สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การพัฒนาแอปพลิเคชันบนโทรศัพท์มือถือระบบปฏิบัติการแอนดรอยด์เพื่อควบคุมอุปกรณ์ไฟฟ้า

DEVELOPMENT OF APPLICATION ON ANDROID OPERATING SYSTEM IN MOBILE PHONE FOR CONTROLLING ELECTRICAL APPLIANCES

T117519

โดย

นายณัทภพ ศรีไทย

นายปัญญ์ แสงสว่าง

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เลขหมู่.....
เลขทะเบียน..... 117519
วัน,เดือน,ปี..... 5 ส.ค. 2554

โครงงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2553

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาแอปพลิเคชันบนโทรศัพท์มือถือระบบปฏิบัติการแอนดรอยด์เพื่อควบคุมอุปกรณ์ไฟฟ้า

DEVELOPMENT OF APPLICATION ON ANDROID OPERATING SYSTEM IN MOBILE PHONE FOR CONTROLLING ELECTRICAL APPLIANCES

โดย

นายณัทภพ ศรีไทย 50010791

นายปัญญ์ แสงสว่าง 50010946

อาจารย์ที่ปรึกษา

รศ.ดร.ไกรสิน ส่งวัฒนา

โครงงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2553

ผ่านการตรวจชิ้นงานแล้ว

(ลงชื่อ).....ผู้ตรวจ

ผ่านการตรวจรูปเล่มแล้ว

(ลงชื่อ).....ผู้ตรวจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2553

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การพัฒนาแอปพลิเคชันบนโทรศัพท์มือถือระบบปฏิบัติการแอนดรอยด์เพื่อควบคุมอุปกรณ์ไฟฟ้า

DEVELOPMENT OF APPLICATION ON ANDROID OPERATING SYSTEM IN MOBILE PHONE FOR CONTROLLING ELECTRICAL APPLIANCES

ผู้จัดทำ

1. นายณัทภพ ศรีไทย 50010791
2. นายปัญญ์ แสงสว่าง 50010946

..... อาจารย์ที่ปรึกษา

(รศ.ดร. ไกรสิน ส่งวัฒนา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้คงไม่อาจสำเร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือ และความร่วมมือจากหลายๆฝ่ายด้วยกัน บุคคลแรกที่ต้องกล่าวถึงเพราะเป็นส่วนสำคัญที่ทำให้ปริญญานิพนธ์นี้สำเร็จได้ก็คือ รศ.ดร ไกรสิน ส่งวัฒนา อาจารย์ที่ปรึกษาปริญญานิพนธ์ ที่คอยให้คำปรึกษา เอาใจใส่ดูแล ให้คำแนะนำ คอยย้ำเตือนช่วงระยะเวลาและการวางแผนการดำเนินงานมาโดยตลอด และให้ความช่วยเหลือเป็นอย่างดีโดยตลอดมา

ขอขอบพระคุณคณาจารย์ ภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้ประสิทธิ์ประสาทวิชาให้กับข้าพเจ้า

ขอขอบคุณเพื่อนๆ พี่ๆ น้องๆ ทุกคนที่คอยให้ความช่วยเหลือ และเป็นกำลังใจให้กันโดยตลอดมา จนทำให้ปริญญานิพนธ์ฉบับสุล่วงไปได้ด้วยดี

และสุดท้ายต้องขอกราบขอบพระคุณบุคคลที่สำคัญที่สุดอันหาที่ทดแทนไม่ได้ ที่ทำให้ข้าพเจ้ามีวันนี้ได้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักที่สุดของข้าพเจ้า ที่ได้ให้กำเนิดและเลี้ยงดูข้าพเจ้าเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และให้กำลังใจ เอาใจใส่และให้ความช่วยเหลือในทุกๆด้านเป็นอย่างดี อันหาที่เปรียบไม่ได้

ณัทภพ ศรีไทย
ปัญญ์ แสงสว่าง
ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาแอปพลิเคชันบนโทรศัพท์มือถือระบบปฏิบัติการแอนดรอยด์เพื่อควบคุมอุปกรณ์ไฟฟ้า

DEVELOPMENT OF APPLICATION ON ANDROID OPERATING SYSTEM IN MOBILE PHONE FOR CONTROLLING ELECTRICAL APPLIANCES

โดย นายณัทภพ ศรีไทย 50010791
นายปัญญ์ แสงสว่าง 50010946

อาจารย์ที่ปรึกษา รศ.ดร.ไกรสิน ส่งวัฒนา

บทคัดย่อ

โครงงานนี้นำเสนอการประยุกต์ใช้แอพพลิเคชันบนโทรศัพท์มือถือระบบปฏิบัติการแอนดรอยด์เพื่อควบคุมระบบไฟฟ้าโดยส่งข้อมูลสั่งการผ่าน TCP/IP โปรโตคอลไปยังคอมพิวเตอร์เซิฟเวอร์ เพื่อควบคุมไมโครคอนโทรลเลอร์ที่ทำงานร่วมกับแผงวงจรรีเลย์ ในการทำหน้าที่เปิด-ปิดอุปกรณ์ไฟฟ้า

ABSTRACT

This project presents the design of an Android application mobile phone for controlling electrical appliances by sending the command to computer server via TCP/IP protocol for controlling microcontroller to on-off electrical appliances via relay circuit.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

		หน้า
กิตติกรรมประกาศ		I
บทคัดย่อ		II
สารบัญ		III
สารบัญรูป		V
สารบัญตาราง		VII
บทที่ 1	**บทนำ**	**1**
	1.1 ความเป็นมาและความสำคัญของปัญหา	1
	1.2 วัตถุประสงค์	1
	1.3 ขอบเขตของโครงงาน	1
บทที่ 2	**ทฤษฎีและหลักการที่เกี่ยวข้อง**	**2**
	2.1 ระบบปฏิบัติการแอนดรอยด์	2
	2.2 TCP/IP	3
	2.3 ไมโครคอนโทรลเลอร์ PIC16F877	4
	2.4 การเชื่อมต่อสื่อสารแบบอนุกรม	9
	2.5 มาตรฐาน RS-232	13
	2.6 รีเลย์	15
	2.7 หลักการรับส่งข้อความสั้น	17
	2.8 คำสั่ง AT COMMAND กับโทรศัพท์มือถือ	23
บทที่ 3	**การออกแบบและการจัดทำโครงงาน**	**25**
	3.1 การออกแบบ	25
	3.2 เครื่องมือที่ใช้ในการทดลอง	31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

		หน้า
	3.3 การจัดเก็บผลการทดลอง	31
บทที่ 4	**ผลการทดลอง**	**32**
	4.1 ผลการทดลองชุดควบคุมการเปิด-ปิดอุปกรณ์ไฟฟ้าผ่านแอปพลิเคชัน	32
	4.2 ผลการทดลองการรับข้อความสั้นจากจีเอสเอ็ม โมดูล	39
บทที่ 5	**สรุปผลและข้อเสนอแนะ**	**42**
	5.1 สรุปผล	42
	5.2 ข้อเสนอแนะ	42
บรรณานุกรม		**43**
ภาคผนวก ก	ไมโครคอนโทรลเลอร์ PIC16F877	44
ภาคผนวก ข	โปรแกรมภาษาจาวาในการเขียนแอปพลิเคชันแอนดรอยด์	47
ภาคผนวก ค	โปรแกรมภาษา C# สำหรับเครื่องเซิร์ฟเวอร์	54
ภาคผนวก ง	โปรแกรมภาษา C สำหรับไมโครคอนโทรลเลอร์	58
ภาคผนวก จ	จีเอสเอ็ม โมดูล	61

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่

2.1 สัญลักษณ์ของระบบปฏิบัติการแอนดรอยด์ 2

2.2 การจัดขามาตรฐานของไมโครคอนโทรลเลอร์ PIC16F877 8

2.3 การถ่ายโอนข้อมูลโดยใช้การเชื่อมต่อแบบอนุกรม 10

2.4 การสื่อสารแบบอะซิงโครนัสในระดับบิต 10

2.5 การส่งข้อมูลแบบทางเดียว 11

2.6 การส่งข้อมูลแบบกึ่งทางคู่ 11

2.7 การส่งข้อมูลแบบทางคู่ 11

2.8 การส่งข้อมูลที่มีขนาด 7 บิตจากไมโครโปรเซสเซอร์ส่งออกที่ช่องสื่อสารแบบอนุกรม 12

2.9 แสดงตัวอย่าง BIT RATE และ BAUD RATE ของสัญญาณดิจิตอล 12

2.10 คอนเน็คเตอร์ DB-25 แบบตัวผู้และตัวเมีย 14

2.11 คอนเน็คเตอร์ DB-9 แบบตัวผู้และตัวเมีย 14

2.12 รีเลย์กำลัง 15

2.13 รีเลย์ควบคุม 15

2.14 โครงสร้างทั่วไปของรีเลย์ 16

2.15 การทำงานของรีเลย์ 16

2.16 การรับส่งข้อความแบบโหมดข้อความ 17

2.17 การรับส่งข้อความแบบโหมดพีดียู 18

3.1 บล็อกไดอะแกรมแสดงการทำงานของโครงงาน 25

3.2 วงจรรับส่งคำสั่งด้วยไมโครคอนโทรลเลอร์ 25

3.3 วงจรควบคุมอุปกรณ์ไฟฟ้าด้วยรีเลย์ 26

3.4 FLOW CHART การทำงานของไมโครคอนโทรลเลอร์ PIC16F877 27

3.5 FLOW CHART การทำงานของคอมพิวเตอร์เซิร์ฟเวอร์ 28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

หน้า

รูปที่

3.6	FLOW CHART การทำงานของแอปพลิเคชันบนโทรศัพท์มือถือระบบปฏิบัติการแอนดรอยด์	29
3.7	หน้าจอแอปพลิเคชันบนโทรศัพท์มือถือระบบปฏิบัติการแอนดรอยด์	30
4.1	หน้าต่างของโปรแกรมบนคอมพิวเตอร์เซิร์ฟเวอร์ ขณะรอรับค่า	32
4.2	แอปพลิเคชันบนหน้าจอโทรศัพท์มือถือระบบปฏิบัติการแอนดรอยด์	33
4.3	หน้าต่างของโปรแกรมบนคอมพิวเตอร์เซิร์ฟเวอร์ ขณะทำงาน	33
4.4	อุปกรณ์ตัวที่ 1 อยู่ในสถานะ ON	35
4.5	อุปกรณ์ตัวที่ 2 อยู่ในสถานะ ON	35
4.6	อุปกรณ์ตัวที่ 3 อยู่ในสถานะ ON	36
4.7	อุปกรณ์ตัวที่ 4 อยู่ในสถานะ ON	36
4.8	แอปพลิเคชันแสดงกล่องข้อความเมื่อไม่ได้ใส่ IP Address หรือ ใส่ผิด	37
4.9	แอปพลิเคชันแสดงกล่องข้อความเมื่ออุปกรณ์ที่ถูกเลือกให้เปิดนั้นได้ถูกเปิดก่อนหน้า	37
4.10	แอปพลิเคชันแสดงกล่องข้อความเมื่ออุปกรณ์ที่ถูกเลือกให้ปิดนั้นได้ถูกปิดก่อนหน้า	38
4.11	แอปพลิเคชันแสดงกล่องข้อความเมื่อไม่ได้เลือกอุปกรณ์ก่อนทำการออกคำสั่ง	38
4.12	ไอคอนบนอแปพลิเคชันเปลี่ยนแปลงเพื่อเตรียมส่งข้อความเมื่อสัมผัสไอคอน SEND SMS TO MOBILEPHONE	39
4.13	แอปพลิเคชันแสดงกล่องข้อความหลังจากสั่งให้ส่งข้อความสั้นและเปลี่ยนไอคอนกลับคืน	40
4.14	แอปพลิเคชันแสดงกล่องข้อความหลังจากสั่งไม่เลือกการส่งข้อความสั้นและเปลี่ยนไอคอนกลับคืน	40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่

2.1 หน้าที่ของขาสัญญาณของพอร์ต A 6
2.2 หน้าที่ของขาสัญญาณของพอร์ต B 6
2.3 หน้าที่ของขาสัญญาณของพอร์ต C 6
2.4 หน้าที่ของขาสัญญาณของพอร์ต D 7
2.5 หน้าที่ของขาสัญญาณของพอร์ต E 8
2.6 การจัดตำแหน่งขาและหน้าที่ของคอนเน็คเตอร์มาตรฐาน RS-232 14
2.7 ความหมายของรหัสข้อความสั้นที่ได้รับ 18
2.8 ความหมายของรหัสในแต่ละกลุ่มตัวเลข 8 บิต 19
2.9 ชุดของตัวแปรมาตรฐาน GSM 03.38 21
2.10 ตารางรหัสตัวอักษรของการเข้ารหัสแบบ UCS2 (16 บิต) 22
2.11 ลักษณะชุดคำสั่งของ AT+CMGF 23
2.12 ลักษณะชุดคำสั่งของ AT+CMGS 24
2.13 ลักษณะชุดคำสั่งของ AT+CSMS 24
4.1 ผลการทำงานระหว่างเซิร์ฟเวอร์ แอปพลิเคชัน และ สถานะอุปกรณ์ 34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันเทคโนโลยีก้าวหน้าเพิ่มมากขึ้นทำให้โทรศัพท์มือถือไม่ได้มีไว้ เพียงแค่ติดต่อสื่อสารพูดคุยกันเท่านั้น แต่ยังมีการพัฒนาให้สามารถทำงานได้อย่างอิสระมากมาย ไม่ว่าจะเป็นเพื่อความบันเทิง ความรู้ หรืออำนวยความสะดวกให้แก่ผู้ใช้ตลอดทุกเวลาและสถานที่

ซึ่งการจัดทำปริญญานิพนธ์นี้ขึ้น เพื่อนำความสามารถของโทรศัพท์มือถือระบบปฏิบัติการแอนดรอยด์ ที่เป็นระบบปฏิบัติการที่เปิดโอกาสให้ผู้คนสามารถพัฒนาได้อย่างอิสระมาพัฒนาแอปพลิเคชันเพื่อใช้ในการอำนวยความสะดวกสบาย โดยนำมาใช้ควบคุมการเปิด-ปิดอุปกรณ์ไฟฟ้าภายในบ้าน

1.2 วัตถุประสงค์

1) เพื่อศึกษาและพัฒนาแอปพลิเคชันบนโทรศัพท์มือถือระบบปฏิบัติการแอนดรอยด์
2) เพื่อออกแบบและสร้างชุดควบคุมการเปิด-ปิดอุปกรณ์ไฟฟ้าผ่านโปรโตคอล TCP/IP โดยสั่งการผ่านเครือข่ายโทรศัพท์มือถือได้

1.3 ขอบเขตของโครงงาน

สามารถพัฒนาแอปพลิเคชันบนโทรศัพท์มือถือระบบปฏิบัติการแอนดรอยด์ให้ควบคุมการเปิด-ปิดอุปกรณ์ไฟฟ้าโดยส่งคำสั่งผ่านโปรโตคอล TCP/IP ไปยังคอมพิวเตอร์เซิร์ฟเวอร์ที่เชื่อมต่อกับวงจรควบคุมอุปกรณ์ไฟฟ้าด้วยไมโครคอนโทรลเลอร์ และ รีเลย์ ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 ระบบปฏิบัติการแอนดรอยด์

ระบบปฏิบัติการแอนดรอยด์ (Android Operating System) คือ ระบบปฏิบัติการหรือแพลตฟอร์ม ที่จะใช้ควบคุมการทำงานบนอุปกรณ์อิเล็กทรอนิกส์ สำหรับโทรศัพท์มือถือและอุปกรณ์พกพา โดยมี กูเกิล อิงก์, ที-โมบาย, เอชทีซี, ควอลคอมม์, โมโตโรลา และบริษัทชั้นนำอื่นๆร่วมพัฒนาโปรเจ็คแอนดรอยด์ ผ่านกลุ่มพันธมิตรเครื่องมือสื่อสารระบบเปิด (Open Handset Alliance) ซึ่งเป็นกลุ่มพันธมิตรชั้นนำระดับนานาชาติด้านเทคโนโลยีและเครื่องมือสื่อสารเคลื่อนที่ ซึ่ง Android ประกอบด้วยระบบปฏิบัติการ ไลบรารี เฟรมเวิร์ค และซอฟต์แวร์ที่จำเป็นในการพัฒนา ซึ่งเทียบเท่ากับ Windows Mobile, Palm OS, Symbian, OpenMoko และ Maemo ของโนเกีย โดยใช้องค์ประกอบที่เป็นโอเพนซอร์สหลายอย่าง เช่น Linux Kernel, SSL, OpenGL, FreeType, SQLite, WebKit และเขียนไลบรารีเฟรมเวิร์คของตัวเองเพิ่มเติม ซึ่งทั้งหมดจะเป็นโอเพนซอร์สที่ใช้ Apache License

รูปที่ 2.1 สัญลักษณ์ของระบบปฏิบัติการแอนดรอยด์

จุดเด่นของ Android นั้นอยู่ที่การออกแบบระบบปฏิบัติการมาให้มีความสามารถได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใกล้เคียงกับ คอมพิวเตอร์พกพาขนาดย่อมที่ทำงานได้บนโทรศัพท์มือถือ โดยยังคงเรื่องความคล่องตัวในการใช้งานที่ค่อนข้างมาก นอกจากนี้ Google android นั้นเป็นแบบระบบเปิด คือ เปิดโอกาสให้บริษัทและนักพัฒนาสามารถมีส่วนร่วมในการสร้างระบบปฏิบัติการได้ด้วย และยังไม่จำกัดการใช้งานเฉพาะบนโทรศัพท์มือถือจากค่ายใดค่ายหนึ่ง นั่นทำให้เราจะได้เห็น Android จากผู้ผลิตโทรศัพท์มือถือเกือบทุกราย ต่างจาก Apple iPhone หรือ BlackBerry ที่ใช้ระบบปิด ทำให้มีโทรศัพท์มือถือออกมาจากค่ายใดค่ายหนึ่งเพียงค่ายเดียว

2.2 TCP/IP

TCP/IP เป็นชุดของโปรโตคอลที่ถูกใช้ในการสื่อสารผ่านเครือข่ายอินเตอร์เน็ต โดยมีวัตถุประสงค์เพื่อให้สามารถใช้สื่อสารจากต้นทางข้ามเครือข่ายไปยังปลายทางได้ และสามารถหาเส้นทางที่จะส่งข้อมูลไปได้เองโดยอัตโนมัติ ถึงแม้ว่าในระหว่างทางอาจจะผ่านเครือข่ายที่มีปัญหา โปรโตคอลก็ยังคงหาเส้นทางอื่นในการส่งผ่านข้อมูลไปให้ถึงปลายทางได้

โปรโตคอล TCP (Transmission Control Protocol) เป็นโปรโตคอลที่มีการรับส่งข้อมูลแบบ stream oriented protocol หมายความว่า การรับส่งข้อมูลจะไม่คำนึงถึงปริมาณข้อมูลที่จะส่งไป แต่จะแบ่งข้อมูลเป็นส่วนย่อยๆก่อน แล้วจึงจะส่งไปยังปลายทางอย่างต่อเนื่องเป็นลำดับข้อมูล ในกรณีที่ข้อมูลส่วนใดส่วนหนึ่งสูญหายไป ก็จะส่งข้อมูลส่วนนั้นใหม่อีกครั้ง สำหรับปลายทางก็จะทำหน้าที่จัดเรียงส่วนของข้อมูล Datagram ใหม่ให้ต่อเนื่องกันและประกอบกลับเป็นข้อมูลทั้งหมดได้ ซึ่งจะแยกข้อมูลที่ไม่ถูกต้องออก ดังนั้นแอปพลิเคชันหรือโปรเซสใดที่อาศัยการส่งผ่านข้อมูลด้วยโปรโตคอลTCP จะต้องใช้หน่วยความจำและขนาดของช่องสัญญาณ (Bandwidth) เป็นปริมาณมากกว่าโปรโตคอลอื่นในเลเยอร์เดียวกัน

การติดต่อระหว่างกันจะต้องเป็นแบบ Connection-oriented คือ ต้องมีการสร้างการติดต่อกันเป็น Session ทั้ง 2 ด้านเสียก่อน แล้วจึงจะรับส่งข้อมูลไปได้พร้อมกัน (Full duplex) เหมือนกับการใช้โทรศัพท์ติดต่อกัน ซึ่งในระหว่างการติดต่อกันนั้น แม้ว่าฝ่ายใดฝ่ายหนึ่งหรือทั้งสองฝ่ายจะเงียบไป คือไม่พูดอะไรเป็นเวลานาน ๆ แต่การเชื่อมโยงระหว่างทั้งสองด้านยังคงมีอยู่ไม่ขาดไปจนกว่าฝ่ายใดฝ่ายหนึ่งจะวางสาย เช่นเดียวกันกับการติดต่อกันด้วยกลไกโปรโตคอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TCP เมื่อแอปพลิเคชันต้องการส่งผ่านข้อมูลจะใช้โปรโตคอลที่เหมาะสมในชั้น Process layer ติดต่อไปและมีการสร้างช่องส่งข้อมูลผ่าน port ที่กำหนดเพื่อส่งผ่านข้อมูลไปยังโปรโตคอล TCP

ในระหว่างการรับส่งข้อมูลนี้ โปรโตคอล TCP จะเพิ่มขบวนการสอบทานข้อมูล เพื่อให้ข้อมูลมีความถูกต้องไม่ผิดพลาดไปจากเดิม โดยการส่งสัญญาณสอบทานข้อมูล (acknowledgement) และส่งข้อมูลให้ใหม่อีกครั้ง ถ้าปลายทางไม่ได้รับหรือเกิดความผิดพลาดขึ้น

2.3 ไมโครคอนโทรลเลอร์ PIC16F877

2.3.1 คุณลักษณะพื้นฐานของ PIC16F877

1) มีคำสั่งให้ใช้งาน 35 คำสั่ง
2) คำสั่งหนึ่งๆ ใช้เวลาทำงาน 1 ถึง 2 Cycle
3) ทำงานได้สูงสุดที่สัญญาณนาฬิกาตั้งแต่ไฟตรงถึง 20 MHz
4) ทำงานแบบ Pipe-line ทำให้สามารถทำงานพร้อมกันในเวลาเดียวกันได้
5) หน่วยความจำโปรแกรมเป็นแบบ Flash มีขนาด 8KWord
6) มีหน่วยความจำข้อมูล (Data Memory RAM) ขนาด 368 ไบต์
7) มีหน่วยความจำแบบ EEPROM ขนาด 256 ไบต์
8) ตอบสนองกับอินเตอร์รัพต์ได้ทั้งหมด 14 แหล่ง
9) มี Stack ให้ใช้ได้สูงสุด 8 ระดับ
10) มีระบบ Power On Reset , Power Up Timer , Oscillator Start-up timer
11) มี Watchdog timer
12) มีระบบ Code Protection ป้องกันการคัดลอกเลียนแบบ
13) มีโหมดประหยัดพลังงาน (Sleep Mode)
14) สัญญาณนาฬิกามีหลายโหมดให้เลือกใช้งาน เช่น XTAL และ วงจร RC
15) สามารถโปรแกรมด้วยไฟ +5VDC ได้
16) ใช้การโปรแกรมแบบ In-Circuit Serial Programming
17) ทำงานที่ไฟเลี้ยง 2VDC ถึง 5.5VDC
18) Current Sink และ Current Source อยู่ที่ 25mA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

19) มีTimer/Counter 3 ตัว

20) มีโมดูล Capture/Compare/PWM อีก 2 ชุด

21) มี A/D Converter แบบ 10 บิต

22) มีระบบ USART สำหรับต่อกับการสื่อสารแบบ RS232

23) มีระบบตรวจระดับไฟเลี้ยง (Brown-out reset)

24) มี I/O พอร์ตทั้งหมด 5 พอร์ต

2.3.2 โครงสร้างขาสัญญาณของไมโครคอนโทรลเลอร์ PIC16F877

ไมโครคอนโทรลเลอร์ตระกูลPICเบอร์ PIC16F877 เป็นไมโครคอนโทรลเลอร์ขนาด 40 ขา มีขาสัญญาณต่างๆดังนี้

1.MCLR/Vpp : Master Clear(Reset) Input/Programming Voltage Input ทำหน้าที่เป็นสัญญาณรีเซ็ต(Reset)เมื่อขานี้ได้รับลอจิก 0 ไมโครคอนโทรลเลอร์จะถูกรีเซต และทำหน้าที่เป็นขาสัญญาณรับแรงดัน ขณะทำการบันทึกโปรแกรมลงหน่วยความจำของไมโครคอนโทรลเลอร์

2.VDD : Positive Supply (+2.00 V ถึง +5.5V) ทำหน้าที่เป็นขาไฟเลี้ยง

3.VSS : Ground ทำหน้าที่เป็นขากราวน์

4.OSC1/CLKIN : Oscillator Crystal Input/External Clock Source Input

5.OSC2/CLKOUT : Oscillator Crystal Output/External Clock Source Outputm

ทั้งสองขาทำหน้าที่เป็นขาสัญญาณสำหรับต่อคริสตัล ในกรณีที่อยู่ในโหมดการใช้สัญญาณนาฬิกาจากภายนอก(Crystal Oscillator Mode)

6.RA0 – RA5 : พอร์ต A มีจำนวน 6 ขา เป็นพอร์ตแบบสองทิศทาง(Bi-directional I/O Port)คือ เป็นได้ทั้งพอร์ตอินพุตและพอร์ตเอาต์พุตใช้ในการรับและส่งข้อมูล นอกจากนี้ยังทำหน้าที่อื่นๆแสดงดังตารางที่ 2.1

ตารางที่ 2.1 หน้าที่ของขาสัญญาณของพอร์ต A

พอร์ต	สัญญาณ	หน้าที่
RA0	AN0	รับสัญญาณอินพุตสำหรับ ADC ช่อง 0
RA1	AN1	รับสัญญาณอินพุตสำหรับ ADC ช่อง 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 หน้าที่ของขาสัญญาณของพอร์ต A (ต่อ)

พอร์ต	สัญญาณ	หน้าที่
RA2	AN2	รับสัญญาณอินพุตสำหรับ ADC ช่อง 2
RA3	AN3	รับสัญญาณอินพุตสำหรับ ADC ช่อง 3
RA4	TOCK1	รับสัญญาณ Input Clock ของ Timer 0
RA5	AN4	รับสัญญาณอินพุตสำหรับ ADC ช่อง 4
	SS	รับสัญญาณ Slave Select จากการติดต่อของ Serial Port แบบSynchronize

7.RB0-RB7 : พอร์ต B มีจำนวน 8 ขา ขนาด 8 บิต เป็นพอร์ตแบบสองทิศทาง ใช้ในการส่งและรับข้อมูล นอกจากนี้บางขายังทำหน้าที่รับสัญญาณอินพุตจากการอินเตอร์รัปต์ (Interrupt) จากภายนอกด้วย แสดงดังตารางที่ 2.2

ตารางที่ 2.2 หน้าที่ของขาสัญญาณของพอร์ต B

พอร์ต	สัญญาณ	หน้าที่
RB0	INT	รับสัญญาณอินพุตจากการอินเตอร์รัพต์จากภายนอก
RB3	PGM	รับสัญญาณอินพุตแรงดันต่ำในการบันทึกโปรแกรม(ถ้ามีการ Enable)
RB6	PGC	ขาสัญญาณนาฬิกาในการบันทึกโปรแกรม
RB7	PGD	ขาสัญญาณข้อมูลในการบันทึกโปรแกรม

8.RC0 – RC7 : พอร์ต C มีจำนวน 8 ขา ขนาด 8 บิต เป็นพอร์ตแบบสองทิศทางใช้ในการส่งและรับข้อมูล นอกจากนี้ยังทำหน้าที่อื่นๆ แสดงดังตารางที่ 2.3

ตารางที่ 2.3 หน้าที่ของขาสัญญาณของพอร์ต C

พอร์ต	สัญญาณ	หน้าที่
RC0	T1OSO	ขาสัญญาณเอาต์พุตของวงจรออสซิลเลเตอร์ของ Timer 1
	TICK1	ขาสัญญาณอินพุตของสัญญาณนาฬิกาของ Timer 1
RC1	T1OSI	ขาสัญญาณอินพุตของวงจรออสซิลเลเตอร์ของ Timer 1
	CCP2	ขาสัญญาณเอาต์พุตของโมดูล CCP 2 (Cature2,Compare2,PWM2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 หน้าที่ของขาสัญญาณของพอร์ต C (ต่อ)

พอร์ต	สัญญาณ	หน้าที่
RC2	CCP1	ขาสัญญาณเอาต์พุตของโมดูล CCP 1 (Cature1,Compare1,PWM1)
RC3	SCK	ขาสัญญาณนาฬิกาของวงจร SPI
	SCL	ขาสัญญาณนาฬิกาของวงจร I2C
RC4	SDI	ขาสัญญาณอินพุตและ Serial Data ของระบบ SPI
	SDA	ขาข้อมูลของระบบบัส I2C
RC5	SDO	ขาสัญญาณเอาต์พุตและ Serial Data ของระบบ SPI
RC6	TxD	ขาส่งข้อมูลแบบ Serial Port
	CK	ขาสัญญาณนาฬิกา แบบ Synchronize
RC7	RxD	ขารับข้อมูลแบบ Serial Port
	DT	ขาข้อมูลแบบ Synchronize

9.RD0 – RD7 : พอร์ต A มีจำนวน 8 ขา ขนาด 8 บิต เป็นพอร์ตแบบสองทิศทาง ใช้ในการส่งและรับข้อมูล นอกจากนี้ยังมีหน้าที่อื่นๆ แสดงดังตารางที่ 2.4

ตารางที่ 2.4 หน้าที่ของขาสัญญาณของพอร์ต D

พอร์ต	สัญญาณ	หน้าที่
RD0	PSP0	ขาสัญญาณขยายพอร์ตแบบขนาน บิต 0
RD1	PSP1	ขาสัญญาณขยายพอร์ตแบบขนาน บิต 1
RD2	PSP2	ขาสัญญาณขยายพอร์ตแบบขนาน บิต 2
RD3	PSP3	ขาสัญญาณขยายพอร์ตแบบขนาน บิต 3
RD4	PSP4	ขาสัญญาณขยายพอร์ตแบบขนาน บิต 4
RD5	PSP5	ขาสัญญาณขยายพอร์ตแบบขนาน บิต 5
RD6	PSP6	ขาสัญญาณขยายพอร์ตแบบขนาน บิต 6
RD7	PSP7	ขาสัญญาณขยายพอร์ตแบบขนาน บิต 7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10.RE0 –RE2 : พอร์ต E มีจำนวน 3 ขา เป็นพอร์ตแบบสองทิศทาง ใช้ในการรับส่งข้อมูล นอกจากนี้ยังทำหน้าที่อื่นๆ แสดงดังตารางที่ 2.5

ตารางที่ 2.5 หน้าที่ของขาสัญญาณของพอร์ต E

พอร์ต	สัญญาณ	หน้าที่
RE0	AN5	รับสัญญาณอินพุตสำหรับ ADC ช่อง 5
	RD	ขาสัญญาณขยายพอร์ตแบบขนานควบคุมการอ่าน
RE1	AN6	รับสัญญาณอินพุตสำหรับ ADC ช่อง 6
	WR	ขาสัญญาณขยายพอร์ตแบบขนานควบคุมการเขียน
RE2	AN7	รับสัญญาณอินพุตสำหรับ ADC ช่อง 6
	CS	ขาสัญญาณขยายพอร์ตแบบขนานควบคุมการเลือกอุปกรณ์

รูปที่ 2.2 การจัดขามาตรฐานของไมโครคอนโทรลเลอร์ PIC16F877 [2]

2.2.3 หน่วยความจำ(Memory Unit)

ในการเขียนโปรแกรมด้วยภาษาซีให้กับไมโครคอนโทรลเลอร์นั้นต้องคำนึงถึงชนิดของหน่วยจำและวิธีการเข้าถึงด้วย ซึ่งต่างจากการเขียนบนคอมพิวเตอร์ที่สนใจเพียงชนิดของตัวแปรว่าจะใช้เก็บข้อมูลประเภทใดสำหรับหน่วยความจำในระบบไมโครคอนโทรลเลอร์ PIC นั้นจะมีหน่วยความจำในการใช้งาน 3 ประเภท ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3.1 หน่วยความจำโปรแกรมแบบแฟลช (Flash program memory)

หน่วยความจำแบบแฟลช (Flash ROM) ในปัจจุบันนี้หน่วยความจำชนิดนี้ได้ถูกนำมาใช้กับไมโครคอนโทรลเลอร์หลายบริษัทหลายรุ่น โดยมีคุณสมบัติในการเขียนโปรแกรมและลบโปรแกรมได้มากกว่า 100000 ครั้ง ซึ่งการทำงานจะมีความเร็วสูงมากเหมาะกับการพัฒนางานที่มีขนาดใหญ่

2.2.3.2 หน่วยความจำโปรแกรม (Data memory RAM)

หน่วยความจำส่วนนี้มีไว้ใช้เก็บข้อมูลขณะประมวลผลโปรแกรม สามารถอ่านและเขียนข้อมูลได้ขณะมีไฟเลี้ยง แต่เมื่อไม่จ่ายไฟเลี้ยงข้อมูลต่างๆจะสลายไป หากหน่วยความจำส่วนนี้ไม่พอใช้งานจะต้องต่อหน่วยความจำแรมภายนอกเพิ่ม (External RAM หรือ Data Memory) ปัจจุบันเทคโนโลยีก้าวหน้าขึ้นมากชิพบางตัวจะมีการบรรจุหน่วยความจำประเภท Data Memory เข้าไปในชิพเลย

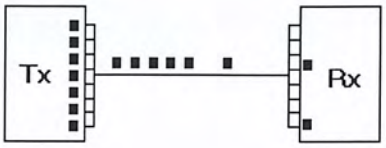
2.2.3.3 หน่วยความจำแบบอีอีพรอม (EEPROM data memory)

หน่วยความจำแบบ EEPROM เป็นหน่วยความจำที่สามารถเขียนและลบโปรแกรมด้วยกระแสไฟฟ้าในหน่วยความจำถาวรของ PROM (Programmable read only memory) โดยภายในมีการพัฒนาให้ RAM (Random Access Memory) ที่มีหน่วยความจำชั่วคราวให้เก็บข้อมูลได้ถาวรแบบหน่วยความจำ ROM (Read Only Memory) โดยสามารถเขียนและลบโปรแกรมจำนวนหลายๆครั้งได้

2.4 การเชื่อมต่อสื่อสารแบบอนุกรม

การเชื่อมต่อสื่อสารแบบอนุกรมเป็นอีกหนึ่งวิธีที่ใช้ในการส่งผ่านข้อมูลจากจุดส่งไปยังจุดรับ ผ่านทางพอร์ตอนุกรม (Serial port) ซึ่งความเร็วในการเชื่อมต่อเพื่อถ่ายโอนข้อมูลของการเชื่อมต่อสื่อสารแบบอนุกรมนี้อยู่ในระดับช้าเมื่อเทียบกับการเชื่อมต่อในรูปแบบอื่น เพราะในการถ่ายโอนนั้น ข้อมูลจะมีการส่งออกมาเพียงครั้งละ 1 บิตระหว่างจุดรับและจุดส่ง โดยมีตัวกลางในการสื่อสารเพียงช่องเดียว ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 การถ่ายโอนข้อมูลโดยใช้การเชื่อมต่อแบบอนุกรม

จะเห็นว่าการส่งข้อมูลแบบอนุกรม ข้อมูลจะถูกเปลี่ยนให้เป็นแบบอนุกรมเสียก่อน แล้วจึงส่งออกเพียงครั้งละ 1 บิต ไปยังฝั่งรับ โดยที่จุดรับจะต้องมีกลไกในการเปลี่ยนข้อมูลที่ส่งมาครั้งละบิตให้เป็นสัญญาณแบบขนาน เพื่อให้ข้อมูลที่ได้ครบถ้วนสมบูรณ์ ซึ่งกลไกดังกล่าวนั้นสามารถเลือกใช้ได้ 2 ประเภท คือ แบบซิงโครนัส (Synchronous) และแบบอะซิงโครนัส (Asynchronous)

1. การสื่อสารแบบอะซิงโครนัส

การสื่อสารแบบนี้ประกอบด้วยบิตเริ่มต้น(Start bit) และบิตสิ้นสุด (Stop bit)

รูปที่ 2.4 การสื่อสารแบบอะซิงโครนัสในระดับบิต

ขณะที่สถานะของการส่งเป็นแบบว่าง (Idle) คือยังไม่มีสัญญาณที่ส่งออกมาแต่จะมีสัญญาณ หรือมีแรงดันตลอดเวลาเพื่อความแน่ใจว่าฝ่ายรับยังติดต่ออยู่กับฝ่ายส่ง ฝ่ายส่งจะเริ่มส่งข้อมูลบอกจุดเริ่มต้นสัญญาณของอะซิงโครนัสจะเป็น "0" หนึ่งช่วงสัญญาณนาฬิกา บิตนี้เรียกว่าบิตเริ่มต้น ข้อมูล 1 ตัวอักษรที่ตามหลังบิตเริ่มต้นจะมีขนาดตั้งแต่ 5 บิต จนถึง 8 บิต เป็นรหัสแอสกี (ASCII code) โดยใช้ 5 บิตในการแทนอักขระ 1 ตัวและมีส่วนที่ตามหลังข้อมูลเป็นบิตตรวจสอบความผิดพลาด (Parity bit) ซึ่งอาจจะใช่หรือไม่ก็ได้ โดยที่บิตนี้จะทำหน้าที่เป็นตัวตรวจสอบความถูกต้องของสัญญาณที่ได้รับ

2.การสื่อสารแบบซิงโครนัส

การสื่อสารแบบซิงโครนัส มีความแตกต่างกับการสื่อสารแบบอะซิงโครนัสในเรื่องความต่อเนื่องของข้อมูลที่ส่ง เพราะข้อมูลที่ส่งจากฝั่งส่งเป็นแบบต่อเนื่อง ไม่มีบิตเริ่มต้นหรือบิตสิ้นสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรือกระทั่งบิตตรวจสอบความผิดพลาด รูปแบบที่ใช้ในการส่งข้อมูลแบบซิงโครนัส จึงแตกต่างไปจากการส่งข้อมูล แบบอะซิงโครนัสโดยสิ้นเชิง

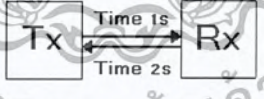
การสื่อสารแบบซิงโครนัสจะทำในระดับอักขระซึ่งหมายความว่าอักขระแต่ละตัวมีขอบเขตที่แน่นอนโดยเริ่มกระทำจากจุดเริ่มต้นของการส่งข้อมูล สถานีส่งจะส่งสัญญาณที่เรียกว่าตัวอักษรนำ (Leading pad character) ไปยังฝั่งรับก่อนที่จะเริ่มส่งข้อมูล ตัวอักษรนำจะประกอบด้วย "0" และ "1" สลับกัน เพื่อให้สถานีรับจัดสัญญาณนาฬิกาให้ตรงกันก่อน หลังจากนั้นจะมีการส่งอักขระซิงค์ (Sync character) เพื่อแจ้งฝั่งรับอีกว่า ข้อมูลมีความยาวและเริ่มต้นหรือสิ้นสุดที่ใด

วิธีการส่งผ่านข้อมูลทั้งสองประเภทของการเชื่อมต่อสื่อสารแบบอนุกรมนั้นจะถูกกำหนดรูปแบบข้อมูลในการส่ง ซึ่งข้อมูลในการส่งนั้นจะแบ่งได้เป็น 3 รูปแบบ คือ

1. แบบทางเดียว (Simplex) เป็นการที่ฝั่งส่งจะส่งข้อมูลไปเพียงช่องทางเดียวเท่านั้น โดยที่ฝั่งรับไม่สามารถส่งกลับข้อมูลไม่ว่าจะเป็นช่วงเวลาเดียวกันหรือต่างเวลา

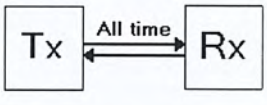
รูปที่ 2.5 การส่งข้อมูลแบบทางเดียว

2.แบบกึ่งทางคู่ (Half-duplex) เป็นการที่ฝั่งส่งจะส่งข้อมูลไปยังฝั่งรับ โดยที่ฝั่งรับไม่สามารถส่งกลับข้อมูลหรือเปลี่ยนเป็นผู้ส่งในช่วงเวลาเดียวกันได้



รูปที่ 2.6 การส่งข้อมูลแบบกึ่งทางคู่

3.แบบทางคู่ (Full-duplex) เป็นการที่ฝั่งส่งและฝั่งรับสามารถติดต่อและส่งผ่านข้อมูลในเวลาเดียวกันได้ หรือสามารถเปลี่ยนแปลงตัวเองเป็นฝั่งรับหรือฝั่งส่งได้ตลอดเวลา



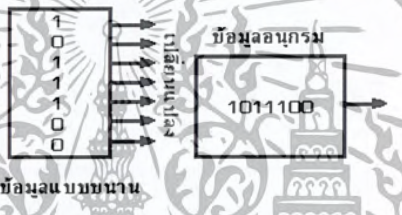
รูปที่ 2.7 การส่งข้อมูลแบบทางคู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1 ความเร็วในการส่งถ่ายโอนข้อมูลแบบอนุกรม

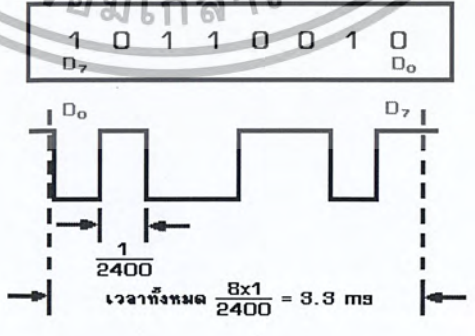
ความเร็วของการถ่ายโอนข้อมูลแบบอนุกรมมีหน่วยวัดเป็น บิตต่อวินาที หรือที่เรียกว่า บีพีเอส (bps) แต่เรายังมีหน่วยที่นิยมใช้กันมากคือ บอดเรต หรือ อัตราบอด (baud rate) ซึ่งหมายถึง การเปลี่ยนแปลงของสัญญาณใน 1 วินาที หลายคนยังเข้าใจสับสนระหว่างหน่วยบีพีเอส กับอัตราบอด กล่าวคือ การเปลี่ยนแปลงของสัญญาณของสัญญาณ 1ครั้งอาจจะแสดงถึง การส่งข้อมูลแบบอนุกรมมากกว่า 1 บิต อัตราการส่งข้อมูลเป็นจำนวนบิตจึงเท่ากับ อัตราบอดคูณกับจำนวนบิตใน 1 บอด

2.4.2 ระยะเวลาและอัตราการส่งข้อมูล



รูปที่ 2.8 การส่งข้อมูลที่มีขนาด 7 บิตจากไมโครโปรเซสเซอร์ส่งออกที่ช่องสื่อสารแบบอนุกรม

ในการส่งข้อมูลแบบอนุกรมจะต้องพิจารณา ความเร็วของข้อมูลในการส่งเรียกว่า bit rate และ อัตราการเปลี่ยนแปลงของสัญญาณ 1 ครั้งต่อข้อมูล 1 บิต เรียก baud rate โดย baud rate ที่ใช้ในการส่งข้อมูลทั่วไปคือ 110, 150, 300, 1200, 2400, 4800 และ 9600 ซึ่งยกตัวอย่างได้ว่า หากต้องการส่งข้อมูลด้วย baud rate 2400 baud ข้อมูลจะได้รับการส่งออกไปดังรูป



รูปที่ 2.9 แสดงตัวอย่าง bit rate และ baud rate ของสัญญาณดิจิตอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงให้เห็นช่วงเวลาของการส่ง ซึ่งในช่วงเวลาของแต่ละบิตจะมีขนาดช่วงเวลาเท่ากับ 1/2400วินาทีซึ่งเท่ากับ 0.000416 วินาที หรือ 416 ไมโครวินาที ดังนั้นถ้าต้องการส่งข้อมูลที่มีขนาด 8 บิต ก็จะใช้เวลาทั้งสิ้น 8×416 ไมโครวินาที หรือ เท่ากับ 3,2328 ไมโครวินาที และเมื่อเทียบกับการส่งข้อมูลแบบขนาน จะพบว่าใช้เวลาน้อยกว่า 1 ไมโครวินาที

2.5 มาตรฐาน RS-232

RS-232 เป็นมาตรฐานการเชื่อมต่อข้อมูลแบบอนุกรมที่มีคนนิยมใช้มากที่สุด กำหนดโดย EIA (Electronics Industry Association) หรือสมาคมผู้ประกอบการอุตสาหกรรมอิเล็กทรอนิกส์ ของสหรัฐอเมริกา ตั้งแต่ปี 1969 โดยมีจุดเริ่มต้นจากความต้องการที่จะกำหนดมาตรฐานการเชื่อมต่อระหว่าง คอมพิวเตอร์กับโมเด็มในสมัยนั้น แต่ในปัจจุบันนั้นมาตรฐาน RS-232 ได้มีการปรับปรุงแก้ไขจากมาตรฐานเดิมเป็น RS-232-C อันที่จริงแล้วยังมีมาตรฐาน RS-232-D ที่มีการเพิ่มข้อกำหนดของคอนเน็คเตอร์แบบ DB เข้าไปด้วย เช่น DB-9 และ DB-25 จึงสามารถรวมข้อกำหนดเข้าไว้ด้วยกันได้

ลักษณะโดยทั่วไปของการเชื่อมต่อข้อมูลแบบอนุกรมตามมาตรฐาน RS-232 คือเป็นการสื่อสารข้อมูลแบบจุดต่อจุด ซึ่งเดิมทีเป็นการสื่อสารข้อมูลระหว่างคอมพิวเตอร์กับโมเด็ม มีรูปแบบข้อมูลในการสื่อสารเป็นแบบสองทางพร้อมกัน (Full-duplex) โดยอาจมีการใช้สายสัญญาณอื่นร่วมเพื่อทำแฮนด์เชค (Hand-shake) มาตรฐาน RS-232 จำกัดความยาวสายไว้ที่ 15 เมตร สำหรับการส่งสัญญาณที่ความเร็ว 19,200 บิตต่อวินาที โดยที่ความยาวสายจะต้องสั้นลงถ้าต้องการสื่อสารที่ความเร็วสูงขึ้น ตัวมาตรฐานจะกำหนดสิ่งที่เกี่ยวข้องกับการเชื่อมต่อนี้ด้วยกันทั้งหมด 4 หัวข้อหลักๆ ด้วยกันคือ

1. คุณสมบัติทางไฟฟ้าของสัญญาณ
2. คุณสมบัติทางกลของการเชื่อมต่อ ซึ่งหมายถึงตัวคอนเน็คเตอร์นั่นเอง
3. หน้าที่การทำงานของวงจรสำหรับแลกเปลี่ยนข้อมูล
4. มาตรฐานการเชื่อมต่อสำหรับระบบสื่อสารเฉพาะอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.1 คอนเน็คเตอร์สำหรับมาตรฐาน RS-232

คอนเน็คเตอร์สำหรับมาตรฐาน RS-232 มี 2 ประเภท คือ แบบ 25 ขา เรียกว่า DB-25 และ แบบ 9 ขา เรียกว่า DB-9 โดยทั้ง 2 ประเภทนี้จะประกอบไปด้วย 2 ลักษณะ คือ แบบตัวผู้และตัวเมีย ตามแต่การเลือกใช้ในการติดตั้ง ดังรูป

รูปที่ 2.10 คอนเน็คเตอร์ DB-25 แบบตัวผู้และตัวเมีย

รูปที่ 2.11 คอนเน็คเตอร์ DB-9 แบบตัวผู้และตัวเมีย

โดยที่แต่ละขามีลักษณะตามตำแหน่งดังนี้

ตารางที่ 2.6 การจัดตำแหน่งขาและหน้าที่ของคอนเน็คเตอร์มาตรฐาน RS-232

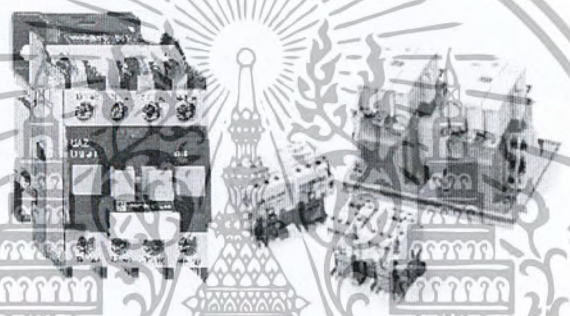
ตำแหน่งขา DB-9	ตำแหน่งขา DB-25	หน้าที่การเชื่อมต่อ	ชนิดข้อมูลที่ส่งผ่าน
1	8	data carrier detect : DCD	input
2	3	received data : RxD	input
3	2	transmitted data : TxD	output
4	20	data terminal teady : DTR	output
5	7	signal ground : GND	-
6	6	data set ready : DSR	input
7	4	request to send : RTS	output
8	5	clear to send : CTS	input
9	22	ring indicator : RI	input

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 รีเลย์

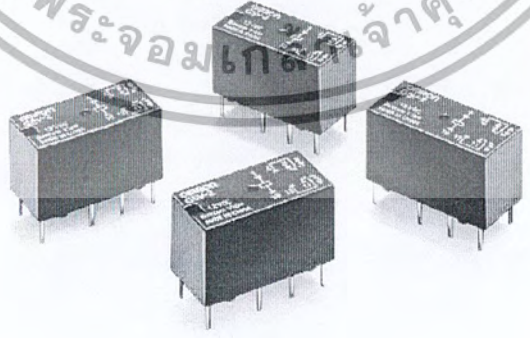
รีเลย์ (Relay) คือ อุปกรณ์อิเล็กทรอนิกส์ที่ทำหน้าที่เป็นสวิตช์มีหลักการทำงานคล้ายกับ ขดลวดแม่เหล็กไฟฟ้า (solenoid) รีเลย์ใช้ในการควบคุมวงจรไฟฟ้าได้อย่างหลากหลาย และยังเป็นอุปกรณ์แม่เหล็ก (Magnetics device) ที่นิยมใช้อย่างแพร่หลายโดยไม่มีการแก้ไขสภาพโครงสร้างมากมายนัก รีเลย์สามารถแบ่งออกตามลักษณะการใช้งานได้เป็น 2 ประเภทคือ

1. รีเลย์กำลัง (Power relay) ใช้ในการควบคุมไฟฟ้ากำลัง มีขนาดใหญ่กว่ารีเลย์ธรรมดา นิยมเรียกกันว่า คอนแท็คเตอร์ (Contactor)



รูปที่ 2.12 รีเลย์กำลัง

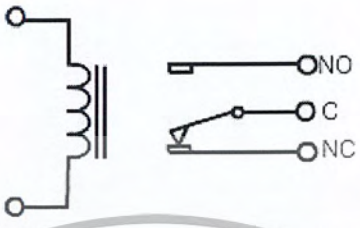
2. รีเลย์ควบคุม (Control relay) มีขนาดเล็กกำลังไฟฟ้าต่ำ ใช้ในวงจรควบคุมทั่วไปที่มีกำลังไฟฟ้าไม่มากนัก หรือเพื่อการควบคุมรีเลย์หรือคอนแทกเตอร์ขนาดใหญ่



รูปที่ 2.13 รีเลย์ควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.1 คุณสมบัติและโครงสร้างของรีเลย์



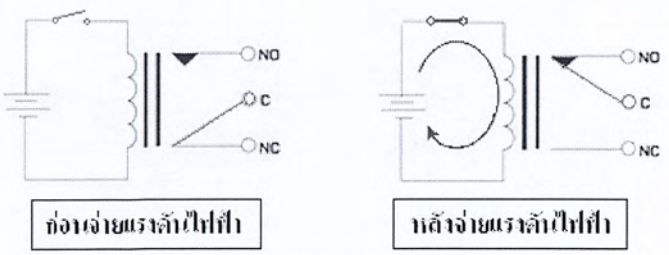
รูปที่ 2.14 โครงสร้างทั่วไปของรีเลย์

ภายในโครงสร้างของรีเลย์ จะประกอบไปด้วยขดลวด (Coil) 1 ชุด และ หน้าสัมผัส (Contactor) ซึ่งในหน้าสัมผัส 1 ชุด จะประกอบไปด้วย

- ขา C (Common) เป็นขาต่อระหว่าง NO และ NC เมื่อมีการทำงานของรีเลย์ โดยที่จะมีแรงดันที่ต้องการใช้งานไหลผ่านขานี้เสมอ
- ขา NO (Normally opened) หน้าสัมผัสแบบปกติเปิด โดยปกติขานี้จะเปิดเอาไว้ และจะทำงานเมื่อมีการจ่ายแรงดันไฟฟ้าให้รีเลย์
- ขา NC (Normally closed) หน้าสัมผัสแบบปกติปิด โดยปกติขานี้จะต่อกับขา C ในกรณีที่ไม่ได้จ่ายแรงดันไฟฟ้าหน้าสัมผัสของ C และ NC จะต่อถึงกัน

2.6.2 การทำงานของรีเลย์

รีเลย์โดยทั่วไปจะทำงานก็ต่อเมื่อมีกระแสไฟฟ้าไหลผ่านขดลวด จะทำให้ขดลวดเกิดสนามแม่เหล็กไปดึง แผ่นหน้าสัมผัส C ดึงลงมา แตะหน้าสัมผัส NO ทำให้มีกระแสไหลผ่านหน้าสัมผัสไปได้



รูปที่ 2.15 การทำงานของรีเลย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

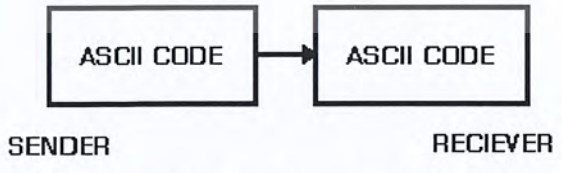
2.7 หลักการรับส่งข้อความสั้น

การส่งข้อความสั้น (SMS : Short Message Service) คือ การส่งข้อความสั้นๆหรือข้อมูลสั้นจากเครื่องโทรศัพท์มือถือผู้ส่ง ไปยังเครื่องโทรศัพท์มือถือของผู้รับโดยส่งผ่านเครือข่ายศูนย์บริการ (Short Message Service Center : SMSC) โดยการส่งแบบข้อความสั้นนี้เราจะสามารถเลือกได้ว่าจะส่งข้อความสั้น หรือ รูปภาพโลโก้ เสียงเพลงริงโทน ซึ่งจะมีวิธีการส่งที่แตกต่างกัน 2 แบบ คือ โหมดตัวอักษร (Text-Mode) และโหมดพีดียู(Protocol Data Unit : PDU) โดยโหมดตัวอักษร คือ โหมดที่เราสามารถส่งข้อความสั้นๆประมาณ 160 ตัวอักษรไปยังเครื่องโทรศัพท์มือถือของผู้รับโดยลักษณะข้อความนั้นจะอยู่ในรูปแบบรหัสแอสกี้ ส่วนโหมดพีดียู คือ โหมดที่สามารถส่งได้ทั้งข้อความสั้นๆส่งรูปภาพและเพลงริงโทนได้ ซึ่งโหมดพีดียูนี้จะมีรูปแบบการวางข้อมูลที่จะส่งแตกต่างกับโหมดตัวอักษร คือ โหมดพีดียูจะมีการเข้ารหัสที่จะแปลงข้อความเป็นรูปแบบของเลขฐานสิบหกและต้องมีการส่งหัวข้อของชุดข้อมูล (Heading) แต่ในโหมดตัวอักษรจะเป็นการส่งแบบรหัสแอสกี้และไม่จำเป็นต้องส่งหัวข้อของชุดข้อมูล

2.7.1 โหมดการรับส่งข้อมูล ข้อความสั้น

2.7.1.1 โหมดข้อความ (Text Mode)

จัดเป็นการส่งข้อมูลในรูปของตัวอักษรโดยตรง ซึ่งตัวเครื่องนั้นจะเก็บข้อความเป็นรหัสแอสกี แล้วส่งให้กับเลขหมายปลายทาง และเลขหมายปลายทางจะแปลงรหัสแอสกีที่ได้มาให้เป็นตัวอักษรดังแสดงในรูปที่ 2.16 เพื่อแสดงผลออกทางหน้าจอ ซึ่งตัวเครื่องโทรศัพท์มือถือส่วนใหญ่จะไม่รองรับการส่งข้อมูลในรูปแบบนี้ผ่านทาง AT Command

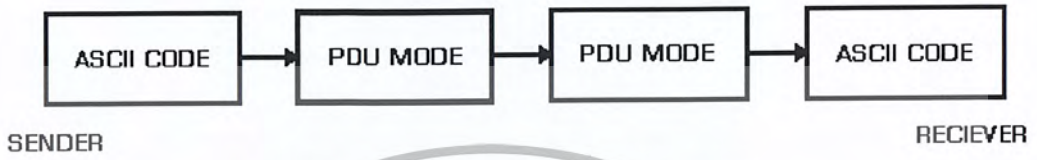


รูปที่ 2.16 การรับส่งข้อความแบบโหมดข้อความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
117519

2.7.1.2 โหมดพีดียู (Protocol Description Unit Mode)

จัดเป็นการส่งข้อมูลในรูปแบบที่ต้องมีการนำเอาข้อมูลที่เป็นแอสกีมาแปลงให้เป็นรหัสพีดียู และนำรหัสพีดียูส่งให้กับหมายเลขปลายทาง จากนั้นเมื่อหมายเลขปลายทางได้รับข้อมูลแล้วจะแปลงข้อมูลจากรหัสพีดียูให้เป็นแอสกีต่อไปดังแสดงในรูปที่ 2.17



รูปที่ 2.17 การรับส่งข้อความแบบโหมดพีดียู

2.7.2 โหมดพีดียู

โหมดพีดียู คือ รูปแบบของข้อมูลที่ใช้ในการส่งข้อความสั้นซึ่งเกิดจากการนำเอาข้อมูลดิจิตอลมาเข้ารหัสเพื่อใช้ในการส่งข้อมูลผ่านทาง Air Interface โดยการรับข้อมูลและส่งข้อมูลข้อความสั้น แบบโหมดพีดียู จะมีโครงสร้างที่แตกต่างกัน รหัสพีดียู คือ ข้อมูลข้อความสั้นที่อยู่ในรูปโหมดพีดียู ประกอบด้วยเลขฐาน 10 และเลขฐาน 16 โดยตัวเลขแต่ละคู่เรียกว่ากลุ่มตัวเลข 8 บิต

2.7.2.1 การรับข้อมูล ข้อความสั้น โหมดพีดียู

ข้อมูลที่ได้รับประกอบด้วยข้อมูลของผู้ส่ง ข้อมูล SMSC Time Stamp และอื่นๆ ตามด้วยส่วนของข้อความซึ่งจะอยู่ท้ายสุด ยกตัวอย่างเช่น 07 91 6698019021F0 04 0B 91 6668353407F2 00 00 80706161924382 09 C7F79130C62D7C76B โดยความหมายของรหัสนี้สรุปได้ดังตารางที่ 2.7

ตารางที่ 2.7 ความหมายของรหัสข้อความสั้นที่ได้รับ

กลุ่มตัวเลข 8 บิต	รายละเอียด
07	ความยาวของ SMSC Information 7 Octet
91	รูปแบบของเลขหมายแบบสากล
66 98 01 90 21 F0	เลขหมาย SMSC ซึ่งเป็นเลขฐาน 10 สลับ nibble ในกรณีนี้ หมายถึงเลขหมายจริง Server center คือ +66 89 10 09 12 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.7 ความหมายของรหัส ข้อความสั้นที่ได้รับ (ต่อ)

กลุ่มตัวเลข 8 บิต	รายละเอียด
04	First octet of these SMS-DELIVER message
0B	ความยาวของเลขหมายผู้ส่ง (0B hex = 11 ตัว)
91	รูปแบบของเลขหมายผู้ส่ง 91 หมายถึงเลขหมายแบบสากล
66 68 35 34 07 F2	เลขหมายผู้ส่งเป็นเลขฐาน 10 สลับ nibble ในกรณีนี้หมายถึงเลขหมายจริงของผู้ส่งคือ +66 86 53 43 70 2
00	TP – PID (Protocol identifier)
00	TP – DCS (Data coding scheme) 00 คือการเข้ารหัสข้อความแบบ 7 บิต Default Alphabet
80706161924382	TP – SCTS ข้อมูล Time stamp สลับ nibble
09	TP – UDL (User data length) จำนวนตัวอักษรของข้อความที่ส่งในที่นี้คือ 9 ตัว (Good luck)
C7F79130C62D7C76B	TP – UD (User data) ข้อความ “Good luck” ที่เข้ารหัสแบบตัวอักษร 7 บิต

2.7.2.2 การส่งข้อมูล ข้อความสั้น โหมดพีดียู

ข้อมูลที่ส่งจะประกอบด้วย ข้อมูลของ SMSC หมายถึงผู้รับ ช่วงเวลาหมดอายุของข้อความ และอื่นๆ ปิดท้ายด้วยข้อมูลในส่วนของข้อความสั้น ยกตัวอย่างเช่น00 11 00 0A 91 8031309280 00 00 AA 09 C7F79130C62D7C76B โดยมีความหมายดังแสดงในตาราง

ตารางที่ 2.8 ความหมายของรหัสในแต่ละกลุ่มตัวเลข 8 บิต

กลุ่มตัวเลข 8 บิต	รายละเอียด
00	ความยาวของ ข้อความสั้นC Information “00” หมายถึงใช้ ข้อความสั้นC Information ที่เก็บอยู่ภายในเครื่อง
11	First octet of the ข้อความสั้น-SUBMIT message

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.8 ความหมายของรหัสในแต่ละ กลุ่มตัวเลข 8 บิต (ต่อ)

กลุ่มตัวเลข 8 บิต	รายละเอียด
00	TP – Message Reference “00” คือ ให้เครื่องตั้งหมายเลขอ้างอิงข้อความนี้เอง
0A	Address Length ความยาวของหมายเลขผู้รับ(10 ตัว)
91	Type of Address (91 indicates international format of the phone number)
8013032908	หมายเลขโทรศัพท์ของผู้รับ (แบบ semi octets) หมายเลขที่แท้จริงคือ +0831309280
00	TP-PID (Protocol identifier) เป็น 00
00	TP-DCS (Data coding scheme เป็น 00
AA	TP-Validity Period “AA” หมายถึง ช่วงเวลาหมดอายุของข้อความ 4 วัน ถ้าภายในช่วงเวลานี้ยังส่งไม่ถึงปลายทางข้อความจะถูกยกเลิก
09	TP – UDL (User data length) จำนวนตัวอักษรของข้อความที่ส่งในที่นี้คือ 9 ตัว (Good luck)
C7F79130C62D7C76B	TP – UD (User data) ข้อความ “Good luck” ที่เข้ารหัสแบบตัวอักษร 7 บิต

2.7.3 การแปลงตัวอักษรชนิด 7 บิตเป็นข้อมูล 8 บิต (Octet)

จากการส่ง ข้อความสั้น ด้านบนในส่วนของชุดข้อความ จะเป็นส่วนที่เราสามารถใส่รหัสของข้อความที่ต้องการส่ง แต่เนื่องจากไม่สามารถนำรหัสของตัวอักษรแบบ 7 บิต ใส่ไปได้โดยตรงจำเป็นต้องผ่านการแปลงให้เป็นรหัสข้อมูลแบบ 8 บิตก่อนโดยตัวอย่างต่อไปนี้เป็นการแปลงข้อความ “hellohello” ยาว 10 ตัวอักษรซึ่งแต่ละตัวเป็นอักษรเป็นชนิด 7 บิตให้เป็นข้อมูล 8 บิต สำหรับใช้ในการส่ง ข้อความสั้น การแปลงเริ่มจากการนำรหัส 7 บิต ของตัวอักษรตัวแรก (h) มาเติมข้างหน้าด้วย 1 บิต ท้ายสุดของรหัส 7 บิตของอักษรตัวที่ 2 (e) จะได้ผลลัพธ์ 8 บิต (1 ไบต์) เป็น “E8” ขั้นตอนต่อมาให้เอา 6 บิตที่เหลือของอักษรตัวที่ 2 มาเติมข้างหน้าด้วย 2 บิตท้ายของรหัส 7 บิตของอักษรตัวที่ 3 (I) จะได้ผลลัพธ์ 8 บิต เป็น “32” และทำเช่นนี้เรื่อยไปโดยจำนวนบิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่นำมากระทำจะเพิ่มขึ้นเป็น 3 บิต 4 บิต จนกระทั่งถึง 7 บิต แล้วเริ่มกระบวนการใหม่จนกระทั่งหมดชุดตัวอักษร หลังจากการแปลงข้อความ “hellohello” จะได้ข้อมูลเป็นเลขฐาน 16 จำนวน 9 ไบต์ เป็น E8 32 9B FD 46 97 D9 EC 37 โดยที่ตัวอักษรชนิด 7 บิตถูกกำหนดโดยมาตรฐาน GSM 03.38 ดังตารางที่ 2.9

ตารางที่ 2.9 ชุดของตัวแปรมาตรฐาน GSM 03.38

Dec		0	16	32	48	64	80	96	112
	Hex	0	10	20	30	40	50	60	70
0	0	@	Δ	SP	0	¡	P		p
1	1	£	_	!	1	A	Q	A	q
2	2	$	Φ	"	2	B	R	B	r
3	3	¥	Γ	#	3	C	S	c	s
4	4	è	Λ	¤	4	D	T	d	t
5	5	é	Ω	%	5	E	U	e	u
6	6	ù	Π	&	6	F	V	f	v
7	7	ì	Ψ	‘	7	G	W	g	w
8	8	ò	Σ	(	8	H	X	h	x
9	9	Ç	Θ	)	9	I	Y	i	y
10	A	LF	Ξ	*	:	J	Z	j	z
11	B	Ø	<ESC>	+	;	K	Ä	K	ä
12	C	ø	Æ	,	<	L	Ö	l	ö
13	D	CR	æ	-	=	M	Ñ	m	ñ
14	E	Å		.	>	N	Ü	n	ü
15	F	å	É	/	?	O	§	o	à

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยถ้าต้องการส่งข้อความเป็นภาษาไทยก็สามารถทำได้ แต่จากตารางที่ 2.6 จะสังเกตเห็นว่าเป็นตารางสากลจึงไม่มีอักษรภาษาไทยอยู่ ถ้าต้องการส่งข้อความภาษาไทยจึงต้องเปลี่ยนวิธีการเข้ารหัสข้อความเป็นแบบ UCS2 (16 บิต) ซึ่งจะสามารถส่งข้อความภาษาไทยได้ โดยมีรหัสของตัวอักษรต่างๆ แสดงดังตารางที่ 2.10

ตารางที่ 2.10 ตารางรหัสตัวอักษรของการเข้ารหัสแบบ UCS2 (16 บิต)

	0E0	0E1	0E2	0E3	0E4	0E5	0E6	0E7
0	-	ฐ	ภ	ะ	เ	O	-	-
1	ก	ฑ	ม	ั	แ	๑	-	-
2	ข	ฒ	ย	า	โ	๒	-	-
3	ฃ	ณ	ร	ำ	ใ	๓	-	-
4	ค	ด	ฤ	ิ	ไ	๔	-	-
5	ฅ	ต	ล	ี	ๅ	๕	-	-
6	ฆ	ถ	ฦ	ึ	ๆ	๖	-	-
7	ง	ท	ว	ื	็	๗	-	-
8	จ	ธ	ศ	ุ	่	๘	-	-
9	ฉ	น	ษ	ู	้	๙	-	-
A	ช	บ	ส	ฺ	๊	๚	-	-
B	ซ	ป	ห	-	๋	๛	-	-
C	ฌ	ผ	ฬ	-	์	-	-	-
D	ญ	ฝ	อ	๎	๎	-	-	-
E	ฎ	พ	ฮ		๏	-	-	-
F	ฏ	ฟ	ฯ	฿	๏	-	-	-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 คำสั่ง AT Command กับโทรศัพท์มือถือ

การสื่อสารกับอุปกรณ์สื่อสารต่างๆ เช่น โมเด็มหรืออุปกรณ์ DTE (Data Terminal Equipment) นั้นสามารถใช้ชุดคำสั่งที่เป็นมาตรฐานที่เรียกว่า AT command ในการติดต่อเพื่อโต้ตอบตั้งค่าหรือสั่งอุปกรณ์เหล่านั้น ให้ทำงานตามที่ต้องการ โดยชุดคำสั่งพื้นฐานจะถูกกำหนดไว้ใน Hayes AT Command ซึ่งบริษัท Hayes เป็นผู้คิดค้นชุดคำสั่งเพื่อใช้กับโมเด็มของตนและต่อมาได้กลายเป็นมาตรฐานสำหรับผู้ผลิตโมเด็มรายอื่นๆ โดยอาจจะมีชุดคำสั่งขยาย (Extended AT Command) เพื่อใช้เป็นการเฉพาะสำหรับผู้ผลิตรายนั้นๆ

การติดต่อกับมือถือก็เช่นกันสามารถใช้ชุดคำสั่งที่กำหนดไว้ใน GSM AT Command ซึ่งมีคำสั่งเพิ่มเติมที่เหมาะสมสำหรับการใช้งานและควบคุมมือถือ และเนื่องจากมีรายละเอียดค่อนข้างมากจึงจะพูดถึงเฉพาะคำสั่งที่จำเป็นสำหรับโครงงานนี้เท่านั้น การเชื่อมต่อคอมพิวเตอร์กับมือถือ จะทำผ่านสายข้อมูล (Data Link) ซึ่งเป็นการเชื่อมต่อแบบอนุกรมโดยใช้โปรแกรมเทอร์มินอลต่างๆ เช่น ไฮเปอร์เทอร์มินอล (Hyper terminal) ของ windows ส่วนความเร็วในการสื่อสารมักจะใช้ 9600 bps

คำสั่ง AT COMMAND

AT	เช็คความพร้อมของมือถือ
AT+CMGF	เป็นคำสั่งในการเลือกโหมดของข้อความที่จะส่ง
AT+CMGS	เป็นคำสั่งที่ใช้ในการส่งข้อความไปยัง Address ที่เลือกไว้
AT+CSMS	เป็นคำสั่งที่ใช้เลือกบริการข้อความ

ตารางที่ 2.11 ลักษณะชุดคำสั่งของ AT+CMGF

คำสั่ง	ค่าตอบสนอง
AT+CMGF =?	+CMGF: (list of support <mode>s)
AT+CMGF?	+CMGF: <mode>
AT+CMGF=[<mode>]	OK/ERROR

<mode> 0 เป็นโหมดพีดียู

1 เป็นโหมดตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.12 ลักษณะชุดคำสั่งของ AT+CMGS

คำสั่ง	ค่าตอบสนอง
AT+CMGS =?	OK
If PDU mode (+CMGF=0) AT+CMGS =<length><CR>PDU is given <ctrl-Z/ES>	If sending is successful: +CMGS : <mr> If sending is not successful: +CMS ERROR :<err>

<length> ความยาวของส่วนของชุดข้อความโดยจะนับแบบ Octets

<pdu> ข้อความที่เป็นส่วนของศูนย์บริการส่งข้อความสั้นรวมกับส่วนของชุดข้อความ

<mr> จำนวนครั้งที่เราส่งข้อความสั้นหรือตัวอ้างอิงข้อความ

ตารางที่ 2.13 ลักษณะชุดคำสั่งของ AT+CSMS

คำสั่ง	ค่าตอบสนอง
AT+CSMS=?	+CSMS: (list of supported <service>s)
AT+CSMS?	+CSMS: <service>,<mt>,<mo>,<bm>
AT+CSMS=[<service>]	+CSMS : <mt>,<mo>,<bm> OK/ERROR/+CMS ERROR

<service> 0 GSM 3.40 และ 3.41

<mt> 1 รองรับรูปแบบ Mobile Terminate Massage

0 ไม่รองรับรูปแบบ Mobile Terminate Massage

<mo> 1 รองรับรูปแบบ Mobile Originatee Massage

0 ไม่รองรับรูปแบบ Mobile Originatee Massage

<bm> 1 รองรับรูปแบบ Broadcast Type Massage

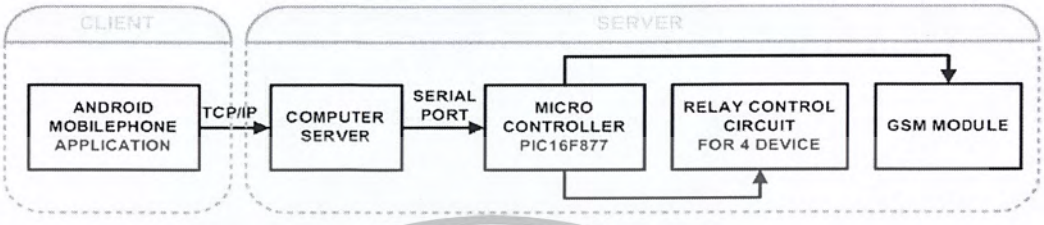
0 ไม่รองรับรูปแบบ Broadcast Type Massage

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและการจัดทำโครงงาน



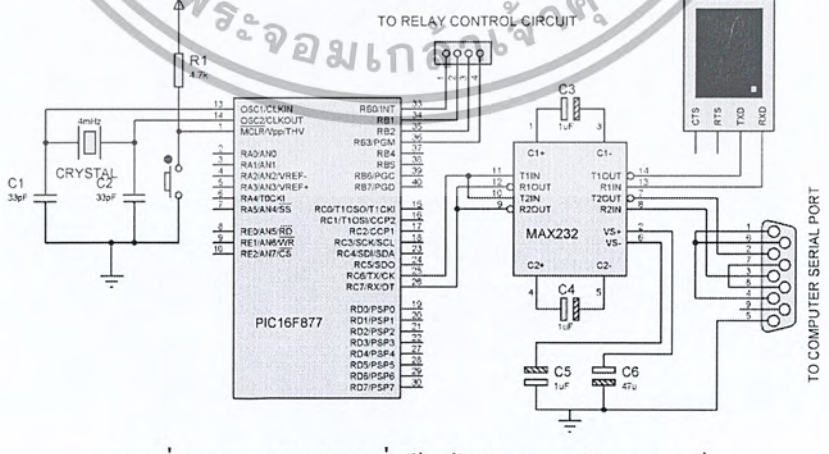
รูปที่ 3.1 บล็อกไดอะแกรมแสดงการทำงานของโครงงาน

3.1 การออกแบบ

จากบล็อกไดอะแกรมได้แสดงถึงการทำงานร่วมกันของโปรแกรมแอพพลิเคชันบนโทรศัพท์มือถือระบบปฏิบัติการแอนดรอยด์ที่จะเป็นเสมือนตัวไคลเอนท์ที่จะส่งคำสั่งควบคุมผ่านโปรโตคอล TCP/IP ไปยังคอมพิวเตอร์ที่เป็นเซิร์ฟเวอร์ เพื่อให้ไมโครคอนโทรลเลอร์รับคำสั่งดังกล่าวและนำไปควบคุมการทำงานของวงจรควบคุมอุปกรณ์ไฟฟ้าให้มีสถานะเปิด – ปิด รวมถึงการส่งข้อความสั้นยืนยันสถานะของอุปกรณ์

3.1.1 การออกแบบทางด้านอุปกรณ์

3.1.1.1 วงจรรับส่งคำสั่งด้วยไมโครคอนโทรลเลอร์

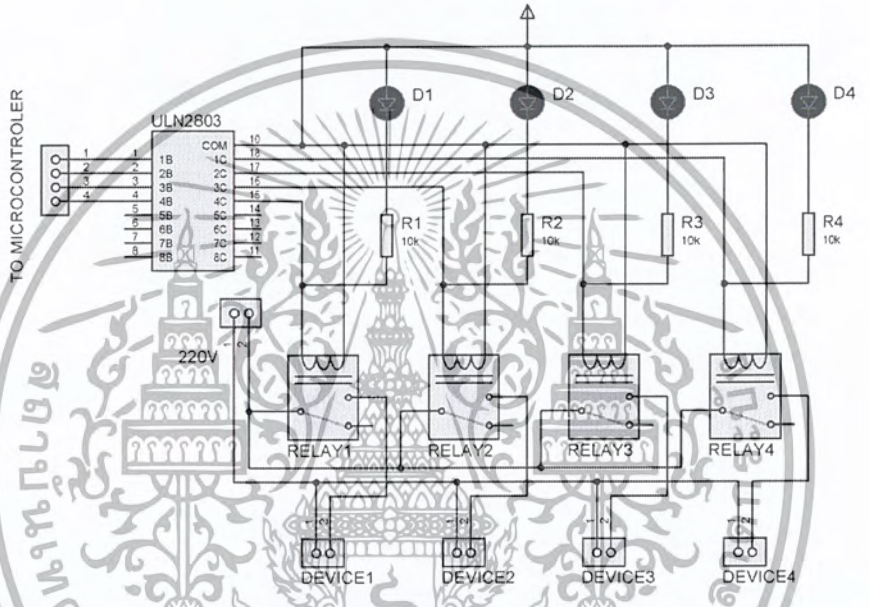


รูปที่ 3.2 วงจรรับส่งคำสั่งด้วยไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปจะพบว่าวงจรรับส่งคำสั่งด้วยไมโครคอนโทรลเลอร์ประกอบไปด้วยไมโครคอนโทรลเลอร์ PIC16F877 โดยทำการเชื่อมต่อกับพอร์ตอนุกรมของคอมพิวเตอร์เพื่อรับคำสั่งควบคุมผ่าน IC MAX232 จากนั้นก็ประมวลคำสั่งและทำการควบคุมรีเลย์เพื่อทำการเปิด-ปิดอุปกรณ์ไฟฟ้ารวมทั้งส่งข้อความสั้นจากจีเอสเอ็มโมดูล

3.1.1.2 วงจรควบคุมอุปกรณ์ไฟฟ้าด้วยรีเลย์



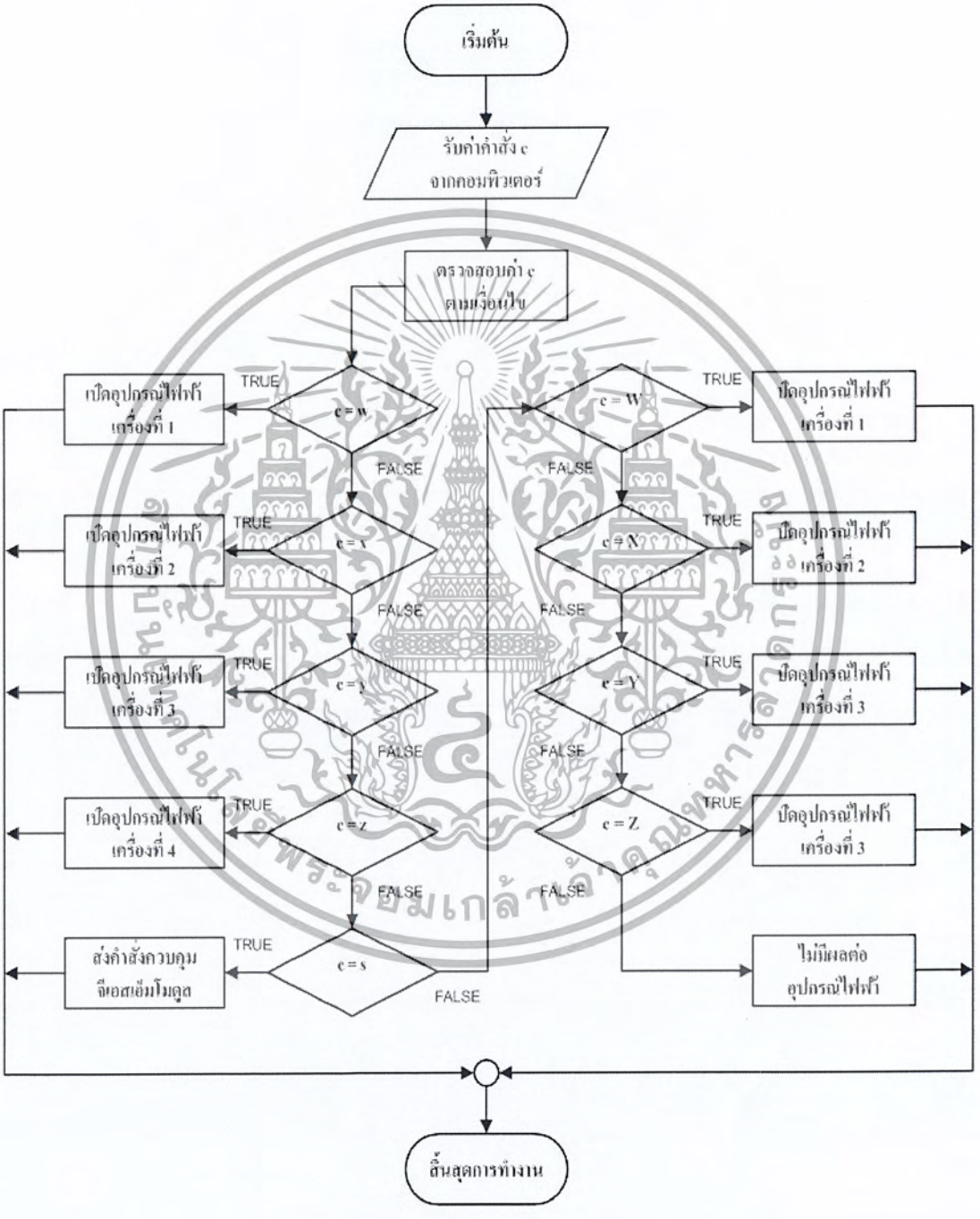
รูปที่ 3.3 วงจรควบคุมอุปกรณ์ไฟฟ้าด้วยรีเลย์

จากรูปเมื่อวงจรรับส่งสัญญาณด้วยไมโครคอนโทรลเลอร์ส่งคำสั่งควบคุมไปที่วงจรควบคุมอุปกรณ์ไฟฟ้าด้วยรีเลย์ จะต้องทำการปรับค่าแรงดันดังกล่าวก่อนเพราะมีปริมาณแรงดันและกระแสไฟฟ้าไม่เพียงพอจึงจำเป็นต้องใช้ IC ULN2803 มาปรับแรงดันเพื่อนำไปใช้ในการควบคุมรีเลย์ และใช้ LED แสดงผลการทำงานของรีเลย์ โดยเมื่อรีเลย์มีการทำงานที่ขา NO สถานะของ LED จะเปลี่ยนแปลงเป็นสว่างขึ้น ซึ่งหมายถึงอุปกรณ์ไฟฟ้าที่เชื่อมต่ออยู่กับรีเลย์นั้นจะมีสถานะเปิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 การออกแบบทางด้านโปรแกรม

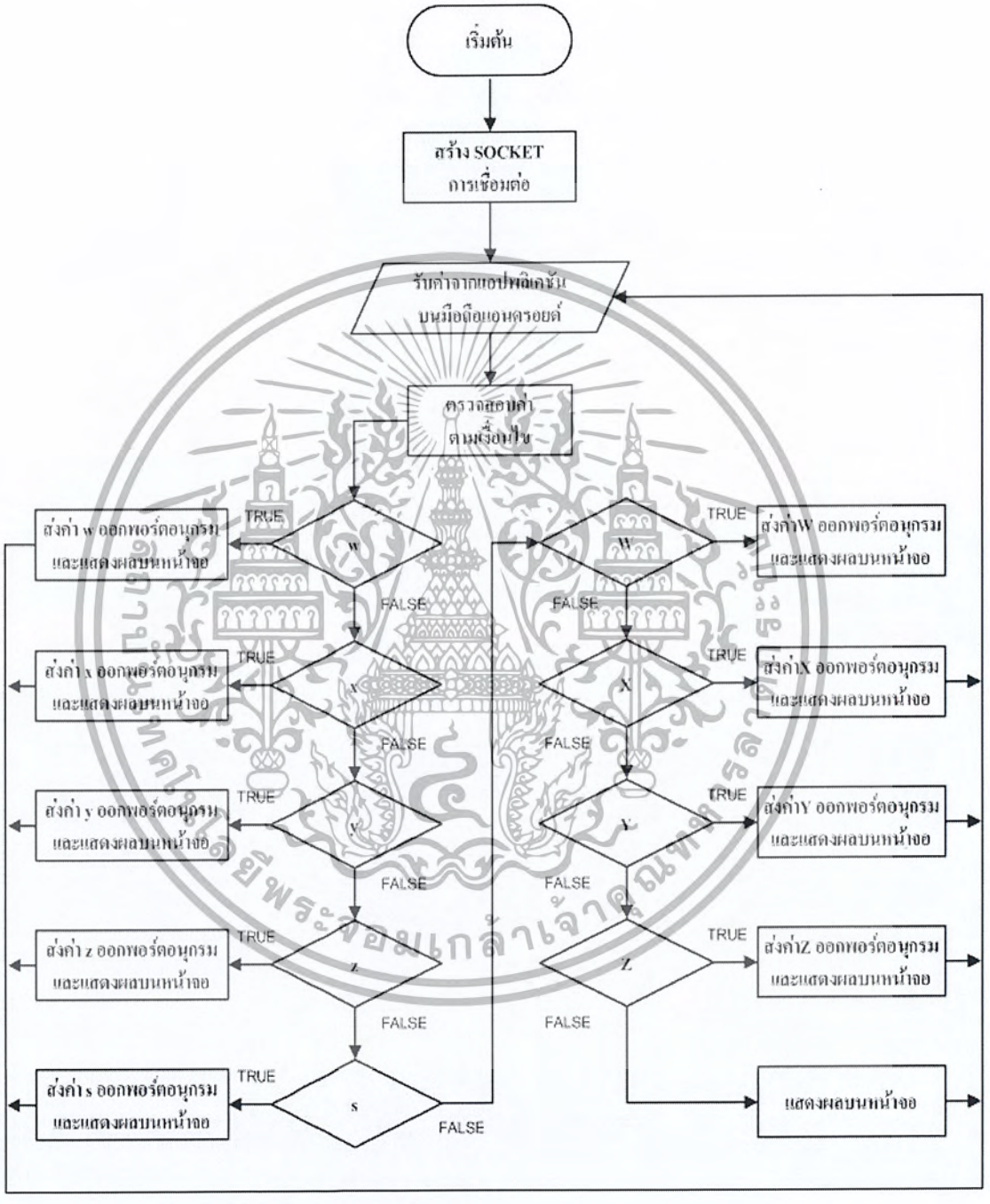
3.1.2.1 โปรแกรมการทำงานของไมโครคอนโทรลเลอร์ PIC16F877



รูปที่ 3.4 Flow chart การทำงานของไมโครคอนโทรลเลอร์ PIC16F877

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

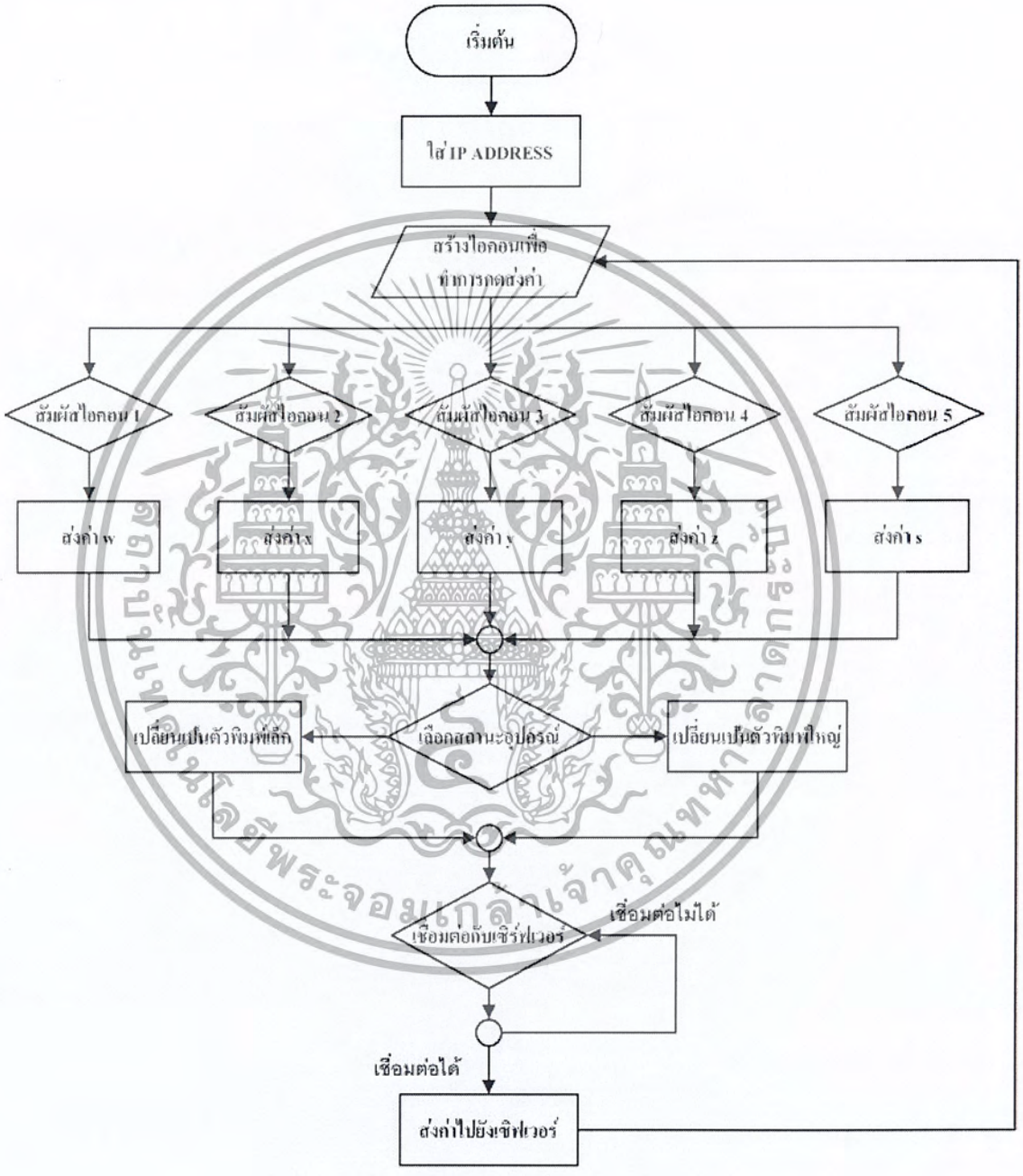
3.1.2.2 โปรแกรมการทำงานของคอมพิวเตอร์เซิร์ฟเวอร์



รูปที่ 3.5 Flow chart การทำงานของคอมพิวเตอร์เซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2.3 โปรแกรมการทำงานของแอปพลิเคชันบนโทรศัพท์มือถือระบบปฏิบัติการแอนดรอยด์ (โปรแกรมฝั่งไคลเอนท์)



รูปที่ 3.6 Flow chart การทำงานของแอปพลิเคชันบนโทรศัพท์มือถือระบบปฏิบัติการแอนดรอยด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2.3 ลักษณะหน้าจอแอปพลิเคชันบนโทรศัพท์มือถือระบบปฏิบัติการแอนดรอยด์

รูปที่ 3.7 หน้าจอแอปพลิเคชันบนโทรศัพท์มือถือระบบปฏิบัติการแอนดรอยด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 เครื่องมือที่ใช้ในการทดลอง

1. คอมพิวเตอร์
2. เครื่องจ่ายแรงดันไฟฟ้า
3. เครื่องโปรแกรมหน่วยความจำไมโครคอนโทรลเลอร์
4. จีเอสเอ็มโมดูล
5. โทรศัพท์มือถือระบบปฏิบัติการแอนดรอยด์

3.3 การจัดเก็บผลการทดลอง

3.3.1 การทดสอบชุดควบคุมการเปิด-ปิดอุปกรณ์ไฟฟ้าผ่านแอปพลิเคชัน

1) ทำการเชื่อมต่ออุปกรณ์ไฟฟ้าเข้ากับคอมพิวเตอร์ผ่านทางพอร์ตอนุกรม

2) เปิดโปรแกรมบนคอมพิวเตอร์และแอปพลิเคชันบนโทรศัพท์มือถือ

3) ทดลองสัมผัสหน้าจอบริเวณปุ่มคำสั่งและบันทึกผลสถานะของอุปกรณ์ไฟฟ้า

4) บันทึกผลหน้าจอคอมพิวเตอร์ของโปรแกรมฝั่งเซิฟร์ฟเวอร์

3.3.2 การทดสอบการรับข้อความสั้นจากจีเอสเอ็มโมดูล

1) ทำการเชื่อมต่ออุปกรณ์ไฟฟ้าเข้ากับคอมพิวเตอร์ผ่านทางพอร์ตอนุกรม

2) เปิดโปรแกรมบนคอมพิวเตอร์และแอปพลิเคชันบนโทรศัพท์มือถือ

3) ทดลองสัมผัสหน้าจอบริเวณปุ่ม SEND SMS TO MOBILEPHONE รอรับข้อความสั้นและทำการบันทึกผล

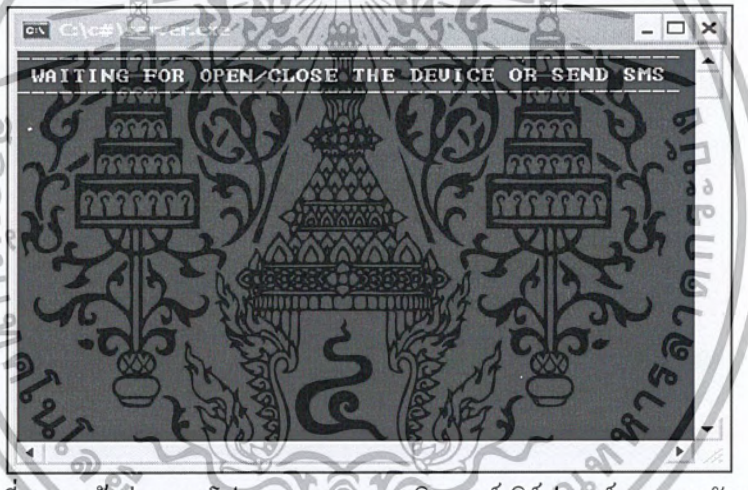
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

4.1 ผลการทดสอบชุดควบคุมการเปิด-ปิดอุปกรณ์ไฟฟ้าผ่านแอปพลิเคชัน

เริ่มเปิดโปรแกรมบนคอมพิวเตอร์ ซึ่งถูกเขียนด้วยภาษา C# บนโปรแกรม Microsoft .NET framework V2.0 ซึ่งทำหน้าที่เป็นเซิร์ฟเวอร์ในการรอรับค่าจากแอปพลิเคชันบนโทรศัพท์ เพื่อส่งค่าผ่านพอร์ตอนุกรมไปยังวงจรควบคุม จะพบว่ามีหน้าต่างรอรับค่าดังแสดงในรูป



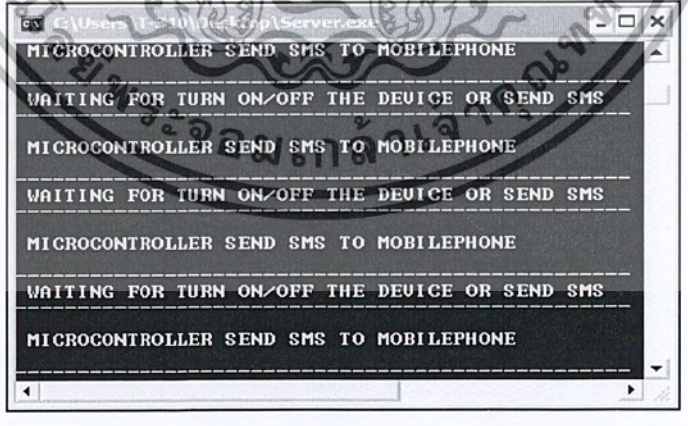
รูปที่ 4.1 หน้าต่างของโปรแกรมบนคอมพิวเตอร์เซิร์ฟเวอร์ ขณะรอรับค่า

ทำการเปิดแอปพลิเคชันบนโทรศัพท์มือถือระบบปฏิบัติการแอนดรอย์ ที่ถูกพัฒนาด้วยภาษา Java ซึ่งเปรียบเสมือนกับตัวไคลเอนท์ที่เป็นตัวส่งค่าคำสั่งควบคุมไปยังเครื่องคอมพิวเตอร์เซิร์ฟเวอร์ จะพบว่าหน้าตาแอปพลิเคชันบนหน้าจอโทรศัพท์ขนาด 3.2 นิ้วมีลักษณะดังรูปที่ 4.2 ซึ่งอธิบายการใช้งานไว้ในรูปที่ 3.7 หัวข้อที่ 3.1.2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.2 แอปพลิเคชันบนหน้าจอโทรศัพท์มือถือระบบปฏิบัติการแอนดรอยด์

จากนั้นทำการใส่ IP Address ซึ่งในการทดลองนี้ใช้ 161.246.18.39 และทำการทดลองสัมผัสหน้าจอที่ไอคอนต่างๆตามลำดับการทำงานจะพบว่า สถานะของอุปกรณ์ที่ใช้ความสว่างของหลอดไฟเป็นตัวบ่งชี้ กับการทำงานของแอปพลิเคชันและคอมพิวเตอร์มีความสัมพันธ์กันตามตารางที่ 4.1 และเมื่อเซิฟเวอร์ได้รับค่าคำสั่งจากแอปพลิเคชันก็จะแสดงผลออกทางหน้าจอดังรูปที่ 4.4



รูปที่ 4.4 หน้าต่างของโปรแกรมบนคอมพิวเตอร์เซิร์ฟเวอร์ ขณะทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 ผลการทำงานระหว่างเซิร์ฟเวอร์ แอปพลิเคชัน และ สถานะอุปกรณ์

ไอคอนที่ถูกสัมผัสบนหน้าจอโทรศัพท์	ตำแหน่งของตัวอุปกรณ์				ข้อความบนหน้าต่างโปรแกรมคอมพิวเตอร์เซิร์ฟเวอร์
	1	2	3	4	
:1: + TURN ON	ON	OFF	OFF	OFF	TURN ON FISRT DEVICE
:2: + TURN ON	ON	ON	OFF	OFF	TURN ON SECOND DEVICE
:3: + TURN ON	ON	ON	ON	OFF	TURN ON THIRD DEVICE
:4: + TURN ON	ON	ON	ON	ON	TURN ON FOURTH DEVICE
:1: + TURN OFF	OFF	ON	ON	ON	TURN OFF FIRST DEVICE
:2: + TURN OFF	OFF	OFF	ON	ON	TURN OFF SECOND DEVICE
:3: + TURN OFF	OFF	OFF	OFF	ON	TURN OFF THIRD DEVICE
:4: + TURN OFF	OFF	OFF	OFF	OFF	TURN OFF FOURTH DEVICE

จากตารางจะพบว่า สถานะอุปกรณ์จะเปิดก็ต่อเมื่อทำการสัมผัสไอคอนเลือกตัวอุปกรณ์ ตามด้วยไอคอนควบคุมการเปิด และ สถานะอุปกรณ์จะปิดก็ต่อเมื่อทำการสัมผัสไอคอนเลือกตัวอุปกรณ์ตามด้วยไอคอนควบคุมการปิด โดยสถานะการเปิดหรือปิดนั้นนอกจากจะแสดงบนหน้าต่างโปรแกรมบนเซิร์ฟเวอร์แล้ว ก็ยังแสดงบนหน้าจอแอปพลิเคชันด้วยดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.5 อุปกรณ์ตัวที่ 1 อยู่ในสถานะ ON

รูปที่ 4.6 อุปกรณ์ตัวที่ 2 อยู่ในสถานะ ON

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.7 อุปกรณ์ตัวที่ 3 อยู่ในสถานะ ON

รูปที่ 4.8 อุปกรณ์ตัวที่ 4 อยู่ในสถานะ ON

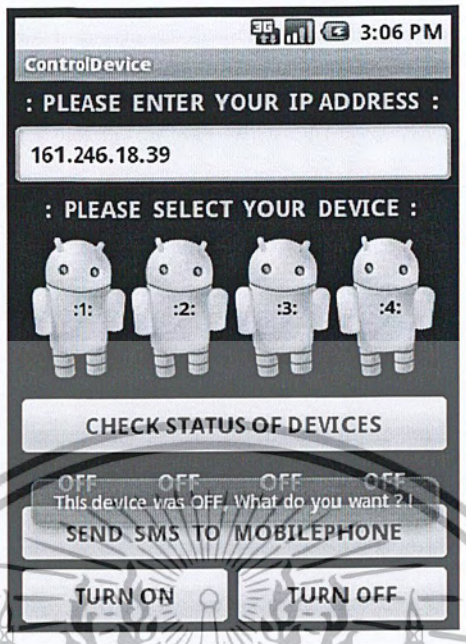
และเมื่อทดสอบสัมผัสหน้าจอโทรศัพท์มือถือโดยที่ไม่ใส่ IP Address แอปพลิเคชันจะทำการแจ้งเตือนในรูปแบบของกล่องข้อความ ในทางเดียวกันนั้นหากมีการส่งคำสั่งในสถานะเดิมของอุปกรณ์ หรือ ไม่ได้ทำการเลือกอุปกรณ์ก่อนที่จะควบคุม แอปพลิเคชันจะทำการแจ้งเตือนว่า “This device was on, You ought to close !!” หรือ “This device was off, You ought to open !!” และ “Please select your device” ตามลำดับ ดังแสดงในรูปที่ 4.9 4.10 4.11 และ 4.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.9 แอปพลิเคชันแสดงกล่องข้อความเมื่อไม่ได้ใส่ IP Address หรือ ใส่ผิด

รูปที่ 4.10 แอปพลิเคชันแสดงกล่องข้อความเมื่ออุปกรณ์ที่ถูกเลือกให้เปิดนั้นได้ถูกเปิดก่อนหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.11 แอปพลิเคชันแสดงกล่องข้อความเมื่ออุปกรณ์ที่ถูกเลือกให้ปิดนั้น ได้ถูกปิดก่อนหน้า

รูปที่ 4.12 แอปพลิเคชันแสดงกล่องข้อความเมื่อไม่ได้เลือกอุปกรณ์ก่อนทำการออกคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

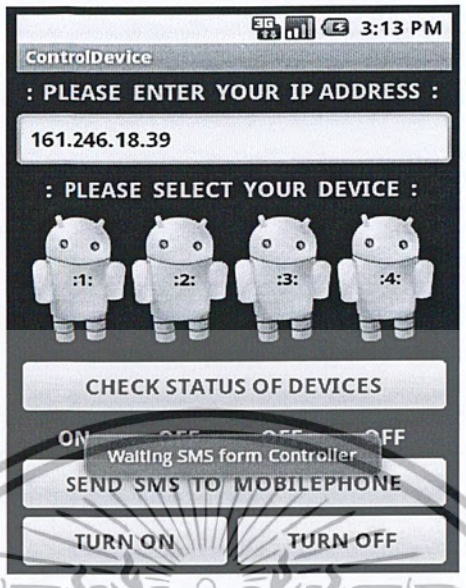
4.2 ผลการทดสอบการรับข้อความสั้นจากจีเอสเอ็มโมดูล

เมื่อทำการใส่ IP Address และ ทำการเปิด – ปิดอุปกรณ์ผ่านแอปพลิเคชัน จากนั้นสัมผัสที่ไอคอน SEND SMS TO MOBILE PHONE จะพบว่าไอคอนคำสั่ง ON และ OFF จะถูกเปลี่ยนเป็น SEND และ DON'T SEND เพื่อรอรับคำสั่งให้ทำการส่งข้อความสั้นดังรูปที่ 4.13 และเมื่อสัมผัสที่ไอคอน SEND แอปพลิเคชันจะทำการส่งค่าไปยังเซิร์ฟเวอร์ที่เชื่อมต่อกับไมโครคอนโทรลเลอร์เพื่อออกคำสั่งให้จีเอสเอ็มโมดูล ทำการส่งข้อความสั้นไปยังหมายเลขที่ได้ทำการตั้งค่าไว้ในการโปรแกรมไมโครคอนโทรลเลอร์ หลังจากนั้นแอปพลิเคชันจะแสดงกล่องข้อความ "SMS is sent" และไอคอนจะเปลี่ยนกลับไปเป็น ON และ OFF เช่นเดิมดังรูป 4.14

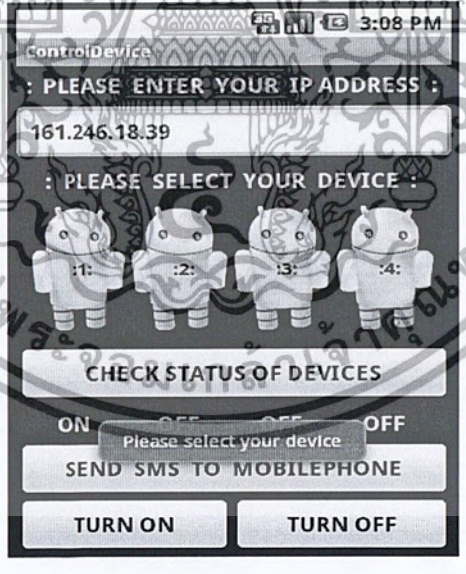
ในทางเดียวกันหากสัมผัสที่ไอคอน NOT SEND แอปพลิเคชันจะแสดงกล่องข้อความ "Please select your device" และไอคอนจะเปลี่ยนกลับไปเป็น ON และ OFF เพื่อรอรับคำสั่งที่ต้องการต่อไปดังรูป 4.15 หลังจากนั้นข้อความสั้นจะส่งไปถึงยังโทรศัพท์มือถือซึ่งในข้อความจะกล่าวถึงอุปกรณ์ที่ถูกเปิดเท่านั้นเช่น เมื่ออุปกรณ์ตัวที่ 1 และ 3 ถูกเปิดจะได้รับข้อความว่า "Open The 1 : 3 : Device" เป็นต้น

รูปที่ 4.13 ไอคอนบนแอปพลิเคชันเปลี่ยนแปลงเพื่อเตรียมส่งข้อความเมื่อสัมผัสไอคอน SEND SMS TO MOBILEPHONE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



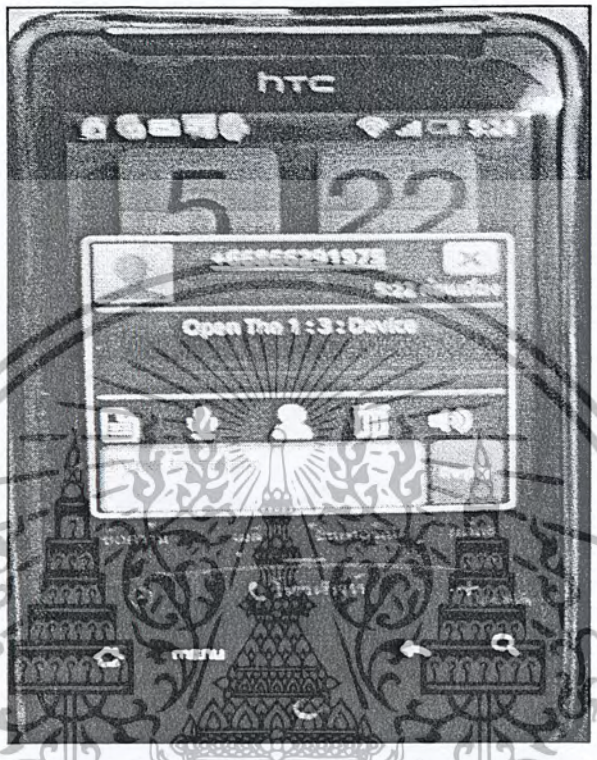
รูปที่ 4.14 แอปพลิเคชันแสดงกล่องข้อความหลังจากสั่งให้ส่งข้อความสั้น และเปลี่ยนไอคอนกลับคืน



รูปที่ 4.15 แอปพลิเคชันแสดงกล่องข้อความหลังจากสั่งไม่เลือกการส่งข้อความสั้น และเปลี่ยนไอคอนกลับคืน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากนั้นข้อความสั้นจะส่งไปถึงยังโทรศัพท์มือถือซึ่งในข้อความจะกล่าวถึงอุปกรณ์ที่ถูกเปิดเท่านั้น ดังตัวอย่างที่แสดงในรูปที่ 4.15



รูปที่ 4.15 ข้อความสั้นบนหน้าจอโทรศัพท์มือถือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

แอปพลิเคชันที่พัฒนาจากโปรแกรมภาษาจาวาบนโทรศัพท์มือถือระบบปฏิบัติการแอนดรอยด์ สามารถส่งคำสั่งควบคุมการเปิด-ปิดอุปกรณ์ไฟฟ้าผ่านโปรโตคอล TCP/IP ไปยังเซิร์ฟเวอร์ที่เชื่อมต่อกับวงจรควบคุมเพื่อทำการเปิด-ปิดอุปกรณ์ไฟฟ้าจำนวน 4 ตัว พร้อมทั้งสามารถทราบสถานะของตัวอุปกรณ์ได้จากข้อความสั้นที่ส่งไปยังผู้ใช้งานด้วยจีเอสเอ็มโมดูล

5.2 ข้อเสนอแนะ

1. พัฒนาชุดควบคุมการเปิด-ปิดอุกรณ์ไฟฟ้าให้มีประสิทธิภาพมากขึ้น โดยอาจเลือกใช้ไมโครคอนโทรลเลอร์ที่สามารถใช้งานอินเตอร์เน็ตได้ในตัว เพื่อลดความสิ้นเปลืองในการใช้คอมพิวเตอร์ที่เป็นเซิร์ฟเวอร์

2. พัฒนาแอปพลิเคชันบนโทรศัพท์มือถือระบบปฏิบัติการแอนดรอยด์ ให้สามารถควบคุมชุดอุปกรณ์ควบคุมการเปิด-ปิดอุปกรณ์ไฟฟ้า ให้ใช้งานได้ง่ายและมีประสิทธิภาพมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] กิตติ ภักดีวัฒนะกุล. *คัมภีร์JAVA เล่ม1*. พิมพ์ครั้งที่ 3. กรุงเทพฯ : เคทีพี คอมพ์ แอนด์ คอนซัลท์, 2546.

[2] ดอนสัน ปงผาบ, และทิพวัลย์ คำน้ำนอง. *ไมโครคอนโทรลเลอร์และการประยุกต์ใช้งาน PIC*. พิมพ์ครั้งที่ 4. กรุงเทพฯ : สำนักพิมพ์ ส.ส.ท, 2553.

[3] Mark L. Murphy. *Beginnig Android*. 1 edition. Apress, 2009.

[4] Mark L. Murphy. *Beginnig Android 2*. 1 edition. Apress, 2009.

[5] Edd Bernette.*Hello, Android: Introducing Google's Mobile Development Platform*. 3 edition. Pragmatic Bookshelf, 2010.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

ไมโครคอนโทรลเลอร์ PIC16F877

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MICROCHIP

PIC16F87X

28/40-Pin 8-Bit CMOS FLASH Microcontrollers

Devices Included in this Data Sheet:

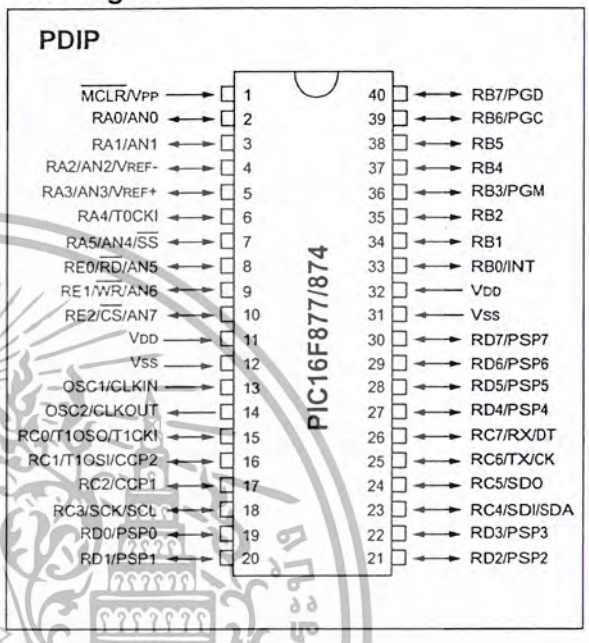
- PIC16F873
- PIC16F874
- PIC16F876
- PIC16F877

Microcontroller Core Features:

- High performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input
  DC - 200 ns instruction cycle
- Up to 8K x 14 words of FLASH Program Memory,
  Up to 368 x 8 bytes of Data Memory (RAM)
  Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to the PIC16C73B/74B/76/77
- Interrupt capability (up to 14 sources)
- Eight level deep hardware stack
- Direct, indirect and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low power, high speed CMOS FLASH/EEPROM technology
- Fully static design
- In-Circuit Serial Programming™ (ICSP) via two pins
- Single 5V In-Circuit Serial Programming capability
- In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA
- Commercial, Industrial and Extended temperature ranges
- Low-power consumption:
  - < 0.6 mA typical @ 3V, 4 MHz
  - 20 μA typical @ 3V, 32 kHz
  - < 1 μA typical standby current

Pin Diagram

PDIP



Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during SLEEP via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Two Capture, Compare, PWM modules
  - Capture is 16-bit, max. resolution is 12.5 ns
  - Compare is 16-bit, max. resolution is 200 ns
  - PWM max. resolution is 10-bit
- 10-bit multi-channel Analog-to-Digital converter
- Synchronous Serial Port (SSP) with SPI™ (Master mode) and I^2C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection
- Parallel Slave Port (PSP) 8-bits wide, with external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for Brown-out Reset (BOR)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Key Features PICmicro™ Mid-Range Reference Manual (DS33023)	PIC16F873	PIC16F874	PIC16F876	PIC16F877
Operating Frequency	DC - 20 MHz	DC - 20 MHz	DC - 20 MHz	DC - 20 MHz
RESETS (and Delays)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)
FLASH Program Memory (14-bit words)	4K	4K	8K	8K
Data Memory (bytes)	192	192	368	368
EEPROM Data Memory	128	128	256	256
Interrupts	13	14	13	14
I/O Ports	Ports A,B,C	Ports A,B,C,D,E	Ports A,B,C	Ports A,B,C,D,E
Timers	3	3	3	3
Capture/Compare/PWM Modules	2	2	2	2
Serial Communications	MSSP, USART	MSSP, USART	MSSP, USART	MSSP, USART
Parallel Communications	—	PSP	—	PSP
10-bit Analog-to-Digital Module	5 input channels	8 input channels	5 input channels	8 input channels
Instruction Set	35 instructions	35 instructions	35 instructions	35 instructions

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

โปรแกรมภาษาจาวาในการเขียนแอปพลิเคชันแอนดรอยด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

// MAIN JAVA CODE

```
package control.device;
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.EditText;
import android.widget.Toast;
import java.io.*;
import java.net.*;
public class Controldevice extends Activity
{
      EditText e1 = null;
      Button first = null ;
      Button second = null ;
      Button third = null ;
      Button fourth = null;
      Button sms = null ;
      Button open =null;
      Button close = null;
      public TextView thetext = null;
      public static String IPSER = null ;
      public final static int PORT = 7000;
      public Socket androidsocket;
      public static int i = 0;
      public static String stat = "Please Select Your Device";
      public static String coma = "";
      public DataOutputStream controlout = null;

    @Override
    public void onCreate(Bundle savedInstanceState)
      {
          super.onCreate(savedInstanceState);
          setContentView(R.layout.main);

        e1 = (EditText)findViewById(R.id.EditText01);
        open = (Button)findViewById(R.id.open);
        close = (Button)findViewById(R.id.close);

        first = (Button) findViewById(R.id.first);
        first.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                        thetext = (TextView)findViewById(R.id.st1);
                        stat = "The 1st is";
                        i = 1;
                        coma = "w";
           }});

        second = (Button) findViewById(R.id.second);
        second.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                        thetext = (TextView)findViewById(R.id.st2);
                        stat = "The 2nd is";
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    i = 1;
                coma = "x";
    }});

        third = (Button) findViewById(R.id.third);
        third.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                    thetext = (TextView)findViewById(R.id.st3);
                    i = 1;
                    stat = "The 3rd is";
                coma = "y";
    }});

        fourth = (Button) findViewById(R.id.fourth);
        fourth.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                thetext = (TextView)findViewById(R.id.st4);
                    i = 1;
                    stat = "The 4th is";
                coma = "z";
    }});

        sms = (Button) findViewById(R.id.sms);
        sms.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                    thetext = (TextView)findViewById(R.id.st4);
                    i = 2;
                    open.setText("SEND");
                    close.setText("NOT SEND");
                    stat = "SMS is sent";
                    coma = "s";
            }});

        open.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                        coma = coma.toLowerCase();
                        IPSER = e1.getText().toString();
                if(i==1){
                    try{ androidsocket = new Socket(IPSER,PORT);
                                controlout = new
DataOutputStream(androidsocket.getOutputStream());
                                controlout.writeBytes(coma);

    Toast.makeText(Controldevice.this,stat+"
opened",Toast.LENGTH_SHORT).show();
                                i=0;
                                stat = "Please select your device";
                                thetext.setText("ON");
                    }catch(IOException
e){Toast.makeText(Controldevice.this,"Error
from"+e,Toast.LENGTH_SHORT).show(); }}
                else if(i==2){
                    try { androidsocket = new Socket(IPSER,PORT);
                                controlout = new
DataOutputStream(androidsocket.getOutputStream());
                                controlout.writeBytes(coma);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      Toast.makeText(Controldevice.this,stat,Toast.LENGTH_SHORT).sho
w();
                                    i=0;
                                    open.setText("OPEN");
                                close.setText("CLOSE");
                                    stat = "Please select your device";
                           }catch(IOException e){
Toast.makeText(Controldevice.this,"Error
from"+e,Toast.LENGTH_SHORT).show(); }}
                    else {Toast.makeText(Controldevice.this,stat,
Toast.LENGTH_SHORT).show();}}}});

          close.setOnClickListener(new View.OnClickListener() {
              public void onClick(View v) {
                                    coma = coma.toUpperCase();
                                    IPSER = e1.getText().toString();
                    if(i==1){
                         try {androidsocket = new Socket(IPSER,PORT);
                                    controlout = new
DataOutputStream(androidsocket.getOutputStream());
                                    controlout.writeBytes(coma);

      Toast.makeText(Controldevice.this,stat+"
closed",Toast.LENGTH_SHORT).show();
                                    i=0;
                                    stat = "Please select your device";
                                    thetext.setText("OFF");
                              }catch(IOException
e){Toast.makeText(Controldevice.this,"Error
from"+e,Toast.LENGTH_SHORT).show(); }}
                    else if(i==2){
                                    stat = "Please select your device";
                                    open.setText("OPEN");
                                    close.setText("CLOSE");

      Toast.makeText(Controldevice.this,stat,Toast.LENGTH_SHORT).sho
w(); }
                    else {Toast.makeText(Controldevice.this,stat,
Toast.LENGTH_SHORT).show();}}});

          }
    }
```

// R.java Code for layout of Application

```
<?xml version="1.0" encoding="utf-8"?>

<AbsoluteLayout android:id="@+id/AbsoluteLayout01"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
xmlns:android="http://schemas.android.com/apk/res/android">

<TextView android:layout_x="7dip"
android:text=": PLEASE ENTER YOUR IP ADDRESS :"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
android:textSize="17dip"
android:textStyle = "bold"
android:layout_height="wrap_content"
android:id="@+id/TextView01"
android:layout_width="wrap_content"
android:layout_y="10dip"></TextView>

<TextView android:layout_x="2dip"
android:text="     :  PLEASE  SELECT  YOUR  DEVICE  :"
android:textSize="17dip"
android:textStyle = "bold"
android:layout_height="wrap_content"
android:id="@+id/TextView02"
android:layout_width="wrap_content"
android:layout_y="100dip"></TextView>

<EditText android:layout_x="0dip"
android:text=""
android:textSize="17dip"
android:textStyle = "bold"
android:layout_height="50dip"
android:id="@+id/EditText01"
android:layout_width="320dip"
android:layout_y="43dip"></EditText>

<Button android:text="OPEN"
android:layout_height="50dip"
android:id="@+id/open"
android:textSize="17dip"
android:textStyle = "bold"
android:layout_width="158dip"
android:layout_x="0dip"
android:layout_y="385dip"></Button>

<Button android:text="CLOSE"
android:layout_height="50dip"
android:textSize="17dip"
android:id="@+id/close"
android:textStyle = "bold"
android:layout_width="158dip"
android:layout_x="162dip"
android:layout_y="385dip"></Button>

<Button android:id="@+id/sms"
android:textStyle="bold"
android:layout_height="50dip"
android:textSize="17dip"
android:layout_width="320dip"
android:text="SEND  SMS  TO  MOBILEPHONE"
android:layout_x="0dip"
android:layout_y="325dip"></Button>
<Button android:textSize="15dip" android:id="@+id/third"
android:textStyle="bold" android:layout_width="90dip"
android:text=":3:" android:layout_height="110dip"
android:background="@drawable/icon2" android:layout_x="155dip"
android:layout_y="145dip"></Button>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
<Button android:textSize="15dip" android:id="@+id/first"
android:textStyle="bold" android:layout_width="90dip"
android:text=":1:" android:layout_height="110dip"
android:background="@drawable/icon2" android:layout_x="5dip"
android:layout_y="145dip"></Button>
<Button android:textSize="15dip" android:id="@+id/second"
android:textStyle="bold" android:layout_width="90dip"
android:text=":2:" android:layout_height="110dip"
android:background="@drawable/icon2" android:layout_x="80dip"
android:layout_y="145dip"></Button>
<Button android:textSize="15dip" android:id="@+id/fourth"
android:textStyle="bold" android:layout_width="90dip"
android:text=":4:" android:layout_height="110dip"
android:background="@drawable/icon2" android:layout_x="230dip"
android:layout_y="145dip"></Button>

<TextView android:layout_x="30dip"
android:text="OFF"
android:textSize="17dip"
android:textStyle = "bold"
android:layout_height="wrap_content"
android:id="@+id/st1"
android:layout_width="wrap_content"
android:layout_y="280dip"></TextView>

<TextView android:layout_x="105dip"
android:text="OFF"
android:textSize="17dip"
android:textStyle = "bold"
android:layout_height="wrap_content"
android:id="@+id/st2"
android:layout_width="wrap_content"
android:layout_y="280dip"></TextView>

<TextView android:layout_x="180dip"
android:text="OFF"
android:textSize="17dip"
android:textStyle = "bold"
android:layout_height="wrap_content"
android:id="@+id/st3"
android:layout_width="wrap_content"
android:layout_y="280dip"></TextView>

<TextView android:layout_x="255dip"
android:text="OFF"
android:textSize="17dip"
android:textStyle = "bold"
android:layout_height="wrap_content"
android:id="@+id/st4"
android:layout_width="wrap_content"
android:layout_y="280dip"></TextView>

</AbsoluteLayout>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

// Android Manifest code

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
      package="control.device"
      android:versionCode="1"
      android:versionName="1.0">
    <uses-sdk android:minSdkVersion="7" />

    <application android:icon="@drawable/icon"
android:label="@string/app_name">
        <activity android:name=".controldevice"
                  android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category
android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

    </application>
</manifest>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

โปรแกรมภาษา C# สำหรับเครื่องเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
using System;
using System.Net;
using System.Net.Sockets;
using System.Text;
using System.IO;
using System.IO.Ports;
class TcpServerSample {
public static void Main (string[] args)
      {while(true){
TcpListener tcpServer = new TcpListener(IPAddress.Any,7000);
Console.WriteLine("-----------------------------------------------");
Console.WriteLine(" WAITING FOR OPEN/CLOSE THE DEVICE OR SEND SMS");
Console.WriteLine("-----------------------------------------------"+"\n");
tcpServer.Start();
TcpClient tcpClient = tcpServer.AcceptTcpClient();
NetworkStream ns = tcpClient.GetStream();
byte[] inData = new byte[1024];
int nRead = ns.Read(inData, 0, inData.Length);
string instring = Encoding.ASCII.GetString(inData,0, nRead);
SerialPort sp = new SerialPort("COM1", 9600, Parity.None, 8, StopBits.One);
switch (instring){
            case "w" :  Console.WriteLine(" OPEN FIRST DEVICE"+"\n");
                        sp.Handshake = Handshake.None;
                        sp.Open();
                        sp.Write(instring);
                        sp.Write("\r");
                        sp.Close();       break;
            case "W" :  Console.WriteLine(" CLOSE FIRST DEVICE"+"\n");
                        sp.Handshake = Handshake.None;
                        sp.Open();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    sp.Write(instring);
                    sp.Write("\r");
                    sp.Close();       break;
case "x" :  Console.WriteLine(" OPEN SECOND DEVICE"+"\n");
                    sp.Handshake = Handshake.None;
                    sp.Open();
                    sp.Write(instring);
                    sp.Write("\r");
                    sp.Close();       break;
case "X" :  Console.WriteLine(" CLOSE SECOND DEVICE"+"\n");
                    sp.Handshake = Handshake.None;
                    sp.Open();
                    sp.Write(instring);
                    sp.Write("\r");
                    sp.Close();       break;
case "y" :  Console.WriteLine(" OPEN THIRD DEVICE"+"\n");
                    sp.Handshake = Handshake.None;
                    sp.Open();
                    sp.Write(instring);
                    sp.Write("\r");
                    sp.Close();       break;
case "Y" :  Console.WriteLine(" CLOSE THIRD DEVICE"+"\n");
                    sp.Handshake = Handshake.None;
                    sp.Open();
                    sp.Write(instring);
                    sp.Write("\r");
                    sp.Close();       break;
case "z" :  Console.WriteLine(" OPEN FOURTH DEVICE"+"\n");
                    sp.Handshake = Handshake.None;
                    sp.Open();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    sp.Write(instring);
                    sp.Write("\r");
                    sp.Close();       break;
            case "Z" :  Console.WriteLine(" CLOSE FOURTH DEVICE"+"\n");
                    sp.Handshake = Handshake.None;
                    sp.Open();
                    sp.Write(instring);
                    sp.Write("\r");
                    sp.Close();       break;
            case "s" :  Console.WriteLine(" MICROCONTROLLER SEND SMS TO
      MOBILEPHONE"+"\n");
                    sp.Handshake = Handshake.None;
                    sp.Open();
                    sp.Write(instring);
                    sp.Write("\r");
                    sp.Close();break;
            default :   Console.WriteLine(" NOT CONNECT TO DEVICE"+"\n");
            break; }
      ns.Close();
      tcpClient.Close();
      tcpServer.Stop();
}}}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง

โปรแกรมภาษา C สำหรับไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include<16F877.h>
#fuses HS,NOWDT,NOPUT,NOPROTECT
#use delay (clock=4000000)
#use rs232(baud=9600,parity=N,xmit=PIN_C6,rcv=PIN_C7,bits=8,errors)
char sms1,sms2,sms3,sms4;
char c;
void main (){  while(1){
       c = getc();
      switch (c) {
case 'w' :   output_high(pin_B7);
             sms1 = '1';                 break;
case 'x' :   output_high(pin_B6);
             sms2=  '2';                 break;
case 'y' :   output_high(pin_B5);
             sms3 = '3';                 break;
case 'z' :   output_high(pin_B4);
             sms4 = '4';                 break;
case 'W' :   output_low(pin_B7);
             sms1 = 'A';                 break;
case 'X' :   output_low(pin_B6);
             sms2 = 'A';                 break;
case 'Y' :   output_low(pin_B5);
             sms3 = 'A';                 break;
case 'Z' :   output_low(pin_B4);
             sms4 = 'A';                 break;
case 's' :   printf("AT\r\n");
             delay_ms(300);
             printf("AT+CMGF=0\r\n");
             delay_ms(300);
             printf("AT+CSMS=0\r\n");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            delay_ms(300);
            printf("AT+CMGS=60\r\n");
            delay_ms(3000);
            printf("\b0011000B916698677739030008AA2E");
            delay_ms(300);
            printf("004F00700065006E00200054006800650020");
            delay_ms(300);
            printf("003");
            printf("%c",sms1);
            printf("0020");
            printf("003");
            delay_ms(300);
            printf("%c",sms2);
            printf("0020");
            printf("003");
            delay_ms(300);
            printf("%c",sms3);
            printf("0020");
            printf("003");
            delay_ms(300);
            printf("%c",sms4);
            printf("0020");
            delay_ms(300);
            printf("004400650076006900630065");
            delay_ms(300);
            printf("%c""\r\n",0X1A);              break;
default   :                                  break;
 }}}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก จ

จีเอสเอ็ม โมดูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

wismo®

FASTRACK
GSM/GPRS - M1206

Hardware Platform

ADD WIRELESS CONNECTIVITY TO ALMOST ANYTHING

The Wavecom Fastrack M1206 external modem delivers all the power of state-of-the-art WISMO technology in one unit. Give your application instant GPRS class 10 capabilities by using the embedded TCP/IP protocol stack. Dramatically reduce time to market thanks to full type-approval.

Housed in a rugged metallic casing, the Fastrack M1206 modem is built to withstand the toughest environments. Avoid extra components and subassemblies by embedding your application right on the platform, using Open AT development tools and built-in spare processing and memory capacity.

The Wavecom Fastrack M1206 modem has been optimized for use as the hardware platform for the Wavecom Machine-to-Machine Development Lab.

SIMPLE, INEXPENSIVE MACHINE-TO-MACHINE APPLICATIONS

Get a reliable GSM/GPRS Class 10 solution
With rugged housing and proven WISMO Within technology

Get connected
With a simple serial RS232 cable and the optional embedded TCP/IP stack

Save time
Because the Wavecom Fastrack M1206 is fully type-approved

Eliminate extra components & subassemblies
By using Open AT development tools to embed your application right on the modem

เอกสารนี้จัดทำขึ้นที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

wavecom®

WIRELESS SOLUTIONS FOR EVERYONE

FASTRACK M1206 SPECIFICATIONS

PRODUCT FEATURES

Dual Band GSM modem (EGSM900/ 1800 MHz) designed for data, fax, SMS and voice applications
Fully Type Approved
Fully compliant with ETSI GSM Phase 2+ specifications (Normal MS)
Output power:
Class 4 (2W @ 900 MHz)
Class 1 (1W @ 1800 MHz)
Power supply:
Input voltage: 5.5-32V
- 5mA in idle mode, 140mA in communication GSM 900 @ 12V
- 5mA in idle mode, 100mA in communication GSM 1800 @ 12V
- Peak 1,7A @ 5.5V

Overall dimensions: 98 x 54 x 25mm
Weight: 105g

VOICE, DATA/FAX, SHORT MESSAGE SERVICES

Voice features:
- Telephony
- Emergency calls
- Full Rate, Enhanced Full Rate and Half Rate (FR/EFR/HR)
- Dual Tone Multi Frequency Function (DTMF)

GSM Data/Fax features:
- Data circuit asynchronous, transparent and non transparent up to 14,400 bits/s
- Automatic fax group 3 (Class 1 and Class 2)
- MNP2, V.42bis

GPRS packet Data features:
- GPRS Class 10

Coding schemes: CS1 to CS4
Compliant with SMG31bis
- Optional embedded TCP/IP stack

Short Messages Services features:
- Text and PDU
- Point to point (MT/MO)
- Cell Broadcast

GSM SUPPLEMENTARY SERVICES

- Call Forwarding
- Call Barring
- Multiparty
- Call Waiting and Call Hold
- Calling Line Identity
- Advice of Charge
- USSD
- Closed User Group
- Explicit Call Transfer

OTHER FEATURES

- ME+SIM phone book management
- Fixed Dialling Number
- SIM Toolkit Class 2
- SIM, network and service provider locks
- Real Time Clock
- Alarm management
- Software upgrade through Xmodem protocol
- UCS2 character set management

INTERFACES

- RS-232 and audio through mini sub-D 15-pin connector supporting:
  - Remote control by AT commands (GSM 07.07 and 07.05)
  - Baud rate from 300 to 115,200 bits/s
  - Autobauding (300 to 38,400 bits/s)
- Power supply through micro-FIT 4-pin connector
- SMA antenna connector
- Sliding SIM holder (3V/5V SIM interface)

DELIVERABLES

- User guide
- Power supply cable
- Y-cable for data and audio connection (optional)
- Fixing equipment

WAVECOM S.A. may, at any time and without notice, make changes or improvements to the products and services offered and/or cease producing or commercialising them.

EARLY BIRD RCS 384 740 643
WAVECOM, WISMO, Wireless and MUSE Platform are registered trademarks or filed trademarks of WAVECOM S.A. in France and/or other countries. © in a copyrighted material © copyright WAVECOM S.A. 04/02

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

wavecom®

WAVECOM S.A. - 12, boulevard Garibaldi - 92442 Issy-les-Moulineaux Cedex - France - Tel: +33 (0)1 46 29 08 00 - Fax: +33 (0)1 46 29 08 08
WAVECOM, Inc. - 4810 Eastgate Mall - Second Floor - San Diego, CA 92121 - USA - Tel: +1 858 362 0101 - Fax: +1 858 558 5485
WAVECOM Asia Pacific Ltd. - 5/F, Shui On Centre - 6/8 Harbour Road - Hong Kong, PRC - Tel: +852 2824 0254 - Fax: +852 2824 0255

www.wavecom.com