Text == "9") {
comboBox4.Text = "09";
comboBox4.Text = "" + comboBox6.Text + ":" + comboBox4.Text + ":" + label7.Text + "";
}

else {
textBox25.Text = "" + comboBox6.Text + ":" + comboBox4.Text + ":" + label7.Text + "";
}

if (comboBox4.Text == "60") {
label7.Text = "00";
int xee = int.Parse(comboBox6.Text);
int aaa = xee + 01;
comboBox6.Text = aaa.ToString();
textBox25.Text = "" + comboBox6.Text + ":" + comboBox4.Text + ":" + label7.Text + "";
}

if (comboBox6.Text == "24") {
comboBox6.Text = "0";
textBox25.Text = "" + comboBox6.Text + ":" + comboBox4.Text + ":" + label7.Text + "";
}

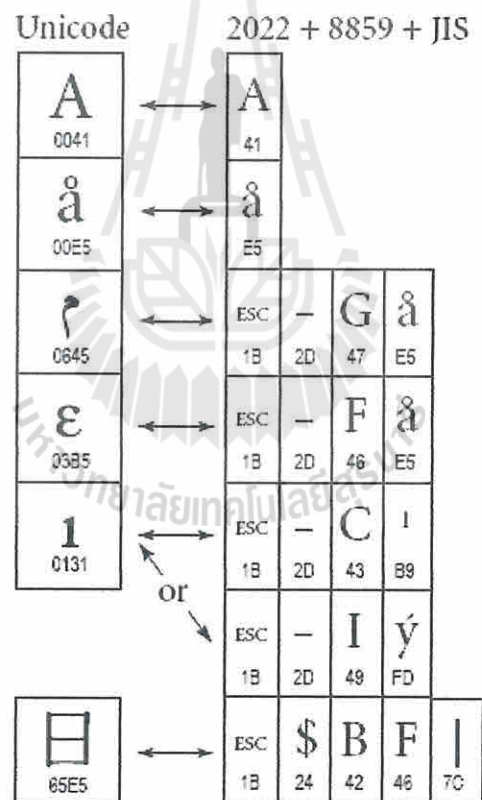
```

```
if (textBox24.Text == "" && textBox19.Text == "") {  
    }  
    save_time();  
    btnNext_Click(sender, e);  
    }  
    }  
    }  
  
private void button11_Click(object sender, EventArgs e) {  
    textBox25.Text = "" + comboBox6.Text + ":" + comboBox4.Text + ":" + label7.Text + "";  
    save_time();  
    MessageBox.Show("ระบบได้ทำการบันทึกการตั้งค่าการส่งข้อความเรียบร้อยแล้ว");  
    }  
  
private void button12_Click(object sender, EventArgs e) {  
    send_time send_sms = new send_time();  
    send_sms.Show();  
    }  
    }  
    }
```

ภาคผนวก ก

หลักการทํางานของ Unicode

- แต่ละตัวอักษรที่ใช้ในภาษาต่างๆ จะมีรหัสเฉพาะของตน แต่รูปแบบของการเข้ารหัสแบบ Unicode นั้นมีได้หลายแบบ เป็นการแปลงรหัส Unicode ที่กำหนดหรือที่เรียกว่า Unicode transformation format (UTF)
- ใช้จำนวน 32 บิตสำหรับแต่ละอักษร (4 bytes/character) (เรียกว่า UTF-32)
- ตัวอักษรที่ใช้บ่อยๆ เก็บ 16 บิต (UTF-16)



รูปที่ 1 แสดงหลักการในการแปลงค่าตัวอักษรให้อยู่ในรูป Unicode

ตัวอย่างตารางรหัส Unicode ที่เป็นภาษาไทย

ก	ข	ฃ	ค	ค	ฅ	ง	จ	ฉ	ช
161 3585 E01	162 3586 E02	163 3587 E03	164 3588 E04	165 3589 E05	166 3590 E06	167 3591 E07	168 3592 E08	169 3593 E09	170 3594 E0A
ช	ฌ	ญ	ฎ	ฏ	ฐ	ฑ	ฒ	ณ	ด
171 3595 E0B	172 3596 E0C	173 3597 E0D	174 3598 E0E	175 3599 E0F	176 3600 E10	177 3601 E11	178 3602 E12	179 3603 E13	180 3604 E14
ต	ถ	ท	ธ	น	บ	ป	ผ	ฝ	พ
181 3605 E15	182 3606 E16	183 3607 E17	184 3608 E18	185 3609 E19	186 3610 E1A	187 3611 E1B	188 3612 E1C	189 3613 E1D	190 3614 E1E
ฟ	ภ	ม	ย	ร	ฤ	ล	ฬ	ว	ศ
191 3615 E1F	192 3616 E20	193 3617 E21	194 3618 E22	195 3619 E23	196 3620 E24	197 3621 E25	198 3622 E26	199 3623 E27	200 3624 E28
ษ	ส	ห	ฬ	อ	ส	ฯ	ะ	ั	า
201 3625 E29	202 3626 E2A	203 3627 E2B	204 3628 E2C	205 3629 E2D	206 3630 E2E	207 3631 E2F	208 3632 E30	209 3633 E31	210 3634 E32
ำ	ิ	ี	ึ	ื	ุ	ู	ฺ	◻	◻
211 3635 E33	212 3636 E34	213 3637 E35	214 3638 E36	215 3639 E37	216 3640 E38	217 3641 E39	218 3642 E3A	219 3643 E3B	220 3644 E3C
◻	◻	฿	฿	฿	฿	฿	฿	฿	฿
221 3645 E3D	222 3646 E3E	223 3647 E3F	224 3648 E40	225 3649 E41	226 3650 E42	227 3651 E43	228 3652 E44	229 3653 E45	230 3654 E46
็	้	๊	์	๋	์	็	็	็	็
231 3655 E47	232 3656 E48	233 3657 E49	234 3658 E4A	235 3659 E4B	236 3660 E4C	237 3661 E4D	238 3662 E4E	239 3663 E4F	240 3664 E50
๑	๒	๓	๔	๕	๖	๗	๘	๙	๐
241 3665 E51	242 3666 E52	243 3667 E53	244 3668 E54	245 3669 E55	246 3670 E56	247 3671 E57	248 3672 E58	249 3673 E59	250 3674 E5A

