สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การควบคุมไร้สาย

WIRELESS CONTROL

T055668

โดย

นาย วีระชัย ปรีดาภัทรพงษ์

นาย ส่งศักดิ์ ดูการณ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมระบบควบคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

เลขหมู่.....
เลขทะเบียน..... 55668
วัน,เดือน,ปี 24 พ.ค. 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น และห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2546

สาขาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การควบคุมไร้สาย

Wireless Control

ผู้จัดทำ

1. นาย วีระชัย ปรีดาภัทรพงษ์ 44015303
2. นาย ส่งศักดิ์ ดูการณ์ 44015305

..... อาจารย์ที่ปรึกษา

(อาจารย์ ดาวร เบญจนราสุทธิ์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมไร้สาย

โดย
นาย วีระชัย ปรีดาภัทรพงษ์ 44015303
นาย ส่งศักดิ์ ดูการณ์ 44015305

อาจารย์ที่ปรึกษา
อาจารย์ ฉาวร เบญจนราสุทธิ์

ปีการศึกษา 2546

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้นำเสนอการประยุกต์ ใช้งาน WAP ผ่านทางอุปกรณ์สื่อสารไร้สาย ให้สามารถทำงานร่วมกับอุปกรณ์ Bluetooth ในลักษณะของไคลเอนต์ และเซิร์ฟเวอร์ โดยมีวัตถุประสงค์เพื่อใช้ในการควบคุมอุปกรณ์ที่อยู่ในระยะไกลแบบไร้สาย การทำงานจะอาศัยอุปกรณ์สื่อสารแบบไร้สายเรียกใช้โปรแกรมที่พัฒนาขึ้นบน WAP Page เพื่อทำการส่งข้อมูลจากอุปกรณ์สื่อสารไปยัง WAP Server ซึ่ง WAP Server จะส่งผ่านไปยังไคลเอนต์ โดยอาศัยอุปกรณ์ Bluetooth จากนั้นเครื่องไคล์เอนต์จะส่งข้อมูลผ่านพอร์ตอนุกรม ไปยังไมโครคอนโทรลเลอร์ เพื่อประมวลผล และควบคุมอุปกรณ์ฮาร์ดแวร์ปลายทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WIRELESS CONTROL

By

Mr. WEERACHAI PREEDAPATTARAPONG

Mr. SONGSAK DOOKAN

Advisor

TAWORN BENJANARASUTH

Academic Year : 2003

ABSTRACT

This thesis introduces WAP application for the wireless communication device to access Bluetooth device on the client and the server . The Objective is to wirelessly control certain devices in a long distance. By using the implemented application in the WAP page, the wireless communication device can send data to the WAP server . Consequently, the WAP Server will transfer the data to the specified client via Bluetooth . The client , then , will send the data through its serial port to a microcontroller for processing and controlling the desired hardware at the destination .

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ สำเร็จลุล่วงด้วยดี เพราะได้รับข้อเสนอแนะ และความช่วยเหลือจาก อาจารย์ ถาวร เบญจนราสุทธิ์ ที่คอยดูแลเอาใจใส่ แนะนำ ชี้แจง และแก้ไขข้อบกพร่องที่เกิดขึ้นกับโครงงานนี้ด้วยดีเสมอมา คณะผู้จัดทำขอกราบพระคุณเป็นอย่างสูง และขอบพระคุณคณาจารย์ภาควิชาวิศวกรรมระบบควบคุมทุกท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้ต่าง ๆ ให้แก่ผู้คณะจัดทำ

ผู้จัดทำขอขอบพระคุณเพื่อนๆ ทุกคนที่คอยช่วยเหลือ เสนอแนะ และให้กำลังใจในการทำปริญญานิพนธ์นี้จนสำเร็จลุล่วงด้วยดี

คณะผู้จัดทำ

นาย วีระชัย ปรีดาภัทรพงษ์

นายส่งศักดิ์ ดูการณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	i
กิตติกรรมประกาศ	iii
สารบัญ	iv
สารบัญรูปภาพ	vi
สารบัญตาราง	x
บทที่ 1 บทนำ	1
1.1 ความเป็นมาของโครงงาน	1
1.2 วัตถุประสงค์ในการดำเนินงาน	1
1.3 ขอบเขตของโครงงาน	1
บทที่ 2 จุดกำเนิดของ WAP เริ่มจากอินเทอร์เน็ตไร้สาย	3
2.1 TCP/ IP ระบบอินเตอร์เน็ต	3
2.2 TCP/IP เครือข่ายไร้สาย	8
2.3 WAP Gateway และโพรโตคอลใน WAP	9
บทที่ 3 Bluetooth	26
3.1 การก่อกำเนิดของ Bluetooth และรู้จักกับ Bluetooth SIG	27
3.1 จุดมุ่งหมายของการใช้Bluetooth	28
3.2 การทำงานของ Bluetooth เมื่อเทียบกับ OSI Model	32
3.3 มาตรฐาน IEEE 802.11	35
3.4 รูปแบบการเชื่อมต่อกายภาพของ Bluetooth	40
3.5 การสถาปนาการเชื่อมต่อกายภาพของ Bluetooth	47
บทที่ 4 การสื่อสารข้อมูลแบบอนุกรม	51
4.1 รูปแบบของการสื่อสารข้อมูลผ่านพอร์ตอนุกรม	51
4.2 มาตรฐานพอร์ตอนุกรมแบบ RS-232	56
4.3 ลักษณะการเชื่อมต่อคอนเน็คเตอร์ของพอร์ต RS-232	58
4.4 ลักษณะสัญญาณอินพุท และเอาท์พุตของพอร์ต RS-232	60
บทที่ 5 ไมโครคอนโทรลเลอร์	61
5.1 สถาปัตยกรรม และโครงสร้างของไมโครคอนโทรลเลอร์ MCS-51 แบบแพลช	63
5.2 การจัดขาของไมโครคอนโทรลเลอร์ MCS-51	64

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 โครงสร้าง และการทำงานของพอร์ต 67
5.4 หน่วยความจำของไมโครคอนโทรลเลอร์ MCS-51 แบบแพลช 71
5.5 พอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51 81
5.6 โหมดการทำงานของพอร์ตอนุกรมใน MCS-51 83
5.7 การเชื่อมต่อกับพอร์ตอนุกรมของคอมพิวเตอร์ 89
บทที่ 6 การติดตั้ง Apache เพื่อจำลอง WAP Server 91
6.1 ทำให้ Apache รู้จักกับ WAP 92
6.2 การติดตั้ง PHP ให้ทำงานร่วมกับ Apache Web Server 94
บทที่ 7 การออกแบบ และขั้นตอนการทำงาน 96
7.1 ส่วนประกอบของโครงงาน 96
7.2 การติดตั้ง และเตรียมโครงงาน 96
7.3 ขั้นตอนการทำงานของ Wireless Control 100
บทที่ 8 ผลการทดลอง และสรุป 107
8.1 การทดลองที่ 1 แสดงการติดต่อระหว่างเซิร์ฟเวอร์ กับไคลเอนต์ 107
8.2 การทดลองที่ 2 แสดงการส่งข้อมูลจาก WAP Browser ไปยัง WAP Server 109
8.3 การทดลองที่ 3 แสดงการส่งข้อมูลจาก WAP Browser ไปยังโปรแกรมไคลเอนต์ 111
8.4 ปัญหาที่เกิดขึ้น และการแก้ไข 116
8.5 แนวทางในการพัฒนาต่อไป 117
ภาคผนวก 118
บรรณานุกรม 143

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

รูปที่		หน้า
2.1	แสดงแบบจำลองชั้นสื่อสาร TCP/IP	4
2.2	โพรโตคอลที่เกี่ยวกับการร้องขอ และตอบกลับของเว็บเบราเซอร์กับเว็บเซิร์ฟเวอร์	7
2.3	แสดงการรับ-ส่งข้อมูลระหว่าง WAP Gateway กับเว็บเซิร์ฟเวอร์	8
2.4	แสดงขั้นตอนการร้องขอเอกสารในอินเตอร์เน็ตผ่าน Proxy server	9
2.5	ขั้นตอนการร้องขอเอกสารระบบ WAP และอินเตอร์เน็ตผ่าน WAP Gateway	10
2.6	โครงสร้างพื้นฐานของ WAP Gateway	12
2.7	ชั้นสื่อสารและโพรโตคอลใน WAP	17
2.8	ส่วนประกอบของ WAE	18
2.9	ระบบรักษาความปลอดภัยของข้อมูลใน WAP และอินเตอร์เน็ต	22
2.10	การเข้ารหัส และถอดรหัสเพื่อรักษาความปลอดภัยของข้อมูลทั้ง 2 ทิศทาง	23
2.11	แบบจำลองการทำงานของ WPKI ใน WAP 1.3	24
2.12	แสดงภาพรวมของโพรโตคอลที่เกี่ยวข้องในเครือข่ายไร้สาย และอินเทอร์เน็ต	25
3.1	แสดงเครือข่ายส่วนตัวโดยใช้อุปกรณ์ Bluetooth	26
3.2	แสดงการก่อร่างสร้างอาณาจักรการสื่อสารย่อยๆ ในบ้าน หรือที่ทำงาน	30
3.3	แสดงลักษณะ และตัวอย่างการใช้งาน Blutooth	30
3.4	แสดงลักษณะการเชื่อมต่ออุปกรณ์อื่นๆ ภายใต้ Bluetooth	32
3.5	แสดงการเปรียบเทียบระดับชั้นการทำงานของ Bluetooth กับ OSI Model	32
3.6	แสดงการเปรียบเทียบระดับชั้นโพรโตคอล Bluetooth กับ อุปกรณ์จริง	34
3.7	แสดงลักษณะการทำงานขั้นพื้นฐานของ Bluetooth	34
3.8	แสดงระดับชั้นการทำงานโปรโตคอลภายในมาตรฐานของ Bluetooth	40
3.9	แสดงลักษณะการเชื่อมต่อระหว่าง Master กับ Slave	41
3.10	แสดงลักษณะของ Time slot ที่ทำงานแบบ Master และ Slave	41
3.11	แสดงขนาด Time Slot สามารถมีขนาดใหญ่และเล็กได้ขึ้นอยู่กับรูปแบบการเชื่อมต่อทางกายภาพ	41
3.12	แสดงการทำงานของ Master กับ Slave	42
3.13	แสดงผังการเชื่อมต่อระหว่าง Master และ Slave บนความถี่ระดับClassต่างๆ	43
3.14	แสดงรูปแบบการเชื่อมต่อแบบจุดต่อจุด และแบบหนึ่งจุดกับหลายจุด	43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.15	แสดงรูปแบบการเชื่อมต่อระหว่างกลุ่มของ Slave และ Master	43
3.16	แสดงรูปแบบ Piconet ระหว่างเครื่อง Slave	44
3.17	แสดงลักษณะการเชื่อมต่อเครือข่าย Piconet	48
3.18	แสดงลักษณะการใช้ ARQ เพื่อควบคุมความผิดพลาดของข่าวสาร	50
4.1	ลักษณะสัญญาณของการสื่อสารแบบอะซิงโครนัส	51
4.2	แสดงรูปแบบของข้อมูลอนุกรมแบบอะซิงโครนัส	52
4.3	แสดงการเปลี่ยนข้อมูลแบบขนานให้เป็นข้อมูลแบบอนุกรม	53
4.4	แสดงการใช้บิตพาริตี้ตรวจสอบความผิดพลาดในการโอนย้ายข้อมูลแบบอนุกรม	56
4.5	รูปแบบการเชื่อมต่อขาพอร์ตอนุกรมมาตราฐาน RS-232 ของ DB-9 และ DB-25	58
4.6	การต่ออุปกรณ์ภายนอกกับพอร์ตอนุกรมของคอมพิวเตอร์ในลักษณะต่างๆ	59
4.6	(ก) แบบ Null Modem	59
4.6	(ข) แบบอาศัยสัญญาณ 3 เส้น	59
4.7	แสดงโครงสร้างทางฮาร์ดแวร์ของพอร์ตอนุกรม	60
5.1	โครงสร้างพื้นฐานของไมโครโปรเซสเซอร์	62
5.2	โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์	62
5.3	โครงสร้างของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชอนุกรมAT89Cxx	64
5.4	การจัดขามาตราฐานของไมโครคอนโทรลเลอร์ MCS-51	67
5.5	รายละเอียดโครงสร้างหลักของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช	68
5.6	แสดงวงจรภายในแต่ละพอร์ตของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช	69
5.7	วงจรพลูอัปภายในพอร์ตของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช	70
5.8	การจัดหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์เบอร์ MCS-51	72
5.9	การเชื่อมต่อหน่วยความจำโปรแกรมภายนอกของไมโครคอนโทรลเลอร์MCS-51	73
5.10	การเชื่อมต่อหน่วยความจำข้อมูลภายนอกของไมโครคอนโทรลเลอร์ MCS-51	74
5.11	การจัดสรรพื้นที่ของหน่วยความจำข้อมูลภายในไมโครคอนโทรลเลอร์MCS-51	75
5.12	เป็นการจัดสรรพื้นที่ จองหน่วยความจำข้อมูลภายในส่วนล่างของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช	75
5.13	แสดงโครงสร้างหน่วยความจำข้อมูลภายในส่วนบนของไมโครคอนโทรลเลอร์	76
5.14	การจัดสรรพื้นที่ของรีจิสเตอร์ฟังก์ชั่นพิเศษ(SFR)	77
5.15	แสดงสัญญาณการรับ และส่งข้อมูลในโหมด 0	84
5.16	แสดงสัญญาณการรับ และส่งข้อมูล 8 bit UART ในโหมด 1	85

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.17	แสดงสัญญาณการรับ และส่งข้อมูลของ 9 bit UART ในโหมด 2	86
5.18	แสดงรีจิสเตอร์การเลือกโหมดการทำงานของไทเมอร์ / เคาน์เตอร์ (TMOD)	86
5.19	การทำงานของไทเมอร์ / เคาน์เตอร์ ในโหมด 0	87
5.20	การทำงานของไทเมอร์ / เคาน์เตอร์ ในโหมด 1	87
5.21	การทำงานของไทเมอร์ / เคาน์เตอร์ ในโหมด 2	88
5.22	ไอซีแปลงสัญญาณเพื่อเชื่อมต่อกับพอร์ตอนุกรมของคอมพิวเตอร์	89
5.23	วงจรเชื่อมต่อ MAX 232 เข้ากับพอร์ตอนุกรมของคอมพิวเตอร์ และไมโครคอนโทรล เลอร์ MCS-51	90
6.1	เป็นการทดสอบโปรแกรม Apache ว่าสามารถทำงานได้โดย Internet Explorer	92
6.2	แสดงผลของสคริปต์ PHP ที่ทำงานร่วมกับ Apache	95
7.1	โฟล์วชาร์ตแสดงการติดตั้ง และเตรียมโครงงาน	97
7.2	แสดงการรับ - ส่งไฟล์ข้อมูลระหว่างกัน	98
7.3	แสดงผลที่ได้จากการจำลองของ WAP Server	99
7.4	แสดงผลที่ได้จากการใช้โทรศัพท์เข้ามาที่ WAP Server	99
7.5	แสดงการ Access WAP จากโทรศัพท์มือถือเข้ามาที่เครื่อง WAP Server	100
7.6	แสดงหน้าตาการ Login และการป้อนรหัสผ่านโทรศัพท์มือถือ	100
7.7 (ก)	เมื่อกรอก Password ถูกต้องจะแสดงหน้าตาเข้าสู่ Control Page	101
7.7 (ข)	เมื่อกรอก Password ผิดจะแสดงหน้าตาการร้องเตือน เพื่อให้กรอกใหม่	101
7.8	แสดงการติดต่อกับเครื่องคอมพิวเตอร์ที่ต้องการควบคุม	101
7.9	แสดงการเลือก Engine ในเครื่องคอมพิวเตอร์ที่ต้องการควบคุม และยังสามารถเลือกใช้ Engine ตัวใด ๆ ON – OFF ได้	102
7.10	แสดงการยืนยัน Engine ที่ได้เลือกไว้อีกครั้งหนึ่ง	102
7.11	แสดง Result Page ซึ่งยืนยันว่า WAP ได้ปฏิบัติตามคำสั่งนี้เรียบร้อยแล้ว	103
7.12	แสดงการทำงานของโปรแกรมเซิร์ฟเวอร์	103
7.13	โปรแกรมไคลเอนต์อ่านข้อมูลที่ส่งมา และนำมาถอดรหัส	104
7.14	แสดงการถอดรหัสข้อมูลที่ส่งมาจากโปรแกรมของไคลเอนต์	104
7.15	โฟลว์ชาร์ตแสดงการทำงานของ Wireless Control	105
8.1	แสดงการส่งข้อมูลจากโปรแกรมของเซิร์ฟเวอร์ ไปยังโปรแกรมของไคลเอนต์	107
8.2	โฟลว์ชาร์ตแสดงการทำงานของโปรแกรมเซิร์ฟเวอร์กับโปรแกรมไคลเอนต์	108

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.3 แสดงการส่งคำสั่งจาก WAP Browser กับ WAP Server 109
8.4 โฟลว์ชาร์ตแสดงการส่งคำสั่งจากโทรศัพท์มือถือกับ WAP Server 110
8.5 โฟลว์ชาร์ตแสดงการทำงานของ WAP Browser กับ WAP Server 110
8.6 แสดงการส่งคำสั่งจาก WAP Browser กับไคลเอนต์ 112
8.7 โฟลว์ชาร์ตแสดงการส่งข้อมูลจาก WAP Browser ไปยังโปรแกรมไคลเอนต์ 113
8.8 โฟลว์ชาร์ตแสดงการทำงานของ MCS-51 114

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

		หน้า
ตารางที่ 3.1	แสดงลักษณะของ Chip ที่เป็นรูปแบบของสัญญาณที่แทนด้วยเลข binary	36
ตารางที่ 3.2	แสดงเปรียบเทียบการทำงานระหว่าง Bluetooth กับ IEEE 802.11	45
ตารางที่ 3.3	แสดงความแตกต่างระหว่าง Bluetooth กับผลิตภัณฑ์อื่นๆ	46
ตารางที่ 4.1	อัตราถ่ายทอดข้อมูล, ช่วงเวลาของแต่ละบิต และอัตราถ่ายทอดข้อมูลที่ใช้ในการย้ายข้อมูลอนุกรม	54
ตารางที่ 4.2	แสดงบิตพาริตี้ของข้อมูลพอร์ตอนุกรม	55
ตารางที่ 5.1	เปรียบเทียบคุณสมบัติของไมโครคอนโทรลเลอร์ และไมโครโปรเซสเซอร์	63
ตารางที่ 5.2	แสดงการเลือกแบงก์ของหน่วยความจำส่วนล่างเพื่อติดต่อรีจิสเตอร์แบงก์ R0-R7	79
ตารางที่ 5.3	แสดงการกำหนดโหมดทำงานของพอร์ตอนุกรม ด้วย บิต SM0 และSM1	82
ตารางที่ 5.4	แสดงการควบคุมการใช้งานไทเมอร์ 1	86
ตารางที่ 8.1	แสดงระยะทาง และเวลาที่เซิร์ฟเวอร์ติดต่อกับไคลเอนต์ผ่าน Bluetooth	107
ตารางที่ 8.2	แสดงระยะทาง และเวลาการส่งข้อมูลจาก WAP Browser ไปยัง WAP Server	109
ตารางที่ 8.3	แสดงระยะทาง และการส่งข้อมูลจาก WAP Browser ไปยังโปรแกรมไคลเอนต์	111
ตารางที่ 8.4	แสดงการถอดรหัสของ MCS-51 ที่รับข้อมูลจากพอร์ตอนุกรมของเครื่องไคลเอนต์มาประมวลผล	115

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

ในส่วนของบทนำจะเป็นการเป็นกล่าวถึงที่มาโครงงานที่ประกอบด้วย ความเป็นมาของโครงงาน วัตถุประสงค์ในการดำเนินงาน และขอบเขตของโครงงาน ซึ่งเป็นเหตุผลที่ทำให้คณะผู้จัดทำริเริ่มที่ทำโครงงานนี้ขึ้นมา โดยมีวัตถุประสงค์เพื่อศึกษา เรียนรู้ และเป็นการนำอุปกรณ์สื่อสารไร้สายมาประยุกต์ใช้ในงานควบคุม

1.1 ความเป็นมาของโครงงาน

ปัจจุบันการติดต่อสื่อสารมีความสะดวกและแม่นยำมาก ทำให้การส่งข้อมูลจากจุดหนึ่งไปยังอีกจุดหนึ่งได้อย่างสมบูรณ์ และรวดเร็ว เนื่องจากความก้าวหน้าของเทคโนโลยีได้เชื่อมต่อทั่วทุกมุมโลกเข้าด้วยกัน ในระดับที่ผู้ใช้ Computer ทั่วไปสามารถเข้าถึงได้โดยง่าย

คณะผู้จัดทำจึงคิดที่จะทำโครงงานนี้ เพื่อให้สามารถเข้าถึงการควบคุมอุปกรณ์ในระยะไกลได้อย่างสะดวกสบายขึ้น โดยอาศัยโครงข่ายโทรคมนาคมขั้นพื้นฐานที่มีให้บริการอยู่ทั่วไปมาประยุกต์ให้เหมาะสมกับการควบคุมทางฮาร์ดแวร์ ซึ่งต้องใช้โปรแกรมบางส่วนที่พัฒนาขึ้นเองทั้งฝั่งของโทรศัพท์มือถือ และคอมพิวเตอร์ เพื่อให้การทำงานสอดคล้องกัน และนำมาซึ่งผลการควบคุมฮาร์ดแวร์ที่แม่นยำ

1.2 วัตถุประสงค์ในการดำเนินงาน

1. ศึกษาอุปกรณ์สื่อสารที่ใช้ เพื่อนำมาประยุกต์ใช้ในการควบคุม
2. ศึกษาการเขียน WAP SITE และโปรแกรมที่พัฒนาโดย Borland Delphi
3. ศึกษาข้อมูลเกี่ยวกับการติดต่อ และส่งข้อมูลผ่านพอร์ตอนุกรม เพื่อนำไปประยุกต์ใช้งานกับไมโครคอนโทรลเลอร์
4. ต้องสามารถทำควบคุมอุปกรณ์ฮาร์ดแวร์ที่อยู่ในระยะไกลได้อย่างแม่นยำ

1.3 ขอบเขตของโครงงาน

1. ค้นคว้า และศึกษาข้อมูลเกี่ยวกับอุปกรณ์ และเทคโนโลยีสื่อสารไร้สาย
2. ออกแบบระบบการทำงานเพื่อให้สามารถทำงานร่วมกับอุปกรณ์ และเทคโนโลยีสื่อสารไร้สายที่คัดเลือกมา ได้อย่างสอดคล้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เขียน WAP SITE และแทรกโปรแกรม WML Script เพื่อประมวลผลร่วม กับ WAP Server ที่ปลายทาง
4. เขียนโปรแกรมที่พัฒนาโดย Borland Delphi เพื่อเชื่อมต่อการรับ-ส่งข้อมูลจาก WAP Server โดยอาศัย Protocal TCP/IP บน Profiles ของ Bluetooth และควบคุมฮาร์ดแวร์ทางพอร์ตอนุกรม
5. สร้างวงจรควบคุมด้วยไมโครคอนโทรลเลอร์ MCS-51 โดยเชื่อมต่อเข้ากับพอร์ตอนุกรมของคอมพิวเตอร์เครื่องไคลเอนต์
6. สามารถนำโครงงานนี้มาใช้ในการควบคุม เปิด-ปิด อุปกรณ์ที่อยู่ในระยะไกลได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
จุดกำเนิดของ WAP เริ่มจากอินเทอร์เน็ตไร้สาย

WAP เป็นรูปแบบหนึ่งของเทคโนโลยีเครือข่ายไร้สาย หรือที่เรียกว่า**"Mobile Internet"** นั้นได้แนวความคิดในเทคโนโลยีที่เกี่ยวกับอินเทอร์เน็ตแบบเดิมมาดัดแปลง เพื่อจะทำให้เหมาะกับลักษณะของการทำงานและการใช้งานในเครือข่ายไร้สาย ในอินเทอร์เน็ตมีโพรโตคอลต่างๆ เช่น HTTP, TCP/IP ฯลฯ มีภาษา HTML เพื่อใช้ในการเขียนเว็บเพจ และอื่นๆอีก ส่วนใน WAP จะมีโพรโตคอลของตัวเองเหมือนกัน และภาษา WML ซึ่งเปรียบเสมือนเป็นภาษา HTML ในโลกไร้สาย

โพรโตคอล คือ ข้อกำหนด หรือกฎกติกาเกี่ยวกับการสื่อสารข้อมูลในระบบคอมพิวเตอร์ ทำไมต้องมีโพรโตคอลเนื่องจากระบบของคอมพิวเตอร์มีอยู่หลายระบบ แต่ละระบบระเบียบการทำงานที่แตกต่างกัน ถ้าจะให้ระบบที่แตกต่างกันติดต่อสื่อสารพูดคุยกันรู้เรื่องก็จะต้องมีข้อกำหนดหรือกฎกติการะหว่างกัน ดังนั้นโพรโตคอลจึงเป็นสิ่งสำคัญมากในการสื่อสารข้อมูล

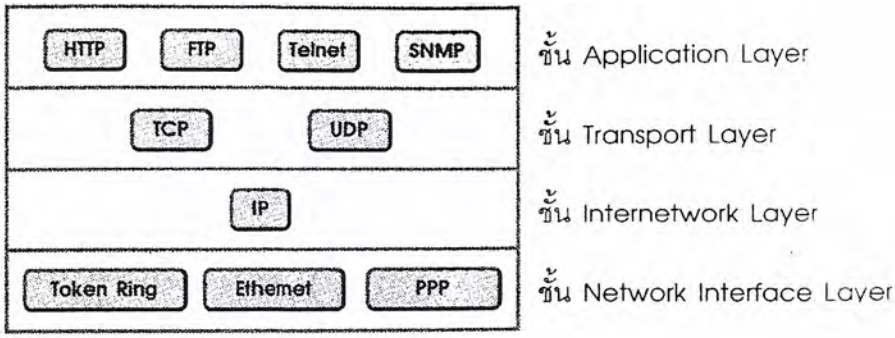
การสื่อสารข้อมูลในระบบอินเทอร์เน็ต ตัวอย่างของโพรโตคอลจริงๆ ในการรับ-การส่งข้อมูล เช่น ต้องระบุที่อยู่ปลายทางของเครื่องรับไว้กับข้อมูลที่ส่งไปด้วย, ต้องแบ่งข้อมูลให้ออกเป็นส่วนย่อยๆ ตามขนาดที่กำหนด, ต้องมีการตรวจสอบว่าข้อมูลไปถึงผู้รับปลอดภัยหรือไม่, ต้องมีการระบุชนิดของข้อมูลที่จะส่งเพื่อทำให้ทราบว่าเป็นข้อมูลแบบธรรมดา หรือรูปภาพ เป็นต้น โดยที่โพรโตคอลของอินเทอร์เน็ตประกอบไปด้วย

2.1 TCP/IP ของระบบอินเทอร์เน็ต

โพรโตคอลในอินเทอร์เน็ตที่รู้จักกันดี ได้แก่ โพรโตคอล HTTP และ TCP/IP อย่างแน่นอน ในอดีตก่อนที่จะเริ่มมีการใช้งานอินเทอร์เน็ตกันอย่างแพร่หลายนั้นกระทรวงกลาโหมของสหรัฐอเมริกาได้พัฒนาระบบเครือข่ายในอินเทอร์เน็ตที่เรียกว่า **"ARPANET"** เพื่อใช้ในการเชื่อมต่อองค์กรภายในประเทศ ต่อมาระบบนี้ได้พัฒนากฎการสื่อสารแบบ TCP/IP ขึ้นมา และได้แพร่หลายไปทั่วโลกจนเป็นแกนหลักของระบบอินเทอร์เน็ตในปัจจุบัน

กฎการสื่อสารแบบ TCP/IP ได้กำหนดขั้นตอนในการสื่อสารระหว่างคอมพิวเตอร์ออกเป็น 4 ขั้นตอนหลักๆ โดยแบ่งเป็นชั้นๆ และแต่ละชั้นก็มีโพรโตคอลที่เกี่ยวข้องในขั้นตอนการสื่อสารนั้นๆ อยู่ด้วย ดังรูปที่ 2.1 ซึ่งเรียกว่า **"แบบจำลองชั้นสื่อสาร TCP/IP"**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 แสดงแบบจำลองชั้นสื่อสาร TCP/IP

ในการรับ-ส่งข้อมูลทางระบบอินเทอร์เน็ตนั้น โพรโตคอลในแต่ละชั้นจะทำหน้าที่ของตนซึ่งแตกต่างกันไป เพื่อให้การรับส่งข้อมูลเป็นไปอย่างสมบูรณ์ ดังนั้นจึงต้องมีการแบ่งขั้นตอนการสื่อสารออกเป็นชั้นๆ (layer) ด้วย เหตุผลก็คือ ต้องการลดความซับซ้อนในการสื่อสารข้อมูลภายในชั้นสื่อสารแต่ละชั้นซึ่งต่างก็มีหน้าที่ของตน และเป็นอิสระจากชั้นอื่นแต่ละชั้นจะติดต่อกันก็จะมีช่องทางที่แน่นอน โดยชั้นอื่นๆไม่ต้องรู้ว่าภายในชั้นที่จะติดต่อด้วยมีขั้นตอนการทำงานอย่างไรจึงถือไม่เป็นการก้าวก่ายงานกัน โดยทั่วไปชั้นสื่อสารข้างล่างจะให้บริการชั้นที่อยู่เหนือขึ้นไป

ย้อนกลับไปที่ TCP/IP จะเห็นได้ว่าแท้จริงแล้ว TCP และ IP เป็นโพรโตคอลคนละตัวกัน แต่เนื่องจากเวลาใช้งานจะต้องสนับสนุนซึ่งกันและกัน โพรโตคอล TCP จะทำงานโดยลำพังไม่ได้ ต้องอาศัยบริการจากโพรโตคอล IP ที่อยู่ชั้นต่ำกว่า ดังนั้นจึงเรียกรวมกันว่า "**TCP/IP**" จากภาพที่จะเห็นได้ว่าแบบจำลองชั้นสื่อสาร TCP/IP ประกอบด้วยชั้น Application Layer ชั้น Transport Layer ชั้น Internetwork Layer และชั้น Network Interface Layer

ชั้น Application Layer

หรือ เรียกอีกอย่างหนึ่งว่า "**Process Layer**" ในชั้นนี้มีโพรโตคอลที่เกี่ยวข้องกับการใช้งานต่างๆซึ่งพบเห็นกันบ่อย เช่น

- โพรโตคอล HTTP เป็นข้อกำหนดที่เกี่ยวข้องกับการรับ- ส่งข้อมูลไฟล์เพื่อเรียกดูข้อมูลในระบบ WWW โดยเฉพาะอย่างยิ่งระหว่างเว็บเบราเซอร์ (ไคล์เอนต์) กับเว็บเซิร์ฟเวอร์
- โพรโตคอล FTP เป็นข้อกำหนดเกี่ยวกับการถ่ายโอนข้อมูลจากเว็บเซิร์ฟเวอร์มายังเครื่องลูกข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- โพรโตคอล Telnet เป็นข้อกำหนดที่เกี่ยวกับการใช้งานของเซิร์ฟเวอร์ซึ่งผ่านทางคอมพิวเตอร์ที่เป็นไคลเอนต์ในระยะไกล โดยอาศัยโปรแกรม Telnet

โพรโตคอลต่าง ๆ ในชั้นนี้จะให้บริการแก่โปรแกรมใช้งานหรือแอพพลิเคชั่นต่าง ๆ เช่น HTTP ให้บริการแก่เว็บเบราเซอร์ เช่น Netscape หรือ Internet Explorer, FTP ให้บริการแก่โปรแกรมถ่ายโอนไฟล์ เช่น Cute FTP หรือ WS_FTP

การเรียกดูข้อมูลเว็บไซต์ซึ่งเป็นการทำงานของโปรแกรมเบราเซอร์ภายใต้ข้อกำหนดของโพรโตคอล HTTP ก็อธิบายได้ว่าเว็บเบราเซอร์จะขอให้โพรโตคอล HTTP กำหนดการติดต่อสื่อสารกับทางฝั่งเว็บเซิร์ฟเวอร์ ส่วนทางฝั่งเว็บเซิร์ฟเวอร์นั้นเมื่อส่งเอกสาร HTML มายังเว็บเบราเซอร์ก็ต้องอาศัยโพรโตคอล HTTP ด้วย ดังนั้นจึงสังเกตได้ว่าเวลาระบุ URL ที่เว็บเบราเซอร์ก็ต้องระบุในลักษณะ http://www.bluetooth.com คือ จะมี HTTP นำหน้าแสดงให้เห็นว่า การติดต่อสื่อสารระหว่างเว็บเซอร์ และเว็บเซอร์เวอร์ เพื่อดูเอกสารจากเว็บไซต์ ยึดถือ HTTP เป็นหลักนั่น แต่ทั้งนี้ก็ต้องอาศัยความสามารถของ TCP เพื่อควบคุมการติดต่อสื่อสารกันด้วย

ชั้น Transport Layer

ในชั้นนี้จะมีโพรโตคอลที่เกี่ยวข้องอยู่ 2 ตัว คือ TCP และ UDP ซึ่งทำหน้าที่ควบคุมการรับ-ส่งข้อมูลระหว่างฝั่งส่งและฝั่งรับ เช่น ควบคุมความเร็วในการรับส่งข้อมูลเพื่อไม่ให้เกิดกรณีที่ฝ่ายส่ง ส่งข้อมูลเร็วจนกระทั่งฝ่ายรับไม่สามารถรับทัน เป็นต้น สามารถจะอธิบายรายละเอียดของโพรโตคอล TCP และ UDP ได้ดังนี้

- TCP (Transmission Control Protocol)

โพรโตคอลนี้ใช้ในการควบคุมการรับส่งข้อมูลระหว่างฝั่งรับและฝั่งส่ง โดยการติดต่อระหว่างทั้ง 2 ฝั่ง จำเป็นต้องสร้างการเชื่อมต่อ (Connection) ก่อน และต้องให้ฝ่ายรับพร้อมที่จะรับข้อมูลก่อนเท่านั้น ถึงจะเริ่มรับส่งข้อมูลกันได้ อีกทั้งสถานะการเชื่อมต่อกันนี้จะคงอยู่จนกว่าฝ่ายใดฝ่ายหนึ่งยกเลิกการติดต่อ เราเรียกการติดต่อแบบนี้ว่า **"Connection - Oriented"**

โพรโตคอลของ TCP มีเส้นทางการติดต่อแล้ว โพรโตคอลนี้ก็จะแบ่งข้อมูลออกเป็นหน่วยย่อย ๆ เรียกว่า **"แพคเก็ต"** (packet) แล้วส่งไปให้แก่โพรโตคอลในชั้น Internetwork Layer เพื่อจัดส่งต่อไป ส่วนกรณีของฝ่ายรับโพรโตคอล TCP จะรับข้อมูลจากชั้น Internetwork Layer ในรูปของแพคเก็ตจากนั้นจึงประกอบกลับคืนเป็นข้อมูลและนำส่งขึ้นไปให้แก่โพรโตคอลในชั้น Application Layer ของฝ่ายรับเพื่อจัดการต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับสาเหตุที่ต้องแบ่งข้อมูลเป็นหน่วยย่อย ๆ ก็เนื่องจากว่าข้อมูลที่มีขนาดเล็กย่อมมีความคล่องตัวในการส่งผ่านเครือข่ายมากกว่า และในกรณีที่ข้อมูลเกิดความเสียหาย ฝ่ายส่งจะต้องส่งแพคเก็ตข้อมูลที่เสียหายไปยังฝ่ายรับใหม่ ก็สามารถจะทำได้ง่ายกว่าข้อมูลที่ใหญ่กว่าและมีความน่าเชื่อถือ เนื่องจากโพรโตรคอล TCP มีการตรวจสอบข้อมูลที่ส่งไปว่าถึงมือผู้รับอย่างถูกต้องหรือไม่ ถ้าไม่ถูกต้องก็จะส่งข้อมูลไปให้ผู้ใช้ใหม่ซึ่งจัดว่าเป็นข้อดี แต่ในขณะเดียวกันก็เป็นข้อเสียด้วยเพราะทำให้เสียเวลาและเปลืองทรัพยากรของระบบ

- UDP (User Datagram Protocol)

โพรโตคอลนี้มีหน้าที่ควบคุมความต่อเนื่องของการติดต่อสื่อสารข้อมูล แต่วิธีนี้การติดต่อไม่เหมือนกับโพรโตคอล TCP ซึ่งจะเรียกวิธีนี้ว่า **"Connectionless"** ไม่มีการสร้างการเชื่อมต่อระหว่างผู้รับกับผู้ส่งก่อน ฝ่ายรับไม่จำเป็นต้องรับรู้ว่าจะมีข้อมูลส่งมาและไม่มีการตรวจสอบว่าแพคเก็ตข้อมูลที่ส่งมาว่าสูญหาย หรือเสียหายซึ่งในแง่ของความน่าเชื่อถือแล้ววิธีนี้มีข้อด้อยมากกว่าวิธี Connection-Oriented แต่มีข้อดีตรงที่ไม่เปลืองทรัพยากรของระบบที่ต้องมาคอยตรวจสอบ และส่งข้อมูลใหม่จึงเหมาะส่งข้อมูลจำพวกมัลติมีเดียในลักษณะ streaming เช่น เพลง ภาพแอนนิเมชั่น ฯลฯ

โพรโตคอลในชั้น Transport Layer ได้ออกแบบให้มีการติดต่อ 2 แบบ คือ Connection – Oriented ใน TCP และ Connectionless ใน UDP ซึ่งสามารถเลือกรูปแบบให้เหมาะสมกับแต่ละงาน แต่ก็ต้องอาศัยความสามารถในส่วนของโพรโตคอลในชั้น Internetwork Layer เพื่อหาเส้นทางที่เหมาะสมที่สุดในการส่งข้อมูล

ชั้น Internetwork Layer

เป็นชั้นสื่อสารที่มีโพรโตคอลหลัก คือ IP (Internet Protocol) ทำหน้าที่ในการค้นหาเส้นทางที่เหมาะสมกับการส่งข้อมูลแต่ละหน่วยย่อย ๆ (แพคเก็ต) การทำงานของโพรโตคอล IP ก็จะเป็นแบบ Packet – Switching คือ แต่ละแพคเก็ตจะเดินทางจากจุดเริ่มต้นผ่านไปยังเครือข่ายต่างๆในระบบอินเทอร์เน็ตได้อย่างอิสระจนถึงจุดหมายปลายทาง โดยไม่จำเป็นต้องไปเส้นทางเดียวกันเสมอ และลำดับของแพคเก็ตที่ไปถึงปลายทางก็ไม่สำคัญแพคเก็ตแรกที่ส่งไปอาจมาถึงปลายทางที่หลังแพคเก็ตอื่น ๆ ก็ได้ เส้นทางที่แต่ละแพคเก็ตผ่านไปก็ไม่แน่นอนกล่าว คือ หากส่งข้อมูลชุดเดิมอีกครั้งแพคเก็ตเดิมอาจจะได้อีกเส้นทางหนึ่งซึ่งตรงกันข้ามกับครั้งแรก ดังนั้นโพรโตคอล IP จะทำงานแบบ Connectionless เพราะไม่มีการระบุเส้นทางตายตัวจากฝ่ายรับไปยังฝ่ายส่ง

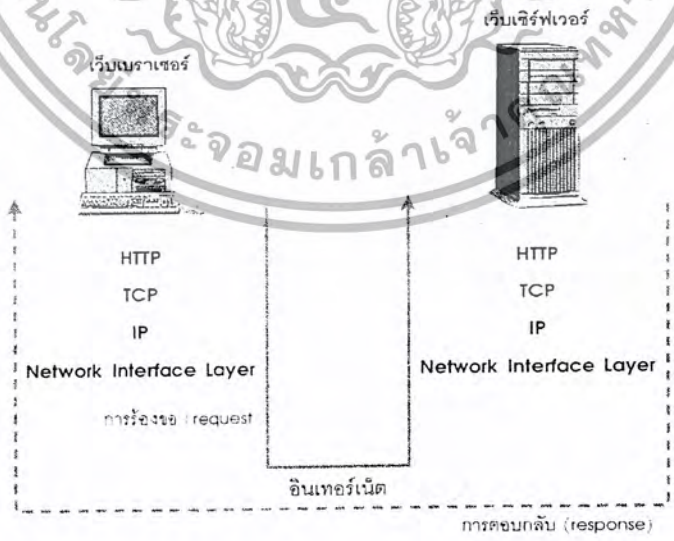
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชั้น Network Interface Layer

เป็นชั้นที่รับข้อมูลมาจากชั้น Internetwork Layer แล้วปรับแพคเก็ตให้อยู่ในรูปที่เหมาะสมกับการส่งไปในเครือข่าย หรือสายสื่อสาร เช่น ปรับเป็นสัญญาณทางไฟฟ้าเพื่อส่งไปในสายสื่อสาร เป็นต้น และในทางกลับกันหากเป็นฝ่ายรับก็จะรับข้อมูลมาจากสายสื่อสารแล้วปรับรูปแบบของข้อมูลเพื่อส่งขึ้นไปยังชั้น Internetwork Layer ต่อไป ในเมื่อการส่งข้อมูลที่อาศัยโพรโตคอล TCP ควบคุมการส่งข้อมูลซึ่งทำงานแบบ Connection – Oriented และต้องอาศัยโพรโตคอล IP ในการหาเส้นทางส่งข้อมูลแพคเก็ตแบบ Connectionless แล้วรูปแบบของข้อมูลจะเป็นอย่างไร ต้องรู้ว่าโพรโตคอลของ TCP กับ IP อยู่คนละชั้นกันและมีหน้าที่แต่ต่างกัน ดังนั้นจึงต้องรู้ว่าข้อมูลอยู่ในชั้นอะไร

จึงสรุปความสัมพันธ์ระหว่างโพรโตคอล HTTP กับ โพรโตคอล TCP /IP ว่าโพรโตคอล HTTP เป็นข้อกำหนดในการแลกเปลี่ยน หรือรับ- ส่งเอกสารในระบบเว็บเบราเซอร์ / เว็บเซิร์ฟเวอร์ ซึ่งต้องอาศัยโพรโตคอล TCP ในการควบคุมการรับ-ส่งข้อมูล เช่น ควบคุมความเร็วในการส่งข้อมูล ควบคุมความถูกต้องของข้อมูล และแบ่งข้อมูลออกเป็นแพคเก็ตเล็ก ๆ ส่วนโพรโตคอล IP ก็ช่วยในการหาเส้นทางสำหรับส่งแพคเก็ตข้อมูลเหล่านั้น

ในการทำงานระหว่างเว็บเซิร์ฟเวอร์กับเว็บเบราเซอร์ในระบบ WWW ซึ่งจะเกี่ยวข้องกับโพรโตคอลต่าง ๆ เป็นไปดังภาพ 2.2 โดยที่การร้องขอ (request) จากไคลเอนต์ไปยังเว็บเซิร์ฟเวอร์จะแทนด้วยเส้นทึบ และการตอบกลับ (response) จะแทนด้วยเส้นประ

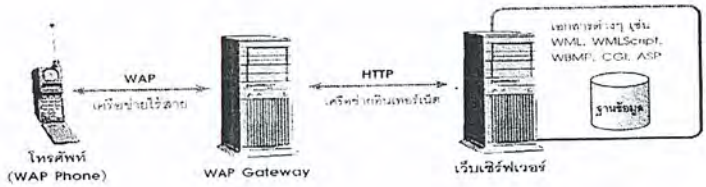


รูปที่ 2.2 โพรโตคอลที่เกี่ยวกับการร้องขอ และตอบกลับของเว็บเบราเซอร์กับเว็บเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 TCP/IP กับเครือข่ายไร้สาย

โพรโตคอลของ TCP/IP เป็นแกนหลักของระบบอินเทอร์เน็ตนั้นความจริงได้มีผู้นำไปทดลองใช้ในเครือข่ายไร้สาย หรือเครือข่ายโทรศัพท์เคลื่อนที่มาแล้ว แต่ด้วยข้อจำกัดในเรื่องสภาพความไม่เสถียรของการเชื่อมต่อจึงส่งผลให้คลื่นสัญญาณขาดหายไปบ้างตามสภาพแวดล้อมซึ่งมีผลต่อประสิทธิภาพ โดยรวมของโพรโตคอล TCP สาเหตุเนื่องจากถ้ามีการขาดหายของสัญญาณจะทำให้การสื่อสารข้อมูลไม่มีความต่อเนื่อง โพรโตคอล TCP จะตีความไปว่าขณะมีการส่งข้อมูลอย่างแออัดคับคั่งในเครือข่ายจึงพยายามลดอัตราการส่งข้อมูลลง และเมื่อคลื่นสัญญาณกลับมาดีดังเดิม โพรโตคอล TCP จะเพิ่มอัตราการส่งข้อมูลให้ใกล้เคียงกับอัตราเดิมแต่วิธีจะกินเวลานาน เมื่อเทียบกับเครือข่ายอินเทอร์เน็ตแบบปกติการส่งข้อมูลจะต่อเนื่องอัตราการส่งข้อมูลจะคงที่ ประสิทธิภาพของโพรโตคอล TCP จะดีกว่า นอกจากข้อมูลทางด้านเสถียรภาพแล้วยังมีข้อจำกัดในทางด้านของ bandwidth ในเครือข่าย (หรือเรียกตามศัพท์เทคนิคว่า **"latency"**) ซึ่งนับว่ามากเกินไป ส่งผลให้ความเร็วในการรับส่งข้อมูลจำกัดอยู่เพียง 9.6 kbps เท่านั้น (หากเปรียบเทียบกับความเร็วโมเด็มในยุคแรกซึ่งอยู่ที่ 9.6 – 1.44 kbps ก็ถือว่าน้อยอยู่ดี) และด้วยข้อจำกัดทั้งหลายเหล่านี้เอง ล้วนเป็นเหตุให้การพัฒนาการใช้งานอินเทอร์เน็ตไร้สายด้วยโพรโตคอล TCP/IP ไม่ประสบความสำเร็จอย่างสิ้นเชิงจึงได้เกิดแนวคิดว่าเมื่อข้อมูลในอินเทอร์เน็ตไม่สามารถรับ-ส่งข้อมูลในเครือข่ายไร้สายอันเนื่องจากโพรโตคอล TCP ทำงานไม่ดีในเครือข่ายไร้สาย อุปกรณ์โทรศัพท์มือถือยังมีพลังความสามารถไม่พอที่ประมวลผลข้อมูลในอินเทอร์เน็ต ทางออกก็คือ เมื่อต้องการส่งข้อมูลจากเว็บเซิร์ฟเวอร์ไปยังโทรศัพท์มือถือ โดยที่ข้อมูลยังคงส่งผ่านเครือข่ายอินเทอร์เน็ตที่ใช้โพรโตคอล TCP/IP เป็นโพรโตคอลหลัก มาให้แก่ **"ตัวกลาง"** ซึ่งทำหน้าที่แปลงข้อมูลไปเป็นอีกรูปแบบหนึ่งก่อนที่จะส่งต่อไปยังโทรศัพท์มือถือ เพื่อให้ข้อมูลอยู่ในลักษณะที่เหมาะสมกับการรับ-ส่งผ่านเครือข่ายไร้สายและความสามารถในการประมวลผลของโทรศัพท์มือถือ โดยตัวกลางที่ทำหน้าที่แปลงข้อมูล คือ **"WAP Gateway"**



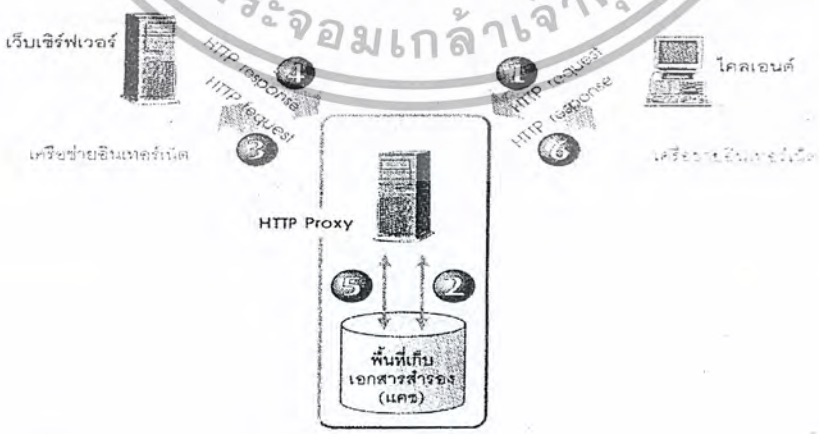
รูปที่ 2.3 แสดงการรับ-ส่งข้อมูลระหว่าง WAP Gateway กับเว็บเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 2.3 แสดงการรับ-ส่งข้อมูลระหว่าง WAP Gateway กับเว็บเซิร์ฟเวอร์ (หรือแอปพลิเคชั่นเซิร์ฟเวอร์) เป็นไปตามมาตราฐานต่าง ๆ ของ WWW ปกติ ได้แก่ โพรโตคอลของ TCP/IP และมาตราฐานของโพรโตคอล HTTP 1.1 ส่วนทางฝั่ง WAP Gateway กับโทรศัพท์มือถือจะเป็นไปตามมาตราฐานชุดโพรโตคอลที่ WAP Forum พัฒนาขึ้น โดยข้อมูลที่รับ- ส่งในฝั่งนี้จะเป็นข้อมูลแบบ binary ที่ผ่านการบีบอัดที่เล็กลงแล้วซึ่งความแตกต่างระหว่างมาตราฐานทั้งสองฝั่งเป็นหน้าที่ของ WAP Gateway ที่ต้องแปลงไปมาเพื่อให้เหมาะสม

2.3 WAP Gateway และโพรโตคอลใน WAP

WAP Gateway มีแนวความคิดมาจากเว็บทุกอย่าง ในเทคโนโลยี WAP ถูกดัดแปลงมาจากแนวความคิดของเว็บ และปรับปรุงให้เหมาะสมกับการสื่อสารในเครือข่ายไร้สายเรื่อง WAP Gateway ถ้าจะกล่าวถึงขั้นตอนในการทำงานบางอย่างของ WAP Gateway มีความคล้ายคลึงกับ proxy server ในระบบอินเทอร์เน็ตซึ่ง **"Proxy server"** เป็นอุปกรณ์ที่มีประโยชน์มากช่วยในการลดปริมาณการจราจรในเครือข่าย เนื่องจากหน้าที่ Proxy server คือ เก็บเอกสารหรือเว็บเพจไว้ในพื้นที่เก็บข้อมูลหรือเอกสารสำรองไว้กับตนเอง ที่เรียกว่า " แคช" (cache) เมื่อมีผู้ใช้เรียกดูเอกสารใน WWW โดยการระบุ URL ที่เว็บเบราเซอร์ โปรแกรมเบราเซอร์นั้นก็จะส่งสัญญาณร้องขอ (request) มาที่ Proxy server ก่อน แล้วก็จะตรวจสอบมีไฟล์เอกสารหรือเว็บเพจนั้นมีแคชหรือไม่ หากพบว่าเอกสารหรือเว็บเพจนั้นมีอยู่ในแคชอยู่แล้ว และเป็นข้อมูลล่าสุด (update) ก็จะส่งข้อมูลนั้นไปให้แก่ผู้ใช้ โดยไม่ต้องเสียเวลาร้องขอเอกสารจริงๆ จากเว็บเซิร์ฟเวอร์อีกที การทำงานของ proxy server สามารถจำลองมาให้ดูดังรูปที่ 2.4 ซึ่งมีขั้นตอนดังนี้

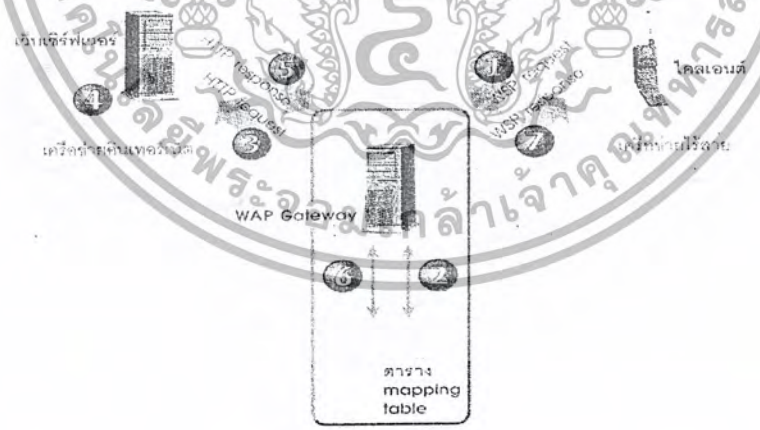


รูปที่ 2.4 แสดงขั้นตอนการร้องขอเอกสารในอินเตอร์เน็ตผ่าน Proxy server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ไคลเอนต์ ส่งการร้องขอ (HTTP request) ไปยัง proxy server
2. หลังจากได้รับการร้องขอแล้ว Proxy server ก็จะแปลความหมายของคำร้องขอนั้น แล้วตรวจสอบดูว่ามีเอกสารตามที่ร้องขอในแคชหรือไม่ ถ้ามีก็จะส่งเอกสารตอบกลับ (HTTP response) ไปยังผู้ใช้
3. หากไม่พบเอกสารนั้นในแคช หรือข้อมูลในเอกสารนั้นไม่ update พอ หรือไม่ใช่ข้อมูลล่าสุด proxy server ก็จะส่งคำร้องขอ (HTTP request) ไปยังเว็บเซิร์ฟเวอร์
4. เมื่อเว็บเซิร์ฟเวอร์ได้รับคำร้องขอจาก Proxy server แล้วส่งเอกสารกลับมา (HTTP response)
5. เมื่อได้รับเอกสารจากเว็บเซิร์ฟเวอร์แล้ว Proxy server อาจจะเก็บเอกสารไว้ในแคช
6. Proxy server ส่งเอกสารนั้นตอบกลับ (HTTP response) ไปให้แก่ ไคลเอนต์

ความสามารถอีกอย่างหนึ่งของ WAP Gateway คือ ทำงานเป็น Proxy server และแคชด้วย แต่หน้าที่หลักจริงนั้น คือ แปลงรูปแบบการสื่อสารระหว่างฝั่งเครือข่าย อินเทอร์เน็ต และฝั่งเครือข่ายไร้สาย หรือที่เรียกว่า "**Protocol conversion**" สามารถแสดงได้ดังรูป แสดงขั้นตอนการเรียกดูข้อมูล WAP ที่อยู่ในเว็บเซิร์ฟเวอร์ โดยใช้โทรศัพท์มือถือเรียกผ่าน WAP Gateway ที่แสดงขั้นตอนดังนี้



รูปที่ 2.5 ขั้นตอนการร้องขอเอกสารระบบ WAP และอินเตอร์เน็ตผ่าน WAP Gateway

1. ผู้ใช้โทรศัพท์มือถือ (ไคลเอนต์) ส่ง URL ของเอกสารที่ต้องการไปยัง WAP Gateway โดยส่งเป็นคำร้องขอในรูปแบบโพรโตคอล WSP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. WAP Gateway ถอดรหัส (Decode) คำร้องที่อยู่ในรูปแบบไบนารี (WSP request) เพื่อแปลงให้อยู่ในรูปแบบ HTTP โดยอาจอาศัย mapping table ที่มีอยู่ใน WAP Gateway เป็นตัวช่วย (ทั้งนี้วิธีการถอดรหัสจะขึ้นกับเทคนิคของผู้ผลิต และผู้พัฒนาระบบของ WAP Gateway แต่ละราย)
3. WAP Gateway สร้างการเชื่อมต่อ (Connection) ไปยังเว็บเซิร์ฟเวอร์แล้วส่งคำร้องขอในรูปแบบโพรโตคอล HTTP
4. เว็บเซิร์ฟเวอร์จะประมวลผลคำร้องขอนั้น และตรวจสอบดูว่า เอกสารตามที่ร้องขอเป็นลักษณะซอร์ชโค้ด WML ธรรมดา (Static) หรือไม่ หากเอกสารนั้นเรียกการทำงานของสคริปต์ต่างๆ เช่น CGI , ASP ก็จะประมวลผลสคริปต์ก่อน เพื่อให้กลายเป็นเอกสาร WML ธรรมดาที่ประกอบแท็กและข้อความ
5. เว็บเซิร์ฟเวอร์ส่งเอกสารนี้กลับมาที่ WAP Gateway โดยส่งคำตอบกลับในรูปแบบโพรโตคอล HTTP
6. WAP Gateway จะเข้ารหัสเอกสาร (Encode) ไปเป็นรูปแบบไบนารี โดยอาจจะอาศัย mapping table เป็นตัวช่วย
7. WAP Gateway สร้างการติดต่อไปยังไคลเอนต์แล้วส่งข้อมูลไบนารี เป็นคำตอบกลับในรูปแบบโพรโตคอล WSP ไปยังไคลเอนต์ต่อไป

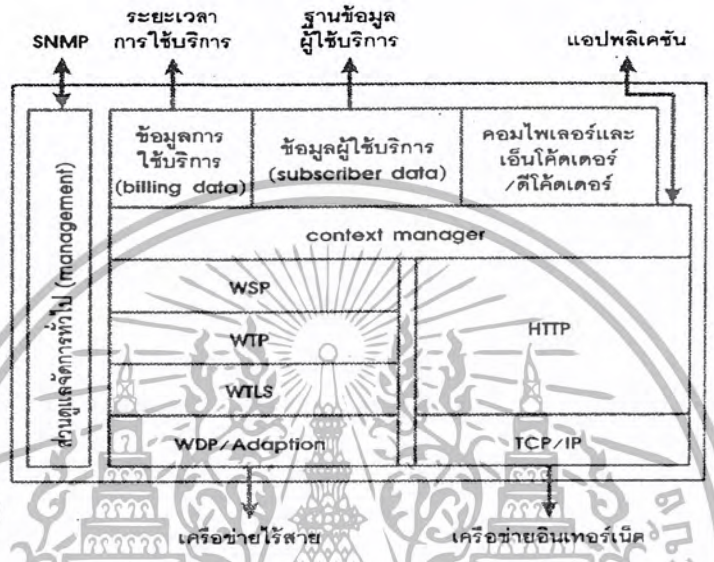
จากหลักการข้างต้น WAP Gateway คือ การแปลงรูปแบบข้อมูลให้เหมาะกับการสื่อสารในแต่ละฝั่ง เอกสารที่ส่งมาจากเว็บเซิร์ฟเวอร์ จะอยู่ในรูปแบบของข้อความและแท็ก WML โดยที่จะอาศัยโพรโตคอล HTTP ช่วยในการส่งไฟล์เอกสาร แต่เมื่อมาถึง WAP Gateway เอกสารเหล่านั้นต้องถูกเข้ารหัสให้เป็นข้อมูลแบบไร้สาย ทั้งนี้เป็นข้อกำหนดของโพรโตคอล WSP (Wireless Session Protocol) ซึ่งเป็นหนึ่ง ในโพรโตคอลของ WAP ที่บังคับว่าข้อมูลต้องเป็นรูปแบบไบนารีจึงจะส่งได้ ดังนั้น **protocol conversion** คือ การเปลี่ยนรูปแบบข้อมูลให้ตรงตามข้อกำหนดของโพรโตคอลที่ใช้ในฝั่งอินเทอร์เน็ตกับที่ใช้ในเครือข่ายไร้สาย

หมายเหตุ mapping table คือ ตารางที่ใช้ในการเปรียบเทียบว่า แท็กคำสั่ง และข้อมูลเอกสาร WML แต่ละแท็ก และแต่ละหน่วยข้อมูลจะต้องแทนด้วยรหัสไบนารีว่าเป็นอย่างไรซึ่งจะต้องเป็นไปตามข้อกำหนด WSP specification และถือเป็นหัวใจของผู้ที่พัฒนา WAP Gateway ต้องเข้าใจอย่างทะลุปรุโปร่งเลยทีเดียว ดังนั้นแท้จริงแล้วขั้นตอนในการเข้ารหัสข้อมูล คือ การเปลี่ยนข้อมูล TEXT ไปเป็นรหัสไบนารีโดยอาศัย mapping table นั่นเอง ส่วนการถอดรหัสข้อมูลก็จะเป็นกระบวนการตรงกันข้ามกันนั่น คือ เปลี่ยนข้อมูลไบนารี (WSP) ไปเป็นข้อมูลแบบ TEXT (HTTP)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้าง WAP Gateway และฟังก์ชั่นพื้นฐาน

เมื่อรู้ขั้นตอนในการทำงานของ WAP Gateway แล้วก็จะต้องรู้ถึงโครงสร้างของ WAP Gateway ด้วยซึ่งแสดงดังรูปที่ 2.6



รูปที่ 2.6 โครงสร้างพื้นฐานของ WAP Gateway

เนื่องจาก WAP Gateway เป็นระบบที่มีผู้ผลิต และพัฒนาออกมากันหลายราย ดังนั้นแต่ละรายอาจเพิ่มเติมความสามารถอื่น ๆ อีกก็ได้ แต่ในที่นี้จะกล่าวรายละเอียดเฉพาะฟังก์ชั่นพื้นฐานในรูปเท่านั้น โดย WAP Gateway จะมีทั้งชุดโพรโตคอล WAP (WSP, WTP, WTLS และ WDP) และในชุดโพรโตคอล TCP/IP ทั้งนี้เนื่องจาก WAP Gateway เป็นตัวกลางระหว่างเครือข่ายอินเทอร์เน็ต และเครือข่ายไร้สายจึงต้องมีคุณสมบัติรองรับโพรโตคอลที่ใช้ในแต่ละฝั่ง เมื่อติดต่อกับเว็บเซอร์เวอร์ก็ต้องอาศัยชุดโพรโตคอล TCP/IP ส่วนการติดต่อกับไคลเอนต์ หรือโทรศัพท์ที่ใช้ระบบ WAP ก็ต้องอาศัยชุดโพรโตคอลของ WAP

กลุ่ม WAP Forum ได้เสนอชุดโพรโตคอลสำหรับ WAP เพื่อการสื่อสารด้วยเครือข่ายไร้สาย โดยเฉพาะซึ่งแนวคิดของชุดโพรโตคอล WAP นี้ก็เป็นอีกเรื่องหนึ่งที่นำเอาแนวความคิดของชั้นสื่อสารในระบบอินเทอร์เน็ตมาประยุกต์อีกครั้ง โพรโตคอลทั้งหลายในชั้นสื่อสาร WAP ก็เทียบเคียงได้กับโพรโตคอลในอินเทอร์เน็ต เช่น HTTP TCP/IP หรือแม้กระทั่งระบบรักษาความปลอด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภัยแบบ SSL (Secure Socket Layer) องค์ประกอบอื่นๆ ของ WAP Gateway ดังรูปที่ 2.6 สามารถอธิบายเพิ่มเติมได้ดังนี้

คอมไพเลอร์ และเอ็นโค้ดเดอร์ / ดีโค้ดเดอร์ เป็นส่วนที่จะขาดไม่ได้เลย สำหรับ WAP Gateway เพราะเอ็นโค้ดเดอร์ (Encoder) เป็นตัวจัดการเกี่ยวกับการเข้ารหัสเอกสาร WML ที่ส่งมาจากเว็บเซิร์ฟเวอร์ให้เป็นข้อมูลรูปเลขไบนารี เพื่อส่งผ่านเครือข่ายไร้สายไปให้แก่ไคลเอนต์ และดีโค้ดเดอร์ (Decoder) เป็นตัวถอดรหัสข้อมูลจากไบนารีที่ได้รับจากไคลเอนต์ ให้เป็นข้อมูลในรูปแบบที่ใช้ในโพรโตคอล HTTP เพื่อส่งไปยังเว็บเซิร์ฟเวอร์ ส่วนคอมไพเลอร์ (Compiler) เป็นตัวจัดการเกี่ยวกับการคอมไพล์โค้ด WML Script ที่ส่งมาจากเว็บเซิร์ฟเวอร์ก่อนส่งต่อไปยังเครือข่ายไร้สาย

ส่วนของข้อมูลการให้บริการ (Billing Data) และข้อมูลผู้ใช้บริการ (Subscriber Data) จะเกี่ยวข้องกับข้อมูลต่าง ๆ ของผู้ใช้โทรศัพท์ เช่น ตรวจสอบ username และ password ว่าถูกต้องหรือไม่ หน้าจอเริ่มต้นการใช้งานของโทรศัพท์เป็นอย่างไร และบันทึกระยะการใช้งานของผู้ใช้โทรศัพท์

หน้าที่หลักของ WAP Gateway

คุณสมบัติของโปรแกรม WAP Gateway แต่ละตัวไม่เหมือนกัน เนื่องจากผู้ผลิตแต่ละรายอาจเพิ่มหน้าที่บางอย่างเข้าไปแต่โดยทั่วไป WAP Gateway มีหน้าที่ดังนี้

1. รองรับชุดโพรโตคอลของ WAP และชุดโพรโตคอลในอินเทอร์เน็ต
2. Protocol Conversion
3. เข้ารหัสเอกสาร WML ให้เป็นรู้แบบข้อมูลแบบไบนารี
4. คอมไพล์โค้ด WML Script
5. เป็น Proxy Server เพื่อให้บริการข้อมูลที่ถูกเรียกใช้บ่อยๆ
6. ดูแลจัดการด้านการรักษาความปลอดภัยของข้อมูล
7. เปลี่ยนเอกสาร HTML ที่ได้รับจากเว็บเซอร์เวอร์ให้เป็นเอกสาร WML

Protocol Conversion

โดยปกติการส่งข้อมูลไปมาระหว่างเบราเซอร์ และเว็บเซอร์เวอร์นั้นจะอาศัยโพรโตคอล HTTP นั้น นอกจากตัวเอกสาร HTML ที่ประกอบไปด้วยแท็กคำสั่งต่างๆที่เรารู้จักกันดี จะต้องมีข้อมูลอีกส่วนหนึ่งอยู่ที่ช่วงต้นของเอกสารด้วยเสมอ เรียกว่า "**เฮดเดอร์**" (HTTP header) ซึ่งบ่งบอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดที่เกี่ยวข้องกับเอกสารนั้น ๆ เช่น ประเภทของข้อมูลในเอกสารว่าเป็นรูปภาพ หรือข้อความธรรมดา ความยาว หรือขนาดของข้อมูลหรือเอกสาร เป็นต้น

ตัวอย่าง เฮดเดอร์ของไฟล์ HTML ที่ส่งมาจากเว็บเซอร์เวอร์เป็นดังนี้

Date Sat , 13 Aug 2000 02:05:47 GMT

Content -Type : text/html

Content - Length : 86

ได้บอกรายละเอียดให้ทราบว่าข้อมูลที่ส่งออกมาเป็นเอกสาร HTML มีความยาว 86 ไบต์ พร้อมทั้งบอกวันเวลาที่ส่ง

ในกรณีที่เอกสารถูกส่งยังเครือข่ายไร้สาย WAP Gateway ก็ต้องเข้ารหัสเฮดเดอร์ให้เป็นแบบไบนารีด้วย เนื่องจากเฮดเดอร์อยู่ในรูปของระบบโพรโตคอล HTTP การเข้ารหัสจึงเป็นการแปลงเฮดเดอร์ให้สามารถส่งต่อไปในระบบ โพรโตคอล WSP เหตุผลที่ต้องเข้ารหัสก็เพื่อลดข้อจำกัดทางด้าน band width และ latency ของเครือข่ายไร้สายซึ่งแสดงให้ดูตามตัวอย่างข้างล่าง

ตัวอย่างที่ 1 HTTP header ที่ส่งจากไคลเอนต์ไปยังเว็บเซอร์เวอร์มีข้อความบ่งบอกภาษาที่ใช้ทางฝั่งไคลเอนต์ ดังนี้

Accept – Language : en,sv

เมื่อเทียบกับรหัสเลขไบนารีที่ใช้ส่งจากโทรศัพท์ผ่านมายังเครือข่ายไร้สายไปยัง WAP Gateway จะมีลักษณะเป็นข้อมูลแบบไบนารีดังนี้

0x83 0x99 0x83 0xF0

ตัวอย่างที่ 2 HTTP header ที่ส่งมาจากเว็บเซอร์ไปยังไคลเอนต์มีข้อความบ่งบอกเวลาส่งข้อมูลดังนี้

Date: Sat, 13 Aug 2000 02:05:47 GMT

เมื่อเข้ารหัสเป็น WSP header แล้วจะได้ผลลัพธ์ของข้อมูลแบบไบนารีดังนี้

0x92 0x04 0x35 0x3f 0x45 0x11

ตัวอย่างที่ 3 WSP header ของคำร้องขอไฟล์ index.wml ใน http:// weco.csie.fju.edu.tw ซึ่งมาจากโปรแกรม Nokia WAP Toolkit (โปรแกรมทูลคิตของโนเกียซึ่งเป็นเครื่องมือที่ใช้ในการพัฒนา WAP Application) มีลักษณะเป็นข้อมูลไบนารีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

01 40 25 68 74 74 70 3α 2f 2f 77 65 63 6f 2e 63 73 69 65 2e 66 6α
75 2e 65 64 75 2e 74 77 2f 69 6e 64 65 78 2e 77 6d 6c 81 6α 81 84
80 94 80 95 80 α1 80 9d α9 4e 6f 6b 69 61 2d 57 41 50 2d 54 6f 6f
6c 6b 69 74 2f 31 2e 33 62 65 74 61 00

เมื่อ WAP Gateway ถอดรหัส WSP header ให้เป็น HTTP header แล้วจะมีลักษณะดังนี้

TID = 1

Type = GET

URL = http://weco.csie.fji.edu.tw/index .wml

Accept - Char set: Somali, Amharic

Accept: application /vnd.wap.wmlc,

Application/vnd.wap.wmlscriptc,image/vnd.wap.wbmp,image/gif

User- Agent : Nokia- WAP-Toolkit/1.3 beta

เข้ารหัสเอกสาร WML ให้เป็นข้อมูลไบนารี

เอกสาร WML เหล่านี้ถูกส่งมาจากเว็บเซอร์เวอร์ผ่านเครือข่ายอินเทอร์เน็ต โดยมีรูปแบบเป็นข้อความที่สามารถอ่านได้ แต่เมื่อ WAP Gateway ได้รับเอกสารแล้วก่อนที่จะเข้ารหัส (Encode) ให้เป็นข้อมูลแบบไบนารีเพื่อส่งต่อไปยังไคลเอนต์ WAP Gateway ก็จะคอยตรวจสอบความถูกต้องตามหลักไวยากรณ์ของเนื้อหาว่าเป็นไปตามกฎของภาษา XML (Extensible Markup Language) หรือไม่เพราะภาษา WML เป็นส่วนหนึ่งของ XML และต้องยึดถือกฎเกณฑ์ของ XML ด้วย หาก WAP Gateway พบว่าเอกสารใน WML ไม่ถูกต้องตามหลักไวยากรณ์ของ XML สำหรับ WAP Gateway บางตัวอาจจะแสดงข้อความผิดพลาดมาแสดงที่ไคลเอนต์เลย แต่บางตัวก็อาจแสดงข้อความที่ไม่ถูกต้องออกมาเลยโดยไม่แจ้งข้อความผิดพลาด

ตัวอย่าง การเข้ารหัสเอกสาร WML

เอกสาร WML ที่ส่งมาจาก WAP Application Server หรือเว็บเซอร์เวอร์ มีเนื้อหาดังนี้

< wml >

< card id = “abc” orderd = “true” >

< p >

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
< do type = “accept”>
        <go herf = http://xyz.org/s/>
<do>
        X: $(X)  <br/>
        Y: $(&#x59;)  <br/>
        Enter name : <input type = “text” name =”N”/>
</p>
</card>
</wml>
```

เมื่อผ่านการเข้ารหัสที่ WAP Gateway แล้วจะได้ข้อมูลที่มีลักษณะเป็นไบนารี ดังนี้

02 08 6A 04 ‘X’ 00 ‘Y’ 00 7F E7 55 03 ‘α’ ‘b’ ‘c’ 00
03 01 60 E8 38 01 AB 4B 03 ‘X’ ‘Y’ ‘Z’ 00 08 03
‘S’ 00 01 01 03 ‘:’ ‘ ’ 00 82 00 26 03 ‘ ’ ‘Y’
‘:’ ‘ ’ 00 82 02 26 03 ‘ ’ ‘E’ ‘n’ ‘t’ ‘e’ ‘r’ ‘ ’ ‘n’
‘α’ ‘m’ ‘e’ ‘:’ ‘ ’ 00 AF 48 21 03 ‘N’ 00 01 01 01 01

คอมไพล์โค้ด WML Script

โค้ด WML Script ที่ส่งจากฝั่งเว็บเซอร์เวอร์จำเป็นต้องให้ WAP Gateway คอมไพล์ (Compile) โดยที่ WAP Gateway นั้นจะต้องคอยตรวจดูไวยากรณ์ (Syntax) ของภาษาก่อน ผลของการคอมไพล์จะได้ข้อมูลเป็นไบต์โค้ด (Byte Code) ซึ่งเป็นข้อมูลไบนารีแบบหนึ่ง เมื่อไบต์โค้ดเหล่านี้ถูกส่งต่อไปยังโทรศัพท์มือถือก็ต้องผ่านกระบวนการแปล และประมวลผลเพื่อให้ทำงานตามโค้ดที่เขียนเอาไว้ การเข้ารหัสเอกสาร WML กับการคอมไพล์โค้ดภาษา WML Script เป็นกระบวนการที่ไม่เหมือนกันซึ่งสามารถอธิบายแจกแจงได้ดังนี้

1. การเข้ารหัสเอกสาร WML อาศัยการเปรียบเทียบ (mapping) แท็ก WML ต่าง ๆ กับรหัสไบนารีที่กำหนดไว้แล้วในตาราง mapping table ซึ่งเรียกว่า WSP specification เช่น คำว่า **“Accept – Language”** ใน HTTP header แทนด้วยรหัส 0x83 ส่วน **“Date”** ก็แทนด้วย 0x92 เป็นต้น ส่วนแท็กอื่นๆ ก็จะมีรหัสไบนารีเฉพาะของแต่ละแท็ก

2. การคอมไพล์โค้ด WML Script เป็นกระบวนการอีกลักษณะหนึ่ง ซึ่งคล้ายกับการคอมไพล์โปรแกรม คือ จะต้องมีการตรวจสอบไวยากรณ์ของภาษาด้วยไม่ได้อาศัยการเปรียบเทียบหรือ mapping เหมือนกับการเข้ารหัสภาษา WML

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

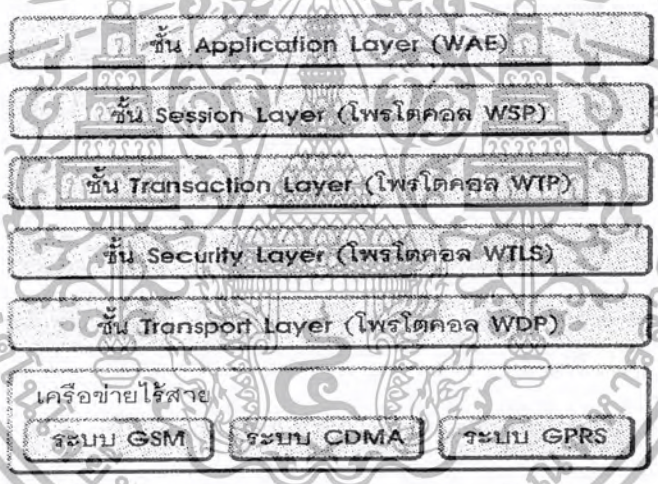
สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

3. เนื้อหา WML ที่ผ่านมาเข้ารหัสข้อมูลแบบไบนารีมาจาก WAP Gateway แล้วก่อนที่จะแสดงผลจริง ๆ โดยเบราเซอร์ที่ไคลเอนต์ได้ก็จะต้องมีการถอดรหัสกลับคืน โดยอาศัยการเปรียบเทียบ (mapping) ในทำนองเดียวกับการเข้ารหัส

4. ส่วนของ WML Script นั้น เมื่อไบต์โค้ดมาถึงไคลเอนต์ก็จะถูกแปลงกลับคืนโดยอาศัยอินเทอร์พรีตเตอร์ (Interpreter) ซึ่งวิธีนี้ต้องไปยุ่งเกี่ยวกับการสถานะของตัวแปร ค่าพอยน์เตอร์ และอื่นๆ อีกมากตามลักษณะการทำงานของโปรแกรมโดยทั่วไป

ชั้นสื่อสารโพรโตคอลของ WAP

สำหรับชั้นสื่อสารโพรโตคอลของ WAP ก็ประกอบไปด้วยโพรโตคอลต่างๆ ได้แก่ WSP WTP WTLS และ WDP ซึ่งสามารถได้ดังรูปที่ 2.7



รูปที่ 2.7 ชั้นสื่อสาร และโพรโตคอลใน WAP

WAE วางองค์ประกอบของการทำงาน

ชั้น WAE (Wireless Application Environment) เป็นโพรโตคอล แต่เรียกว่า เป็นมาตรฐานที่เอื้ออำนวยในการพัฒนาแอปพลิเคชั่น สำหรับเครือข่ายไร้สายจะเหมาะสมกว่า หรือพูดให้เข้าใจง่ายๆ ก็คือ เป็นตัวบ่งบอกให้ทราบว่าเมื่อเราต้องการพัฒนาแอปพลิเคชั่น สำหรับ WAP จะต้องมีอะไรมาเกี่ยวข้องบ้างที่เห็นได้ชัดก็คือ ภาษา WML กับ WML Script หรือแม้กระทั่ง WAP Browser เป็นต้น ภายใน WAE จะมี User Agent ซึ่งก็ คือ ซอร์ฟแวร์ หรืออุปกรณ์ใดๆ ที่แปลเนื้อหาของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

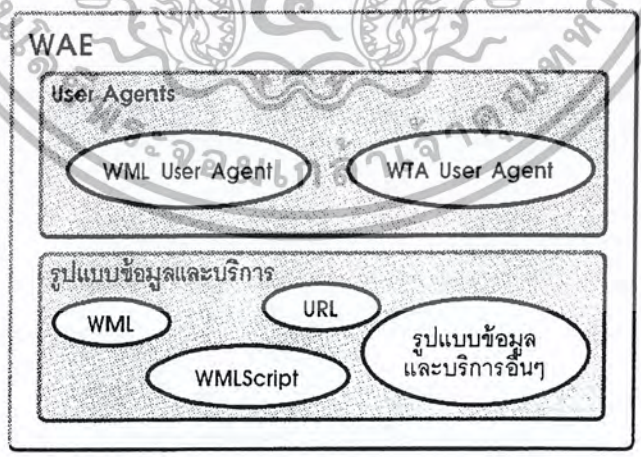
55668

(Interpret) เอกสาร HTML WML WML Script ฯลฯ แล้วนำมาแสดงผล ตัวอย่างของ User Agent ที่เห็นได้ชัดๆ ก็คือ เบราเซอร์ต่างๆ ไม่ว่าเป็นเว็บเบราเซอร์หรือ WAP Browser ซึ่งใน WAE มี User agent 2 ตัว คือ

WML User Agent 2 ตัว เช่น WAP Browser หรือไมโครเบราเซอร์ซึ่งติดตั้งอยู่ในส่วนโทรศัพท์มือถือ หรือใน WAP Emulator ต่างๆ

WTA User Agent จะทำงานที่เกี่ยวข้องกับฟังก์ชันการทำงานของโทรศัพท์ซึ่ง WTA (Wireless Telephony Application) เป็นกลุ่มของแอปพลิเคชันที่มีกลไกในการติดต่อ หรือเกี่ยวข้องกับโทรศัพท์มือถือ ตัวอย่าง เราสามารถสร้าง WAP Application ให้แสดงรายชื่อร้านอาหารจำนวนหนึ่งบนหน้าจอ แต่ละรายชื่อก็มีลิงก์อยู่ด้วย หากมีการคลิกที่รายชื่อร้านใดโทรศัพท์ของเราก็จะโทรติดต่อไปยังร้านอาหารนั้นทันทีเพื่อจองโต๊ะอาหาร จะเห็นได้ว่าแอปพลิเคชันดังกล่าวน่าสนใจไม่น้อยทีเดียว อย่างไรก็ตามโทรศัพท์มือถือที่วางขายกันตามท้องตลาดยังไม่สนับสนุน WTA ทั้งๆ ที่แนวความคิดเกี่ยวกับ WTA มีมาตั้งแต่ WAP 1.1 แล้วก็ตามแต่แทบจะไม่มีการสนับสนุนโดยผู้ผลิตโทรศัพท์ หรือผู้ให้บริการเครือข่ายใดเลย มาจนถึง WAP 1.2 ก็มี specification หรือข้อกำหนดของ WTA ชัดเจนขึ้นแต่ยังไม่ค่อยพบการใช้งานจริง ตามรูปที่ 2.8 และอีกองค์ประกอบหนึ่ง คือ **"รูปแบบของข้อมูลและบริการ"** หมายถึง สิ่งที่จะนำไปใช้กับ User Agent

ตัวอย่าง WML WMLScript ก็สามารถนำไปแสดงใน WML User Agent (WAP Browser) ได้



รูปที่ 2.8 ส่วนประกอบของ WAE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WML

ภาษา WML เป็นภาษาที่เกี่ยวข้องกับแท็กต่างๆ เหมือนกับภาษา HTML ที่เราคุ้นเคยแต่ภาษานี้ได้รับการพัฒนา โดย WAP Forum และได้รับอิทธิพลมาจาก 2 ภาษา คือ ภาษา HDML เวอร์ชั่น 2 ของบริษัท Phon.com (ชื่อเต็ม คือ บริษัท Unwired Planet) และภาษา HTML เวอร์ชั่น 4 โดยนำมาดัดแปลงและปรับปรุงให้เหมาะสมกับสภาพของเครือข่ายไร้สายและอุปกรณ์มือถือ ถ้าเป็นเอกสาร HTML แล้ว 1หน้าจอที่แสดง คือ 1 ไฟล์ แต่ในเอกสาร WML นั้น หน้าจอเรียกว่า 1 คาร์ด หลายๆ คาร์ดจะรวมกันเป็น 1 เดค ซึ่ง 1 เดค ก็เปรียบเทียบ 1 ไฟล์นั่นเอง เหตุผลที่ต้องทำเช่นนี้ก็เนื่องมาจากหน้าจอของอุปกรณ์มือถือมีขนาดเล็กเกินกว่าที่จะแสดงผลข้อมูลจำนวนมากจึงต้องแบ่งคาร์ดย่อยๆ และให้ผู้ใช้งาน "โต้ตอบ" กับคาร์ดได้ โดยเลือกคำสั่งการทำงานหรือป้อนข้อมูลในแต่ละคาร์ด เช่น ใส่ข้อความ (Input) เข้าไปในช่องที่กำหนดไว้ เลือกคำตอบที่ต้องการจากรายการที่แสดงออกมาให้เลือก เป็นต้น เอกสาร WML จะถูกส่งจากเว็บเซอร์เวอร์ หรือWAP Application Server ทีละเดคมาเก็บไว้ในหน่วยความจำของ User Agent ดังนั้นหลังจากที่เราโต้ตอบกับคาร์ดหนึ่ง แล้วเปลี่ยนไปยังอีกคาร์ดของในเดคเดียวกัน User Agent ก็ไม่ต้องร้องขอคาร์ดใหม่จากเว็บเซอร์เวอร์เพราะมีอยู่ในหน่วยความจำแล้วคุณสมบัติของ WML มีดังนี้

- แสดงข้อมูลได้ทั้งข้อความและรูปภาพ คุณสมบัติคล้ายคลึงกับภาษา HTML เช่นกัน โดยการใช้แท็กต่างๆ ในการควบคุมการแสดงผลข้อมูล เช่น ย่อหน้า ตัวอักษรเอน- หนา การจัดตำแหน่งข้อความให้ชิดซ้าย-ชิดขวา เป็นต้น แต่ส่วนที่แตกต่างกันมาก ก็คือ ภาษา HTML รองรับการแสดงรูปภาพหลายรูปแบบ เช่น GIF JPG ส่วนภาษา WML รองรับการแสดงรูปภาพได้รูปแบบเดียว คือ WBMP (Wireless Bitmap) ซึ่งเป็นภาพลักษณะขาวดำและไฟล์ที่มีขนาดเล็ก แต่ถึงแม้ WML จะมีข้อจำกัดมากกว่า HTML ในการแสดงรูปภาพก็ตามดีที่สามารถใช้ซอฟต์แวร์ช่วยแปลงภาพ JPG GIF หรือรูปแบบอื่นๆ มาเป็นภาพชนิด WBMP ได้ไม่ยาก

- แสดงแบบฟอร์มรับข้อมูลในลักษณะต่างๆได้เช่นเดียวกับภาษา HTML ทั้งแบบฟอร์มที่ให้ผู้ใช้กรอกข้อความ ตัวเลือกแบบ checkbox ให้กาเครื่องหมายเลือก และตัวเลือกแบบ radio button เป็นต้น แต่มีข้อแตกต่างกันในแง่การใช้งาน คือ ภาษา HTML จะแสดงปุ่มให้คลิกในตัวแบบฟอร์มเลยแต่ภาษา WML จะต้องใช้ปุ่มที่อยู่บนโทรศัพท์มือถือแทน

- สามารถลิงค์จากคาร์ดหนึ่งไปยังคาร์ดอื่น หรือเอกสารอื่น โดยอาจจะกำหนดไฮเปอร์ลิงก์หรือปุ่ม Back ซึ่งคล้ายคลึงกับในเอกสาร HTML

- รองรับตัวอักษรแบบ Unicode ซึ่งเป็นชุดอักษรที่เป็นมาตรฐานสากล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WML Script

จัดเป็นภาษาสคริปต์ที่ได้รับการออกแบบเพื่ออุปกรณ์มือถือไร้สายโดยเฉพาะ และมีส่วนที่คล้ายคลึงกับภาษา Java Script หรือVB Script ที่ใช้พัฒนา WEB Application ซึ่งเป็นภาษาสคริปต์ทางฝั่งไคลเอนต์ (Client – Side Script) สำหรับการใช้งานภาษา WML Script มีอาทิ เช่น

- ตรวจสอบความถูกต้องของข้อมูลจากฟอร์มก่อนส่งไปยังเว็บเซิร์ฟเวอร์
- ใช้ติดต่อกับฟังก์ชั่นในโทรศัพท์มือถือ
- สร้างแอปพลิเคชั่นในลักษณะโต้ตอบกับผู้ใช้ โดยไม่ต้องอาศัยความสามารถของเว็บเซิร์ฟเวอร์เพียงแต่เวลาใช้งานจะต้องวิ่งไปเอาไฟล์ WML Script ที่เว็บเซิร์ฟเวอร์เนื่องจากฟังก์ชั่นที่เรียกใช้อยู่คนละไฟล์กับเอกสาร WML

ข้อเสียที่เห็นได้ชัดของ WML Script คือโปรแกรม WAP Browser บางส่วนไม่สนับสนุน WML Script ในเชิงเทคนิค WML Script ต้องพึ่งพาความสามารถของโทรศัพท์มือถือซึ่งมีปัญหามาก ดังนั้นผู้พัฒนา WAP Application พยายามเลี่ยงที่จะใช้ภาษา WML Script โดยหันไปใช้พวกภาษาสคริปต์ทางฝั่งเซิร์ฟเวอร์แทนเพราะอาศัยการประมวลผลที่ฝั่งเซิร์ฟเวอร์แทน แล้วจึงส่งเอกสาร WML มาที่โทรศัพท์มือถือแทน

URL (Uniform Resource Location)

ในระบบ WAP มีการอ้างถึงที่อยู่ของเอกสารด้วย และใช้มาตรฐานเดียวกัน โดยระบุโพรโตคอลนำหน้า เช่น http://wap.nectec.or.th/ เป็นต้น

WSP กำหนดสภาพการเชื่อมต่อ

โพรโตคอล WSP (Wireless Session protocol) เปรียบเสมือนเป็นโพรโตคอล HTTP 1.1 ในรูปแบบเลขไบนารีเพราะทำหน้าที่เกี่ยวกับการรับ-ส่งข้อมูลระหว่างไคลเอนต์และเซิร์ฟเวอร์ทางฝั่งเครือข่ายไร้สาย ในมุมมองของWSP นั้น ไคลเอนต์ คือโทรศัพท์มือถือ ส่วนเซิร์ฟเวอร์ คือ WAP Gateway หรือ WAP Server ก็ได้ (เนื่องจาก WAP Server คือ WAP Gateway + WAP Application Server) ข้อมูลส่วนที่เป็น WSP header ก็ต้องอยู่ในรูปแบบไบนารี เพื่อความเหมาะสมในการรับ-ส่งผ่านเครือข่ายไร้สายที่มีข้อจำกัด โดยปกติโพรโตคอล WSP อยู่ในชั้น Session Layer แต่ในความจริงแล้วในชั้นนี้แบ่งได้เป็น 2 โพรโตคอลย่อยๆ คือ WSP/B และWSP ซึ่งมีข้อแตกต่างกันเพียงเล็กน้อย

WSP/B เป็นโพรโตคอลที่ไม่ต้องสร้างการเชื่อมต่อ หรือsession ระหว่างไคลเอนต์ กับ WAP Gateway ก่อน การส่งข้อมูลจะไม่มีการตรวจสอบความถูกต้องด้วย WTP แต่อาศัย WDP ในการส่งข้อมูลโดยตรงเลย ส่วน WSP มีข้อกำหนดในลักษณะตรงข้าม คือ ต้องมีการสร้าง session หรือการเชื่อมต่อระหว่างไคลเอนต์กับ WAP Gateway ที่มั่นคงและยาวนานเพื่อให้การรับ-ส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไม่มีเหตุขัดข้อง และในอีกแง่หนึ่งยังต้องสามารถระงับการติดต่อชั่วคราว (Suspend) และสามารถเรียกเชื่อมต่อกลับมาใหม่ (Resume) โดยไม่เปลืองทรัพยากรของระบบมากนัก เหตุผลที่บางครั้งต้องระงับการเชื่อมต่อชั่วคราว ก็คือ ในกรณีที่มีการหยุดนิ่งนานๆ โดยไม่มีการรับส่งข้อมูล ซึ่งเป็นการสิ้นเปลืองทรัพยากรของระบบ นอกจากนี้ต้องมีการตรวจสอบ ความถูกต้องของข้อมูลตามข้อกำหนดของโพรโตคอล WTP ด้วย จากนั้นจะอาศัยโพรโตคอล WDP ให้ส่งข้อมูลเหมือนกับใน WSP/B

ในส่วนของ WAP ก็มีการติดต่อ Connection-Oriented คือ กรณีที่ใช้โพรโตคอล WSP และอาศัยโพรโตคอล WTP ที่อยู่ถัดไปในการจัดการตรวจสอบความถูกต้องของข้อมูล และจากนั้นก็จะอาศัยโพรโตคอล WDP ส่งข้อมูลเข้าไปในเครือข่ายไร้สาย (WSP -> WTP ->WDP) ส่วนการติดต่อแบบ Connectionless คือ กรณีที่ใช้โพรโตคอล WSP/B และอาศัยโพรโตคอล WDP ให้ช่วยการจัดส่งข้อมูลโดยตรงเลย (WSP/B -> WDP) ไม่ต้องมีการตรวจสอบความถูกต้องของข้อมูลด้วยโพรโตคอล WTP แต่ถ้ามีการป้องกันความปลอดภัยของข้อมูลก็ต้องผ่าน WTLS ด้วยสรุปลำดับชั้นของโพรโตคอลก็คือ WSP-> WTP->WTLS->WDP หรือWSP/B-> WTLS-> WDP แล้วแต่ว่าเป็นการติดต่อแบบ Connection-Oriented หรือแบบ Connectionless ขึ้นอยู่กับการตัดสินใจของผู้พัฒนา WAP Gateway แต่ละราย แต่ในปัจจุบันเป็นการติดต่อแบบ Connectionless เนื่องจากความเร็วในการรับ-ส่งข้อมูลไม่ต้องมาคอยตรวจสอบความถูกต้อง วิธีสังเกตว่าเป็นการติดต่อแบบใดก็คือ ถ้าเป็นการติดต่อแบบ Connection- Oriented จะใช้พอร์ต 9201 แต่หากเป็น Connectionless จะใช้พอร์ต 9200 ซึ่งดูจากการตั้งค่าในโทรศัพท์มือถือรองรับระบบ WAP เพื่อขอใช้บริการของ WAP Gateway แล้วจะต้องระบุพอร์ตด้วย โดยที่ผู้ใช้บริการ WAP Gateway จะกำหนดไว้ชัดเจนว่าเป็น 9200 หรือ 9201)

WTP รับประกันความน่าเชื่อถือ

โพรโตคอล WTP (Wireless Transaction Protocol) เกี่ยวข้องกับการรับประกันความน่าเชื่อถือของการส่งข้อมูลซึ่งมองดูแล้วก็คล้ายคลึงกับหน้าที่บางส่วนของโพรโตคอล TCP แต่สิ่งที่แตกต่างกันก็มีหลายประการอย่าง เช่น โพรโตคอล TCP ในด้านของการเชื่อมต่อกัน หรือ Connection-Oriented ระหว่างผู้รับ และผู้ส่ง (ไคลเอนต์ และเซิร์ฟเวอร์) รวมถึงการควบคุมการส่งข้อมูลด้วย แต่โพรโตคอล WTP จะเอนเอียงไปในเชิงกระบวนการรับ- ส่งข้อมูลไปมา หรือ Transaction-Oriented มากกว่าเพราะหน้าที่ในการเชื่อมต่ออยู่ที่โพรโตคอล WSP แล้ว นอกจากนี้โพรโตคอล WTP ยังพยายามลดกระบวนการส่งข้อมูลไปมาให้เหลือน้อยที่สุด เพราะข้อจำกัดของเครือข่ายไร้สายซึ่งมี band width แคบ และlatency สูง เมื่อผ่านการควบคุมความน่าเชื่อถือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการรับ-ส่งข้อมูลด้วยโพรโตคอล WTP แล้ว ข้อมูลก็จะถูกส่งต่อไปให้แก่โพรโตคอล WDP ซึ่งทำหน้าที่จัดส่งข้อมูลไปในเครือข่ายแบบไร้สาย

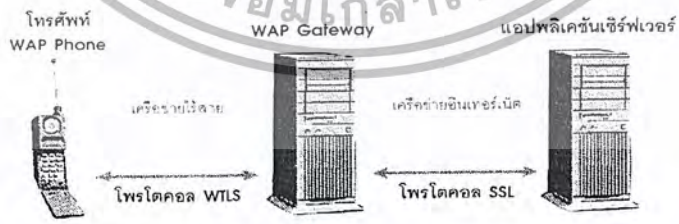
WDP ควบคุมการรับ–ส่ง

โพรโตคอล WDP (Wireless Data gram Protocol) จะคอยดูแลการส่งข้อมูลไปในเครือข่าย แต่เนื่องจากชนิดของเครือข่ายไร้สาย (Bearer) มีหลายรูปแบบ เช่น GSM CDMA GPRS ฯลฯ ดังนั้นคุณลักษณะสำคัญของโพรโตคอล WDP ก็คือ ความเป็นอิสระไม่ผูกติดกับเครือข่าย (Bearer independence) โดย WDP จะคอยอำพรางโพรโตคอลซึ่งอยู่เหนือขึ้นไปว่ากำลังทำงานกับเครือข่ายชนิดไหน ดังนั้นด้วยคุณลักษณะของโพรโตคอล WDP นี้เอง ผู้พัฒนา WAP Application จึงไม่ต้องกังวลเรื่องเครือข่ายไร้สาย โพรโตคอลที่อยู่เหนือจาก WDP ก็คือ โพรโตคอล WTP แต่ในบางกรณีที่มีการป้องกันความปลอดภัยของข้อมูล ก็จะมีอีกโพรโตคอลหนึ่งที่เป็นตัวจัดการที่เกี่ยวกับความปลอดภัยโดยเฉพาะโพรโตคอลนั้น คือ WTLS

WTLS ดูแลความปลอดภัย

เนื่องจากโพรโตคอลนี้เป็นทางเลือกเสริมของโพรโตคอล WDP ซึ่งจะมี หรือไม่มีก็ได้ทั้งนี้ขึ้นอยู่ที่ WAP Gateway และโทรศัพท์มือถือ WAP Phone ว่ารองรับโพรโตคอล WTLS หรือไม่

ส่วน SSL (Secure Socket Layer) ที่เป็นโพรโตคอลจัดการความปลอดภัยในระบบอินเทอร์เน็ตซึ่งเป็นโพรโตคอลจัดการเข้า-ถอดรหัสข้อมูลด้วยเทคนิคและกลไกต่างๆ เมื่อระบบการทำงาน WAP รับแนวความคิดจากอินเทอร์เน็ตเป็นส่วนใหญ่ ดังนั้นจึงได้รับแนวความคิดของโพรโตคอล SSL ที่เกี่ยวกับการรักษาความปลอดภัยมาด้วยเช่นกัน แล้วมาดัดแปลงให้เหมาะสมกับเครือข่ายไร้สายกลายเป็นโพรโตคอลใหม่ชื่อ WTLS ซึ่งอ้างอิงกับโพรโตคอล SSL 3.0

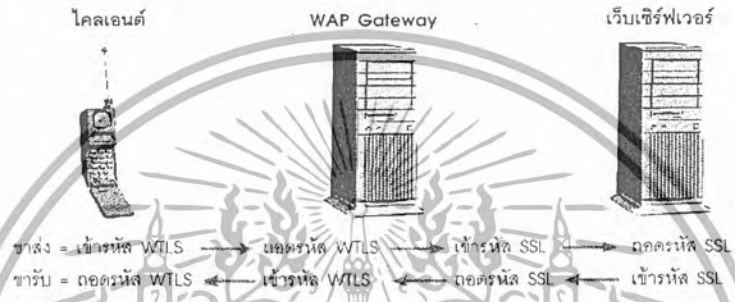


รูปที่ 2.9 ระบบรักษาความปลอดภัยของข้อมูลใน WAP และอินเตอร์เน็ต

จากรูป 2.9 เป็นการแสดงกลไกของการรักษาความปลอดภัย โดยสามารถเข้ารหัสด้วยโพรโตคอล WTLS จะเกิดขึ้นได้ก็ต่อเมื่อ WAP Gateway และโทรศัพท์มือถือล้วนแล้วแต่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รองรับโพรโตคอล WTLS ทั้งคู่ซึ่งในกรณีนี้ข้อมูลที่ส่งไป – มา ระหว่าง WAP Gateway และโทรศัพท์มือถือจะต้องถูกเข้ารหัสด้วยข้อกำหนดของโพรโตคอล WTLS และบีบอัดให้มีขนาดเล็กลง จะได้เหมาะสมกับเครือข่ายไร้สาย ส่วนรูปที่ 2.10 แสดงลำดับการทำงานของการรับส่งข้อมูลซึ่งแบ่งเป็น 2 กรณี คือ การส่ง (จากโทรศัพท์มือถือ -> เว็บเซิร์ฟเวอร์) และจากการรับ (เว็บเซิร์ฟเวอร์ -> โทรศัพท์)



รูปที่ 2.10 การเข้ารหัส และถอดรหัสเพื่อรักษาความปลอดภัยของข้อมูลทั้ง 2 ทิศทาง

กรณีของการส่งทางฝั่งโทรศัพท์มือถือ จะเข้ารหัสข้อมูลด้วยข้อกำหนดของโพรโตคอล WTLS แล้วข้อมูลจะถูกส่งผ่านเครือข่ายไร้สายมาถึง WAP Gateway มาถึงตรงนี้จะมีอยู่ 2 ขั้นตอนย่อยๆ ซึ่งกินเวลาสั้นมากๆ ในหน่วยมิลลิวินาที คือ การถอดรหัสข้อมูลได้ด้วยข้อกำหนดของโพรโตคอล WTLS จากนั้นก็จะถูกถอดรหัสด้วยข้อกำหนดของโพรโตคอล WTLS จากนั้นก็เข้ารหัสอีกครั้งตามข้อกำหนดของโพรโตคอล SSL เพื่อส่งข้อมูลผ่านเครือข่ายอินเทอร์เน็ต และเมื่อมาถึงเมื่อมาถึงเซิร์ฟเวอร์ข้อมูลจะถูกถอดรหัสตามข้อกำหนดของโพรโตคอล SSL เพื่อนำไปใช้ หรือประมวลผลอีกครั้งหนึ่ง กรณีขารับ จะดำเนินการที่ตรงกันข้ามกับการขาส่งซึ่งจุดที่สำคัญ คือ กระบวนการที่เกิดขึ้นในหน่วยความจำของ WAP Gateway อันเป็นช่วงรอยต่อระหว่างการเปลี่ยนโพรโตคอล (WTLS <-> SSL) ซึ่งข้อมูลมิได้เข้ารหัสใดๆ เลยถึงแม้ว่าช่วงเวลานี้จะสั้นมากเพียงแค่ระดับมิลลิวินาทีก็ตาม แต่ผู้พัฒนา WAP Gateway ก็จำเป็นต้องให้ความสำคัญและระมัดระวังอย่างมากในการออกแบบระบบ อย่าให้มีการเก็บข้อมูลที่ถูกถอดรหัสไว้ในแหล่งเก็บข้อมูลอื่นๆ รวมทั้งต้องมีระบบรักษาความปลอดภัยในเสี้ยววินาทีที่ถอดรหัสจากโพรโตคอลหนึ่ง และเข้ารหัสใหม่ด้วยอีกโพรโตคอลหนึ่ง

ส่วน WAP Server คือ อุปกรณ์ที่เป็นทั้งเว็บเซิร์ฟเวอร์กับ WAP Gateway ดังนั้น WAP Server จะไม่มีช่วงเปลี่ยนโพรโตคอล เพราะใช้เพียงโพรโตคอล WTLS เพียงอย่างเดียวซึ่งถือเป็น

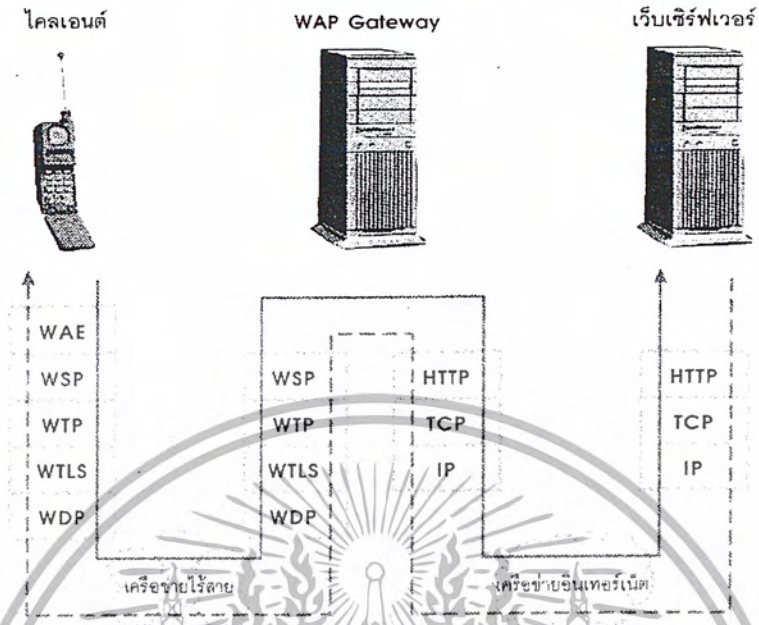
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อปลอดภัยข้อหนึ่งก็ว่าได้ ถึงแม้ว่าโพรโตคอล WTLS จะเป็นทางเลือกเสริมและน่าเสียดายว่าเป็นเพียงแนวคิดที่น่าสนใจ เพราะระบบ WAP จะต้องเกี่ยวข้องกับการทำธุรกิจทางการเงินอยู่แล้ว เทคโนโลยีที่เกี่ยวกับความปลอดภัยใน WAP จึงต้องเกิดขึ้นอย่างหลีกเลี่ยงไม่ได้ ทั้งนี้ใน WAP 1.3 ออกมาในช่วงปี 2000 ได้มีการพัฒนาระบบป้องกันความปลอดภัยขึ้นอีกระดับ คือ Wireless Public Key Infrastructure (WPKI) โดยเป็นระบบที่ออกแบบมาเพื่อการทำธุรกรรม M- Commerce (Mobile Commerce) ที่ใช้บัตรเครดิตโดยเฉพาะ และเกี่ยวข้องกับลายเซ็นดิจิตอลเหมือนกับ WIM แต่มีข้อแตกต่างกันตรงที่ว่าการจัดการเรื่องความปลอดภัยของ WIM จะอยู่ที่ฝั่งผู้ใช้หรือไคลเอนต์ (โดยใช้สมาร์ตการ์ด) แต่ในส่วน WPKI ถูกจัดการโดยองค์กรกลางที่เรียกว่า "CA" (Certification Authority) ซึ่งจะทำหน้าที่จัดการด้านความปลอดภัยทั้งทางฝั่งผู้ใช้ และฝั่งเซิร์ฟเวอร์ด้วยแสดงได้ดังรูปที่ 2.11

รูปที่ 2.11 แบบจำลองการทำงานของ WPKI ใน WAP 1.3

โพรโตคอลของ WAP สามารถแสดงภาพรวมของการเชื่อมต่อ ดังรูปที่ 2.12 แสดงภาพรวมของโพรโตคอลที่เกี่ยวข้องในเครือข่ายไร้สาย และอินเทอร์เน็ต จะเห็นได้ว่า WAP Gateway ต้องเป็นอุปกรณ์ที่รองรับโพรโตคอลทั้ง 2 ชุด แต่จุดที่น่าสังเกตุ WAP Gateway ไม่มีชั้น WAE เพราะไม่ถือว่าเป็นโพรโตคอลสื่อสาร แต่เป็นมาตรฐานที่ใช้ในการพัฒนา WAP Application เช่น ภาษา WML WML Script ฯลฯ ดังนั้นจึงมีการใช้งานแต่เฉพาะทางฝั่งโทรศัพท์มือถือ ส่วนใน WAP Gateway มีเฉพาะโพรโตคอลที่ใช้ในการสื่อสารเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.12 แสดงภาพรวมของโพรโตคอลที่เกี่ยวข้องในเครือข่ายไร้สาย และอินเทอร์เน็ต

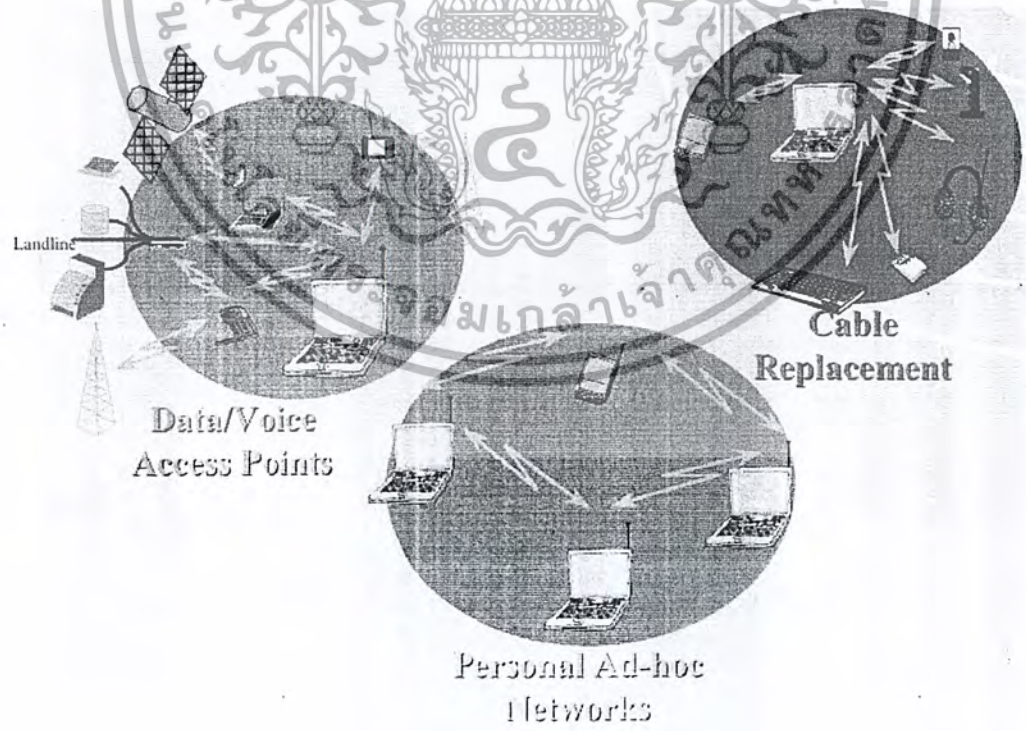
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

Bluetooth

Bluetooth เป็นระบบการสื่อสารแบบไร้สายที่ได้รับความนิยมมาก เมื่อ 3-4 ปีที่ผ่านมา โดยมีผู้ผลิตหลายรายต่างพยายามผลิตอุปกรณ์ขึ้นมาเพื่อยึดการตลาดของผลิตภัณฑ์นี้

Bluetooth เป็นเทคโนโลยีทางคลื่นวิทยุระยะทางพิสัยใกล้ที่มีราคาถูก กำลังงานต่ำ, ถูกออกแบบบนพื้นฐานเพื่อการทดแทนการเดินสายสัญญาณทองแดงขนาดสั้น เช่น ชุดทำการทดแทนการใช้แฮนด์ฟรี (Hand Free) แบบมีสายสำหรับโทรศัพท์มือถือ ชุดหูฟังสำหรับเครื่องเล่นเทปหรือ CD แบบพกพา รวมทั้งการใช้กับเครื่องพีซี เช่น การติดต่อสื่อสารระหว่างเครื่องพีซีกับเครื่องพิมพ์ เป็นต้น เมื่อมองดูผิวเผินจะเห็นได้ว่า Bluetooth ไม่มีพิษสงอะไรไม่ค่อยน่าสนใจเท่าใดนัก แต่เมื่อเปิดอุปกรณ์ที่เกี่ยวข้องกับ Bluetooth เพื่อใช้งานจะพบได้ทันทีว่ากำลังอยู่ในอ้อมกอดของพื้นที่การสื่อสารอันเป็นเครือข่ายส่วนตัว Personal Area Network (PAN) เครือข่ายที่มีขนาดพื้นที่จำกัด แต่จะอยู่ท่ามกลางของการแลกเปลี่ยนข่าวสารระหว่างอุปกรณ์ต่าง ๆ ที่เกี่ยวข้องกับ Bluetooth ทั้งหมด ดังแสดงในรูปที่ 3.1



รูปที่ 3.1 แสดงเครือข่ายส่วนตัวโดยใช้อุปกรณ์ Bluetooth

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเชื่อมต่อกันระหว่างอุปกรณ์ที่ทำงานภายใต้ Bluetooth เป็นการใช้คลื่นวิทยุสื่อสารกันเข้าอย่างกลมกลืนชนิดไร้รอยตะเข็บ ผู้ใช้งานไม่จำเป็นต้องทราบว่าโทรศัพท์มือถือวางอยู่ที่ใด ในห้องหรือ ณ ที่ใดภายในบ้านหรือในรถยนต์ เมื่อเปิด Nothbook ที่มี Web Browser ที่ติดตั้ง Web Browser อยู่เมื่อใดการสถาปนาการเชื่อมต่อจะเกิดขึ้นในทันที โดยทางโทรศัพท์เชื่อมต่ออยู่กับผู้ให้บริการอินเทอร์เน็ต ซึ่งจะทำให้ผู้ใช้สามารถท่องไปยังอินเทอร์เน็ต

3.1 การก่อกำเนิดของ Bluetooth

ในปี ค.ศ. 1999 ได้มีการกำหนดมาตรฐานของ Bluetooth Version 1.0 ขึ้น แต่ความเป็นมาของ Bluetooth มีมาก่อนหน้านี้ 5 ปี เมื่อแผนกสื่อสารแบบเคลื่อนที่ของ บริษัท อีริคสัน ได้ทำการศึกษาและทดสอบวิธีการทดแทนการใช้สายโทรศัพท์ Hand Free เพื่อที่จะเชื่อมโยงระหว่างอุปกรณ์โทรศัพท์ และอุปกรณ์พ่วง เช่น หูฟังโทรศัพท์ อุปกรณ์ไมโครโฟน และพบว่าการใช้คลื่นวิทยุเป็นวิธีการที่ดีที่สุด เนื่องจากสามารถครอบคลุมได้ทุกทิศทุกทางซึ่งจะสามารถลดข้อจำกัดของการใช้อินฟาเรด ซึ่งครอบคลุมพื้นที่อย่างจำกัดและมีใช้ก่อนหน้านี้แล้ว

รู้จักกับ Bluetooth SIG

SIG มาจากคำว่า **"Bluetooth Special Interest Group"** ประกอบขึ้นด้วยกลุ่มของบริษัท จำนวนหนึ่งที่ก่อตั้งมาตรฐาน Bluetooth ขึ้น ในปี ค.ศ. 1998 ประกอบด้วยบริษัทต่าง ๆ ดังนี้

- Ericsson Mobile Communication AB
- Intel Corp.
- Toshiba Corp.
- Nokia Mobile Group Phones

ในเดือนพฤษภาคม 1998 แกนนำของกลุ่มนี้ได้ประกาศตัวอย่างเป็นทางการและเชื้อเชิญให้บริษัทผู้ผลิตอื่น ๆ ได้เข้ามาร่วมกับ SIG เพื่อก่อตั้งคณะกรรมการกำหนดมาตรฐานของ Bluetooth ขึ้น จนได้มาตรฐานขึ้นและประกาศใช้เป็น Version 1.0 ในเดือนกรกฎาคม ปี 1999 และจัดตั้ง Website เพื่อเผยแพร่ต่อสาธารณชนที่ www.bluetooth.com ในเดือนธันวาคมได้มีบริษัทที่เข้าร่วมกับ SIG เพิ่มเติมได้แก่

- Microsoft
- Lucent

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 3Com
- Motorola

3.2 จุดมุ่งหมายของการใช้ Bluetooth

เป้าหมายการใช้งาน Bluetooth ถูกกำหนดไว้สูง เนื่องจาก Bluetooth ถูกออกแบบมาเพื่อรองรับเทคโนโลยีของอุปกรณ์คอมพิวเตอร์ที่เคลื่อนที่ได้ รวมทั้งอุปกรณ์การสื่อสารในทุกรูปแบบ ดังนั้นเป้าหมายของการใช้อยู่ที่

- การใช้งานได้ทั่วทุกแห่งหน ไม่จำกัดพื้นที่
- สนับสนุนการสื่อสารข้อมูลแบบ Data และ Voice
- มีขีดความสามารถในการสื่อสารแบบ Ad-hoc หรือการสื่อสารที่ไม่ต้องมีอุปกรณ์ Access Point เข้ามาเกี่ยวข้อง
- มีขีดความสามารถในการป้องกันการรบกวนจากสัญญาณต่าง ๆ โดยเฉพาะช่องสัญญาณวิทยุต่าง ๆ
- มีขนาดเล็กเพื่อให้สามารถติดตั้งฝังตัว หรือเชื่อมต่อกับอุปกรณ์คอมพิวเตอร์หรือสื่อสารได้โดยง่าย
- กินกระแสไฟที่ค่อนข้างต่ำมาก
- เป็นมาตรฐานเปิด
- มีต้นทุนที่ต่ำ เมื่อเทียบกับอุปกรณ์ที่พ่วงต่อด้วย หรือต่ำกว่าการใช้WLAN
- ทำให้สามารถโยกย้ายอุปกรณ์ไปได้ โดยรอบอาณาบริเวณที่ๆ สายสัญญาณไม่สามารถลากไปถึงได้
- สามารถควบคุมการเปิดปิด หรือใช้งานอุปกรณ์ต่างๆ ได้ โดยเพียงการเปิดสวิตซ์อุปกรณ์ใช้งานเพียงตัวเดียว

ระบบสายสัญญาณที่ Bluetooth ไม่สามารถทดแทนได้

Bluetooth ไม่ได้ถูกออกแบบมาเพื่อให้สามารถรองรับการสัญจรไปมาของข่าวสารที่มีปริมาณมาก หรือหนาแน่น ดังนั้นไม่สามารถนำมาใช้งานเพื่อการทดแทน Back Bone บน LAN และโดยธรรมชาติของมันทำให้ไม่สามารถใช้งานบนเครือข่ายที่ต้องมีเซิร์ฟเวอร์ แต่ Bluetooth ถูกออกแบบมาเพื่อให้รองรับการทำงานสำหรับคอมพิวเตอร์ที่ไม่ได้อยู่กับที่ แต่มีการเคลื่อนย้ายไปบ่อยๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

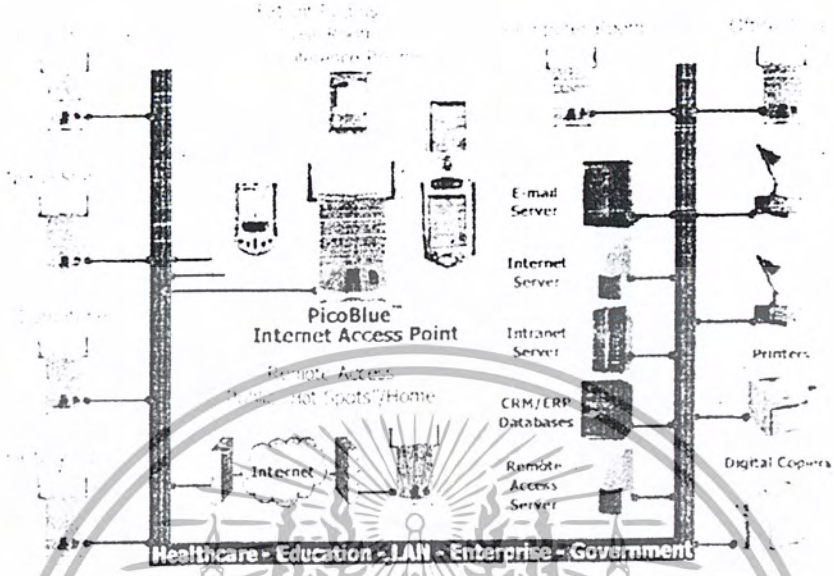
สามารถนำ Bluetooth มาใช้ร่วมกับอะไรบ้าง

ในทางปฏิบัติอุปกรณ์คอมพิวเตอร์ทั้งหมด ที่ติดตั้งในสำนักงานปัจจุบัน หรือที่บ้าน ที่ไม่ได้สื่อสารกันด้วยโปรโตคอลแบบ Synchronous ย่อมสามารถทำงานกับ Bluetooth ได้ทั้งสิ้นประกอบด้วย

- โทรศัพท์ และเพจเจอร์
- โมเด็ม
- อุปกรณ์ที่ใช้ Access เข้าสู่เครือข่าย LAN
- ชุดครอบศีรษะ สำหรับรับฟังเสียง หรือ Hand Set สำหรับโทรศัพท์มือถือ
- Notebook Computer
- เครื่องคอมพิวเตอร์แบบตั้งโต๊ะ
- เครื่องพิมพ์คอมพิวเตอร์
- เครื่องแฟ็กซ์
- Keyboard
- Joystick
- PDA
- MP3 Plaver
- Digital Camara
- Home Audio System
- Access Point

อาจกล่าวได้ว่า อุปกรณ์ดิจิตอลทั้งหลายนั้นต่างก็หันมาให้การสนับสนุนการทำงานของ Bluetooth แทบทั้งสิ้น เทคโนโลยีคลื่นวิทยุของ Bluetooth ยังสามารถเป็นระบบเชื่อมโยงแบบเอนกประสงค์ที่ใช้เป็นสะพานเชื่อมโยงกันระหว่างเครือข่ายข้อมูลที่มีอยู่ได้แล้ว รวมทั้งอุปกรณ์เชื่อมต่อคอมพิวเตอร์ต่างๆ อีก ทั้งยังสามารถก่อร่างขึ้นเป็นกลุ่มของเครือข่ายย่อย ๆ ที่ไม่มีเซิร์ฟเวอร์ แต่เป็นการเชื่อมต่อกันโดยตรง (ad-hoc) รวมทั้งยังสามารถก่อร่างขึ้นเป็นกลุ่มของอุปกรณ์คอมพิวเตอร์ ตัวคอมพิวเตอร์ และอุปกรณ์สื่อสารต่าง ๆ ที่อยู่ในรายล้อมนั้นด้วย Bluetooth สามารถสร้างอาณาจักรการเชื่อมต่ออันน่ามหัศจรรย์เล็กๆ ภายในบ้าน หรือที่ทำงาน เช่น สามารถสั่งให้เอาข้อมูลจาก เครื่องคอมพิวเตอร์ส่งไปพิมพ์ยังเครื่องพิมพ์ที่วาง ณ จุดใดจุดหนึ่งของห้องหรืออาจวางอยู่ห้องข้าง ๆ ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แสดงการก่อร่างสร้างอาณาจักรการสื่อสารย่อยๆในบ้าน หรือที่ทำงาน

ตัวอย่างการใช้งาน Bluetooth

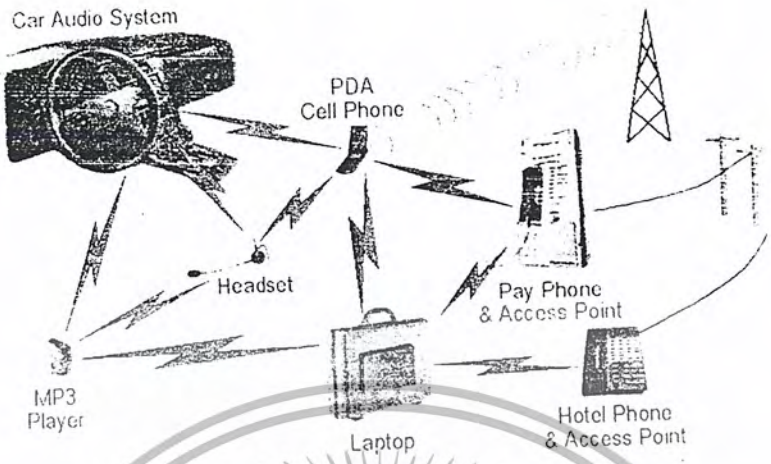
รูปที่ 3.3 แสดงลักษณะ และตัวอย่างการใช้งาน Bluetooth

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างการประยุกต์ใช้งาน Bluetooth มีดังต่อไปนี้

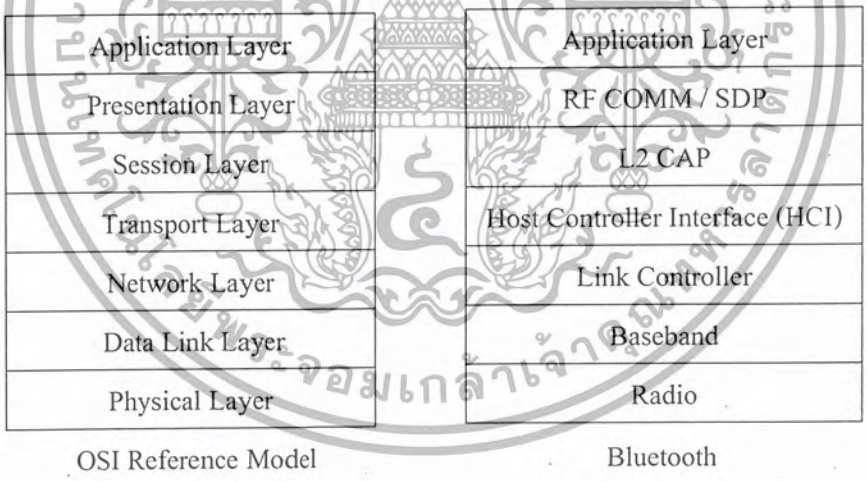
1. Mouse ที่ติดตั้ง Bluetooth สามารถเชื่อมต่อกับคอมพิวเตอร์ในรูปแบบไร้สายและอยู่ห่างออกไปจากเครื่องคอมพิวเตอร์และจอภาพโดยอาจอยู่ห่างกันได้หลายๆ เมตร
2. สามารถวาง Keyboard ที่สนับสนุน Bluetooth ให้อยู่ห่างจากเครื่องคอมพิวเตอร์ และจอภาพได้เพื่อให้ผู้ใช้งานที่มีปัญหาสายตายาวสามารถใช้งานได้สะดวกยิ่งขึ้น
3. Keyboard ที่สนับสนุนการทำงานของ Bluetooth ยังสามารถถูกใช้เพื่อเป็นสวิชท์ควบคุมการทำงานของคอมพิวเตอร์มากกว่าหนึ่งเครื่องได้ ในรูปแบบที่สามารถโยกไปมาระหว่างคอมพิวเตอร์ทั้งสองได้
4. สามารถเชื่อมต่อไปที่เครื่องพิมพ์ หรือเครื่องแฟกซ์ได้
5. เมื่อใดที่เครื่องคอมพิวเตอร์ซึ่งตั้งอยู่ภายในบ้าน หรือที่ทำงานเดียวกัน ได้รับ Email มันจะส่งสัญญาณเตือนให้รับมาทางโทรศัพท์มือถือ และตอบรับหรือดู Email ที่เข้ามาทางมือถือ ซึ่งส่งตรงมาจากเครื่องคอมพิวเตอร์ของเรา หมายความว่า เครื่องคอมพิวเตอร์จะส่งข่าวสารที่เกี่ยวกับการแจ้งเตือนว่า ได้รับ Email เข้ามาแล้ว และส่งมาให้ทางโทรศัพท์มือถือ
6. นักธุรกิจที่เดินทางไปยังที่ต่างๆ สามารถใช้เครื่องคอมพิวเตอร์ที่ติดตัวไปด้วย สามารถค้นหาสถานที่ติดตั้งเครื่องพิมพ์ที่สนับสนุนการทำงานของ Bluetooth ได้อย่างรวดเร็วหลังจากที่ได้เดินเข้ามาในห้องรับแขกของโรงแรม อีกทั้งยังสามารถส่งข่าวสารจากเครื่องพิมพ์ไปยังเครื่องพิมพ์ต่างๆ ที่เครื่องคอมพิวเตอร์ได้ค้นพบ
7. สามารถเชื่อมต่อระหว่าง กล้องดิจิตอลกับ เครื่องฉาย หรือ Projector โดยที่ไม่ต้องใช้สาย
8. สามารถเชื่อมต่อระหว่างโทรศัพท์มือถือกับชุด Hand Free
9. สามารถนำมาใช้สร้างเป็นระบบเครือข่ายไร้สายได้ หรือในการจัดงานตามสถานที่ต่างๆ ก็สามารถที่จะติดต่อสื่อสารได้ เพื่อช่วยความอำนวยความสะดวกให้แก่ผู้ร่วมงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 แสดงลักษณะการเชื่อมต่ออุปกรณ์อื่นๆ ภายใต้ Bluetooth

3.3 การทำงานของ Bluetooth กับ OSI Model (OSI : Open Systems Interconnection)



รูปที่ 3.5 แสดงการเปรียบเทียบระดับชั้นการทำงานของ Bluetooth กับ OSI Model

แม้ว่า Bluetooth ไม่ได้ทำงานบนพื้นฐานของ OSI Model แต่มันมีระดับชั้นการทำงานของมันเองอย่างไรก็ดี พยายามที่จะเปรียบเทียบให้เห็นว่าระดับชั้นที่แปลกแยกของ Bluetooth มีการทำงานอย่างไร เช่น ระดับชั้น Physical ของ OSI Model เป็นระดับชั้น Radio ของ Bluetooth ซึ่งจะเห็นได้ว่าในระดับชั้น Physical ของ OSI Model เป็นระดับชั้นของการอธิบายคุณสมบัติของสายสัญญาณแบบต่างๆ รวมทั้งอธิบายคุณสมบัติทางไฟฟ้าของสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่วิ่งบนสายสัญญาณ รวมทั้งวิธีการเข้ารหัสสัญญาณ เพื่อให้การสื่อสัญญาณมีความเหมาะสมกับสายสัญญาณที่ใช้ ส่วน ระดับชั้น Radio ของ Bluetooth ก็เช่นเดียวกัน ประกอบด้วยวิธีในการเข้ารหัสข่าวสารภายในคลื่นวิทยุ รวมทั้งวิธีการผสมสัญญาณ

ระดับชั้น Physical Layer ของ OSI Model ยังครอบคลุมบางส่วนของ Base Band บน Bluetooth อีกด้วย

ระดับชั้น Data Link รับผิดชอบอธิบายการทำงานใดๆ ที่เกี่ยวกับการสร้างเฟรมของข้อมูล วิธีการแพร่ข่าวสาร รวมทั้งการควบคุมความผิดพลาด ขณะที่มีการเชื่อมต่อระหว่างกัน ระดับชั้น Data Link นี้ ยังครอบคลุมบางส่วนของ Link Controller ของ Bluetooth และบางส่วนของ Base Band ทำให้มีการตรวจสอบความผิดพลาด และแก้ไขความผิดพลาดข้อมูลข่าวสารได้

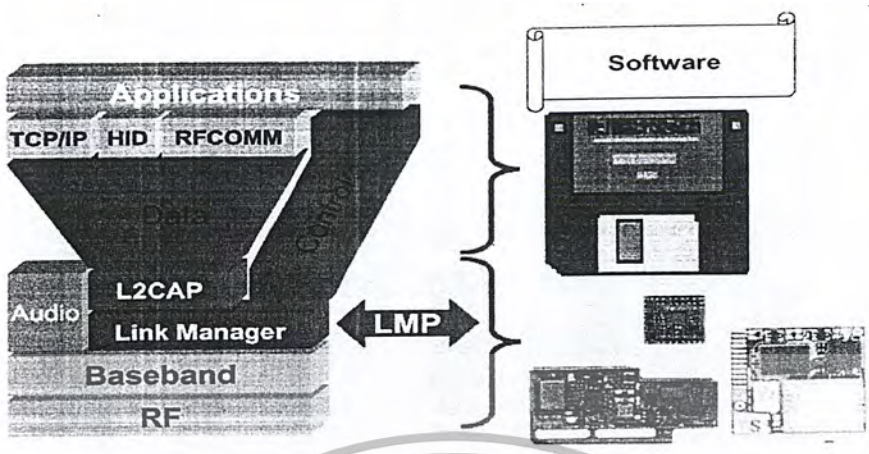
ระดับชั้น Network ของ OSI Model ตรงกับระดับชั้น Link Controller และบางส่วนของ Link Manager (LM) ของ Bluetooth หน้าที่หลักของระดับชั้นนี้ ได้แก่ การแพร่ข้อมูลข่าวสารข้ามเครือข่าย โดยจะไม่นำพาต่อประเภทของสื่อสัญญาณหรือ รูปแบบการเชื่อมต่อโดยมี Link Controller และ Link Manager (LM) ซึ่งทำหน้าที่ ดูแลการสถาปนาการเชื่อมต่อ ระหว่างจุด หรือหลายๆ จุดเกิดขึ้นได้พร้อมกัน

ระดับชั้น Transport ทำหน้าที่เกี่ยวกับการเชื่อมต่อและการสื่อสารที่มีความน่าเชื่อถือ รวมทั้งการสื่อสารข้อมูลข่าวสารหลายประเภทแบบสลับกันไปมา (Multiplexing) และใช้กลไกการขนส่งข้อมูลข่าวสารที่เรียกว่า **"Host Controller Interface หรือ HCI"** ควบคุมกรรมวิธีการรับส่งข้อมูลอย่างมีเสถียรภาพ

ระดับชั้น Session รับผิดชอบดูแลเกี่ยวกับการบริหารจัดการกับการไหลของข้อมูลข่าวสารซึ่งทำงานโดย L2CAP และบางส่วนของ RFCOMM / SDP

ระดับชั้น Presentation ดูแลเกี่ยวกับการสื่อความหมายของ Application โดยการสอดแทรกโครงสร้างพื้นฐานของการให้บริการบนข้อมูลข่าวสารที่จะส่งออกไป และที่สุด คือ ระดับชั้น Application ดูแลเกี่ยวกับการบริหารจัดการกับการสื่อสารระหว่าง คอมพิวเตอร์ผ่านทาง Application

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



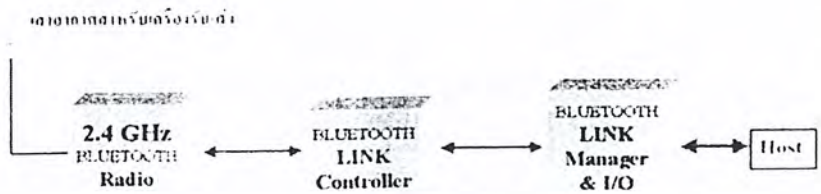
รูปที่ 3.6 แสดงการเปรียบเทียบระดับชั้นโพรโตคอล Bluetooth กับ อุปกรณ์จริง

ความสำคัญของระดับชั้น Physical สำหรับ Bluetooth

อุปกรณ์ Bluetooth ทำงานที่ความถี่ 2.4 GHz โดยที่ความถี่นี้สามารถใช้งานได้ทั่วโลก ความถี่นี้เป็นแบบ License-Free ในย่านความถี่ ISM (Industrial Scientific Medical) ซึ่งย่านความถี่นี้เป็นย่านความถี่ที่สงวนไว้ให้กับวงการอุตสาหกรรม วงการวิทยาศาสตร์ และวงการแพทย์ (Industrial Scientific Medical) และเนื่องจากเป็นความถี่สงวน ดังนั้น จึงไม่มีปัญหาถูกรบกวนจากช่องสัญญาณความถี่อื่น ๆ

แถบความถี่ปฏิบัติงาน ถูกแบ่งออกเป็นช่องสัญญาณที่มีความห่างกัน 1 MHz แต่ละช่องสัญญาณสื่อสารสามารถจัดส่งข้อมูลได้มากถึง 1 Mega symbol ต่อวินาที ด้วยวิธีนี้จะทำให้ Bluetooth สามารถใช้ช่องสัญญาณที่มีอยู่ทั้งหมดได้เต็มที่ รูปแบบการผสมสัญญาณ ได้แก่ GFSK (Guassian Frequency Shift Keying) หลังจากการส่งข้อมูลใน Packet แรกไปแล้ว ผู้ส่งจะเปลี่ยนช่องความถี่อื่น โดยการกระโดดจากความถี่หนึ่งไปยังอีกความถี่หนึ่ง เพื่อส่ง Packet ต่อไป เราเรียกวิธีการนี้ว่า " **FHSS หรือ Frequency Hopping Spread Spectrum**"

การทำงานขั้นพื้นฐานของ Bluetooth



รูปที่ 3.7 แสดงลักษณะการทำงานขั้นพื้นฐานของ Bluetooth

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 มาตรฐาน IEEE 802.11

ในปี 1997 IEEE หรือ Institute of Electrical and Electronics Engineering ซึ่งเป็นสถาบันที่กำหนดมาตรฐานการทำงานของระบบเครือข่าย ได้ทำการกำหนดมาตรฐานของคลื่นความถี่ และการทำงานของคลื่นวิทยุสำหรับเครือข่ายไร้สายเฉพาะโดยนับตั้งแต่นั้นมาจนถึงปัจจุบัน IEEE ได้กำหนดมาตรฐานออกมาดังนี้

- มาตรฐาน IEEE 802.11a
- มาตรฐาน IEEE 802.11 b
- มาตรฐาน IEEE 802.11 e
- มาตรฐาน IEEE 802.11 g

- **มาตรฐาน IEEE 802.11 a**

IEEE 802.11 a เป็นมาตรฐานของระบบเครือข่ายไร้สายประสิทธิภาพสูงทำงานที่ย่านความถี่ 5 GHz (ชนิดไม่มีลิขสิทธิ์) มีความเร็วในการรับส่งข้อมูลที่ 54 Mbps โดยใช้ความถี่คลื่นวิทยุ หรืออุปกรณ์เชิงกายภาพ (Physical) ที่ทำงานบนพื้นฐานของการผสมสัญญาณแบบ Orthogonal Frequency Division Multiplexing (OFDM) รวมทั้งอุปกรณ์ควบคุมการ Access เข้าสู่ เครือข่ายที่ทำงานบนมาตรฐาน IEEE 802.11

คุณประโยชน์ของ IEEE 802.11 a

- ความเร็ว : มีประสิทธิภาพการส่งข้อมูลความเร็วสูง (54 Mbps) ความเร็วขนาดนี้เหมาะสำหรับการแพร่ภาพและข่าวสารที่ต้องการความละเอียดและความเร็วสูง อย่างเช่น HDTV รวมทั้งการแพร่กระแสของภาพ Video แบบ MPEG-2 (คุณภาพระดับ DVD)
- สัญญาณคลื่นรบกวน : เนื่องจากความถี่ขนาด 5 GHz ยังไม่ได้ถูกแพร่หลายมาก ดังนั้นปัญหาการรบกวนกันของคลื่นความถี่จะน้อยกว่าระบบเครือข่ายไร้สายที่ใช้คลื่นความถี่ 2.4 GHz ซึ่งมีใช้อย่างแพร่หลายมากกว่า
- มีการเข้ารหัสแบบ WEB ขนาด 128 บิต
- ระยะทางของการเชื่อมต่อสื่อสารอยู่ที่ 300 ฟิตจากจุด Access Point
- 802.11a สนับสนุนการเชื่อมต่อของ Access Point แบบพ่วงต่อในรูปแบบของ Roaming ได้มากถึง 8 จุดลักษณะนี้ทำให้สามารถครอบคลุมพื้นที่การทำงานที่กว้างขวางมากขึ้น อีกทั้งสามารถเพิ่มจำนวนของผู้ใช้งานในเครือข่ายได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จุดด้อยของ IEEE 802.11a มีดังนี้

- ค่าใช้จ่าย : เป็นระบบที่มีค่าใช้จ่ายสูงกว่า IEEE 802.11b
- ระยะทางการเชื่อมต่อ : มีระยะการเชื่อมต่อที่น้อยกว่า 802.11b
- การสนับสนุนระบบ Multimedia และคุณภาพในการให้บริการ : มาตรฐาน Mac ของ IEEE 802.11 เป็นมาตรฐานเดียวกับ Ethernet และสนับสนุน Application แบบ Multimedia ที่มีการแพร่ข่าวสารเกี่ยวกับภาพและเสียงอย่างต่อเนื่อง หรือพร้อมกันๆ ในเวลาเดียวกันได้ไม่ดีนัก นอกจากนี้ยังมีปัญหาเรื่องความเข้ากันได้ของผลิตภัณฑ์ที่ได้มาตรฐาน IEEE 802.11 bที่กำลังมาแรงในขณะนี้

- **มาตรฐาน IEEE 802.11 b**

เป็นมาตรฐาน WLAN ของ IEEE ทำงานที่ความถี่ 2.4 GHz (ชนิดไม่มีลิขสิทธ์) สามารถส่งถ่ายข้อมูลข่าวสารที่ความเร็ว 11 1Mbps มีการใช้ Physical ที่ทำงานที่ความถี่ผสมสัญญาณแบบ Direct Sequence Sprcad Spectrum (DSSS) ส่วน MAC ของมันทำงานบนสถาปัตยกรรมมาตรฐาน 802.11 ซึ่งวิธีการ Direct Sequence Spread Spectrum (DSSS) เป็น Spread Spectrum และแพร่หลายมากที่สุด โดยมีการกำหนดให้สัญญาณข้อมูลถูกทวีคูณด้วยรหัสที่เรียกว่า **"Pseudo–Ramdom Noise (PN)"** ที่มีอัตราค่าของ Chip ที่สูง (องค์ประกอบของการทำงานแบบลำดับชั้นของ PN ถูกเรียกว่า **"Chip"**) การใช้ DSSS หมายความว่า สัญญาณจะต้องถูกขยายออก นั่นคือ กำลังงานของสัญญาณจะถูกขยายออกไปตามช่องของสัญญาณย่านความถี่มากที่สุด และหากช่องสัญญาณความถี่ใดมีปัญหาความถี่ลดลงก็ไม่มีผล เนื่องจากยังมีกำลังงานของสัญญาณส่วนหนึ่งสามารถส่งออกไปที่ปลายทางที่ช่องสัญญาณหนึ่งได้ ห้วงเวลาของลำดับของสัญญาณ PN มีความยาว 1 บิต และค่ารหัสของ chip มีดังนี้

ค่ารหัส Chip : 0 = 11101100011

: 1 = 00010011100

ตัวอย่าง หากต้องการส่งสัญญาณข้อมูล 1 0 1 ค่าของChip จะเป็นดังนี้

00010011100	11101100011	000100011100
1	0	1

ตาราง 3.1 แสดงลักษณะของ Chip ที่เป็นรูปแบบของสัญญาณที่แทนด้วยเลข binary

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติของ IEEE 802.11b

เป็นมาตรฐานที่กำลังได้รับความนิยมเป็นอย่างมากในขณะนี้ เนื่องจากขณะนี้มีผู้ผลิตอุปกรณ์เครือข่ายไร้สายภายใต้มาตรฐานนี้ออกมามากมาย และที่สำคัญคือ ความที่สามารถเข้ากันได้ดีอีกทั้งปัจจุบันมีการติดตั้งเครือข่ายไร้สายที่ต้องใช้ความถี่นี้ ทั้งในองค์กรธุรกิจและสถานที่สาธารณะทั่วไป (ในต่างประเทศ) และกำลังจะแพร่เข้าสู่สถานที่พักอาศัย

มาตรฐาน 802.11b เปิดช่องให้สามารถเลือกวิธีการเข้ารหัสข้อมูลข่าวสารประเภทแบ่งใช้งานกุญแจที่ผ่านการเข้ารหัสแบบ RC-4 นอกจากนี้มีผู้ผลิตบางรายได้เพิ่มวิธีการควบคุมการเข้าไปใช้งานเครือข่าย โดยมีการตรวจสอบสิทธิ์ของผู้ใช้งานระหว่างสถานีลูกข่ายกับ Access point ก่อนที่จะอนุญาตให้เข้าสู่เครือข่ายได้ นอกจากจะมีน่าเชื่อถือ และมีระบบรักษาความปลอดภัยที่เพียงพอแล้ว ยังสามารถทำงานแบบที่เกี่ยวข้องกับพลังงานได้ 2 แบบ คือ

- โหมดการทำงานแบบ Continuous Aware Mode โดยโหมดนี้ คลื่นวิทยุจะมีการกินกระแสอย่างเนื่องตราบใดที่วิทยุยังคงทำงานอยู่
- โหมดการทำงานแบบ Power Saving Polling Mode ระบบนี้จะไม่กินกระแสไฟ หากไม่มีผู้งานบนเครือข่าย โดยเข้าสู่สถานะที่เรียกว่า "**Sleep**" เพื่อยืดอายุการรมใช้งานของแบตเตอรี่ของอุปกรณ์ที่เป็นแบบเคลื่อนย้ายได้ เช่น Note Book และมี Access Point ทำหน้าที่เก็บข้อมูลไว้ที่ Buffer เสียก่อนที่จะส่งสัญญาณออกไป ไคลเอนต์ เพื่อบอกให้เตรียมตัวรับข้อมูลได้

- **มาตรฐาน IEEE 802.11e**

เป็นมาตรฐานที่เพิ่มขีดความสามารถให้กับ MAC ในอันที่จะสนับสนุนการทำงานของ Multimedia และ QoS และสามารถทำงานร่วมกับ Physical ของ 802.11a และ 802.11b รวมทั้ง 802.11 g เหมาะสำหรับงานด้านธุรกิจ และหน่วยงานที่ต้องการประสิทธิ์ภาพการส่งถ่ายข้อมูลในรูปแบบ Voice Data Vedio

- **มาตรฐาน IEEE 802.11 g**

มีการทำงานในระดับ Physical สนับสนุนความถี่ 2.4 GHz มี MAC ทำงานตามมาตรฐาน IEEE 802.11 ทุกประการ ดังนั้นสามารถเชื่อมต่อกับสถานีเครือข่ายที่เป็นมาตรฐาน IEEE 802.11b ได้

โดยมาตรฐานของ IEEE 802.11 ได้กำหนดโปรโตคอลการทำงานให้กับเครือข่ายไร้ 2 แบบ ซึ่งได้แก่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **แบบ Ad - hoc**

เครือข่ายแบบ Ad - hoc เป็นเครือข่ายแบบเรียบง่ายที่การสื่อสารถูกก่อตั้งขึ้นโดยระหว่างคอมพิวเตอร์ต่างๆ ในพื้นที่เดียวกัน โดยไม่ต้องใช้ Access Point หรือ Server ใดๆ ทั้งสิ้น มาตรฐานของ IEEE 802.11 นี้ ได้กำหนดให้เป็นหน้าที่ของสถานีเครือข่ายต่างๆ จะต้องคอยสอดส่องซึ่งกัน และกัน เพื่อที่จะได้ค้นพบกัน และกัน และเมื่อพบกันแล้ว ก็จะใช้เส้นทางที่ค้นพบระหว่างกันนั้นทำการสื่อสารติดต่อกันทางไร้สาย นอกจากนี้ IEEE ยังกำหนดให้สถานีเครือข่ายต่างๆ จะต้องมีการจัดส่งการร้องขอเพื่อให้มีหลีกเลี่ยงการกระทบกระทั่งระหว่างกัน (CSMA / CA) เพื่อให้สามารถใช้ช่องสัญญาณได้อย่างเต็มที่

- **แบบ Client – Server**

เครือข่ายแบบ Client / Server เป็นเครือข่ายที่ต้องการใช้ Access Point ที่สามารถควบคุมการสื่อสารระหว่างสถานีเครือข่าย แบบไร้สายกับเครือข่าย LAN รวมทั้งทำหน้าที่เป็นผู้ควบคุมการจราจรของเครือข่าย อีกทั้งยังสามารถต่อยอดการเชื่อมต่อกับเครื่องคอมพิวเตอร์ที่อยู่นอกรัศมีการเชื่อมต่อ ซึ่งเราเรียกว่าการ **"Roaming"**

Bluetooth เป็นเทคโนโลยีที่ใช้รูปแบบการเชื่อมต่อแบบ Ad-hoc ซึ่งหมายความว่าอุปกรณ์คอมพิวเตอร์ หรือสื่อสาร จะต้องสังเกตกันและกัน จากนั้นจะถ้อยที่ถ้อยอาศัยในการสื่อสาร บนเครือข่ายเดียวกัน จากรูปที่ 3.7 แสดงให้เห็นว่า บล็อกไดอาแกรมการเชื่อมต่อของ Blue tooth ประกอบด้วยส่วนที่เป็นวงจรเครื่องรับส่งวิทยุขนาดเล็ก และสามารถมีขนาดเล็กเท่ากับหัวนิ้วมือ รวมทั้งเสาอากาศที่มีขนาดเล็กมากจนสามารถซ่อนอยู่ในอุปกรณ์เล็กๆได้ นอกจากเสาอากาศขนาดเล็ก พร้อมอุปกรณ์รับส่งสัญญาณขนาด 2.4 GHz แล้ว ยังต้องพึ่งพาอาศัยโปรโตคอลอันเป็นแกนหลักในการทำงาน ได้แก่

- Logical Link Control และ Adaptation Protocol (L2CAP)
- Service Discovery Protocol (SDP)
- RFCOMM Protocol

L2CAP

หน้าที่หลักของ L2CAP ได้แก่ การทำงานร่วมกันกับโปรโตคอลระดับสูงบนการสื่อสารแบบ Baseband (การสื่อสารที่อยู่บนพื้นฐานข้อมูลดิจิตอลเป็นหลัก) L2CAP ยังทำหน้าที่ดูแลการให้บริการข้อมูลข่าวสารที่มาจากระดับบน (Application) และสามารถให้บริการขนส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลแก่ Application ต่าง ๆ ในรูปแบบการสลับการทำงาน (Multiplexing) รวมทั้งยังสามารถปรับขนาดของข้อมูลข่าวสารให้มีขนาดที่เหมาะสม (Segmentation) เพื่อให้สามารถส่งออกไปกับสัญญาณวิทยุอย่างมีประสิทธิภาพ และเมื่อข่าวสารไปถึงเครื่องรับปลายทาง L2CAP จะต้องทำหน้าที่รวมข่าวสาร (Reassembly) ที่ผ่านการปรับแต่งให้มีขนาดที่พอเหมาะจากเครื่องส่งเสียก่อนที่จะจัดส่งขึ้นไปที่ Application ต่อไป

SDP

ข้อมูลข่าวสารที่เกี่ยวกับอุปกรณ์ การให้บริการ และลักษณะพิเศษของการให้บริการสามารถตรวจสอบได้จาก SDP

RFCOMM

เป็นโปรโตคอลระดับบน ที่อยู่เหนือ L2CAP เป็นโปรโตคอลที่ให้บริการขนถ่ายแก่บริการในระดับสูง (เช่น OBEX Protocol) ที่ใช้สาย Serial เป็นกลไกในการขนถ่ายข้อมูล หรือพูดง่าย ๆ ก็คือ RFCOMM มีรูปแบบการขนถ่ายข้อมูลแบบ Serial Port หรือ RS-232

ความสัมพันธ์ในลักษณะการเชื่อมต่อ Bluetooth ที่มีต่อความว่างเปล่าของอากาศ อยู่บนพื้นฐานของการใช้กำลังส่งของเสาอากาศขนาดกำลัง 0 dBm และมีการเพิ่มเติมระบบการทำงานของ Spread Spectrum เข้าไป เพื่อให้สามารถเอื้ออำนวยความสะดวกแก่การใช้บริการเสริมที่มีระดับกำลังส่งถึง 100 mW ได้ทั่วโลก ส่วนจะทำได้หรือไม่อยู่ที่การใช้เทคนิคที่เรียกว่า **"Frequency Hopping"** โดยมีการกระโดดถึง 79 Hop ต่อความถี่ 1 MHz โดยเริ่มต้นที่ความถี่ 2.402 GHz และสิ้นสุดที่ 2.480 GHz และเนื่องจากข้อจำกัดทางกฎหมายสื่อสารของแต่ละประเทศได้มีการลด Bandwidth ในบางประเทศ เช่น ญี่ปุ่น ฝรั่งเศส และ สเปน ซึ่งทำให้มีจำนวนครั้งของการกระโดดของช่องสัญญาณความถี่ได้ถึง 1600 Hop ต่อวินาที ระยะทางการเชื่อมต่อ เริ่มตั้งแต่ 10 เซนติเมตร ไปจนถึง 10 เมตร และยังสามารถยืดระยะทางการเชื่อมต่อออกไปได้มากกว่า 100 เมตร โดยการเพิ่มกำลังการส่งสัญญาณคลื่นวิทยุ

ช่องสัญญาณของ Bluetooth

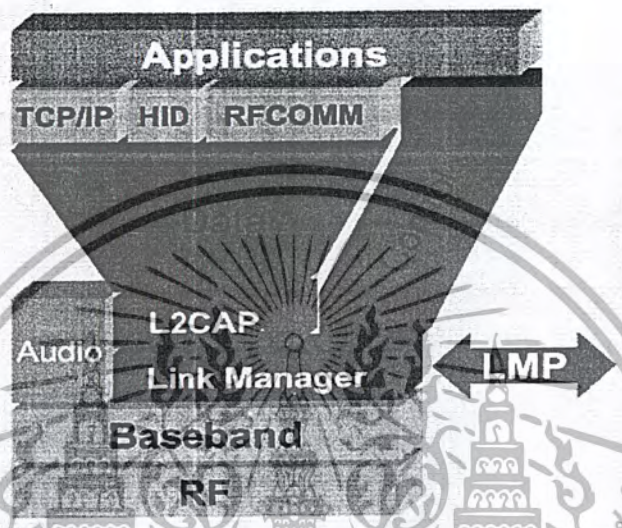
คำว่าช่องสัญญาณในที่นี้ หมายถึง สามสิ่งที่แตกต่างกัน ได้แก่

1. ช่องสัญญาณวิทยุจำนวน 79 (หรือ 23) ช่อง โดยแต่ละช่องมีการคั่นช่องสัญญาณความถี่ 1 MHz
2. หมายถึงช่องสัญญาณสื่อสาร ที่ประกอบด้วยการกระโดดไปมาของช่องสัญญาณทั้ง 79 (หรือ 23) ช่อง โดยช่องสัญญาณเหล่านี้ทำงานเหมือนกับหน้าที่การทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของเครื่องคอมพิวเตอร์บนระดับชั้น Session ของ OSI Model

3. ช่องสัญญาณทางตรรก (Logical Channel) 5 ช่องทาง มีไว้เพื่อใช้ควบคุมการทำงานของระบบ



รูปที่ 3.8 แสดงระดับชั้นการทำงานโปรโตคอลภายในมาตรฐานของ Bluetooth

3.5 รูปแบบการเชื่อมต่อกายภาพของ Bluetooth

รูปแบบการเชื่อมต่อทางกายภาพของ Bluetooth ได้ถูกนิยาม จัดตั้งขึ้น 2 แบบ ดังนี้

- **SCO** (Synchronous Connection-Oriented)

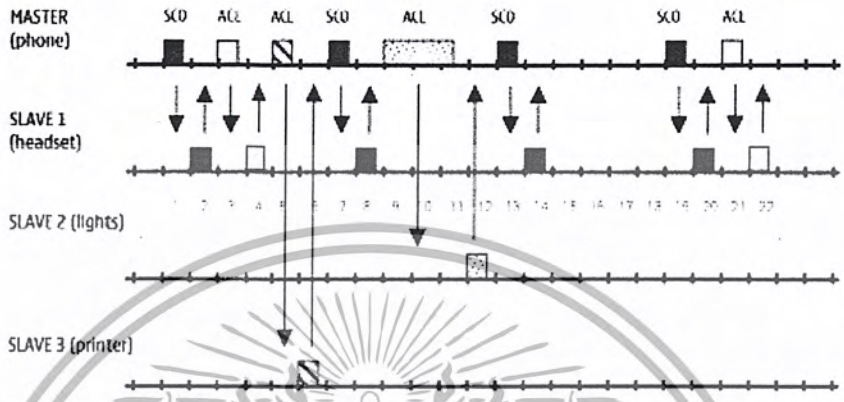
SCO เป็นการเชื่อมต่อแบบ Point To Point ระหว่างเครื่องที่เป็น Master กับ Slave โดยที่ตัว Master จะดูแลการเชื่อมต่อโดยทำการสำรอง หรือจองช่องเวลา (Time Slot) ไว้ให้ เมื่อจะมีการส่งข้อมูล การจับจองช่องเวลานี้ กระทำกันเป็นห้วง ๆ ของเวลาปกติ นอกจากนี้การส่งข้อมูลซ้ำไม่สามารถทำได้ภายใต้การเชื่อมต่อในลักษณะนี้

- **ACL** (Asynchronous Connection-Less)

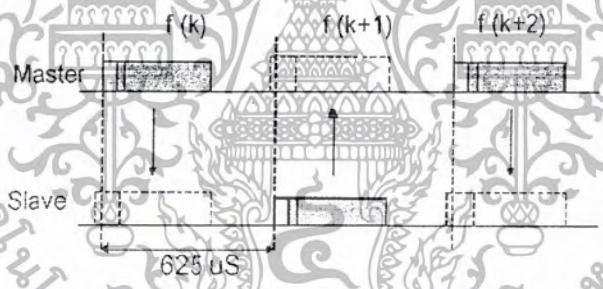
ACL เป็นการเชื่อมต่อที่ทำให้สามารถที่จะมีการสื่อสารกันแบบ Packet Switching ระหว่าง Master กับเครื่องเครือข่ายที่เป็น Slave ทั้งหมดในระบบการเชื่อมต่อนี้ สามารถส่งข้อมูลซ้ำได้เพื่อเป็นหลักประกันความถูกต้อง และความปลอดภัยของข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

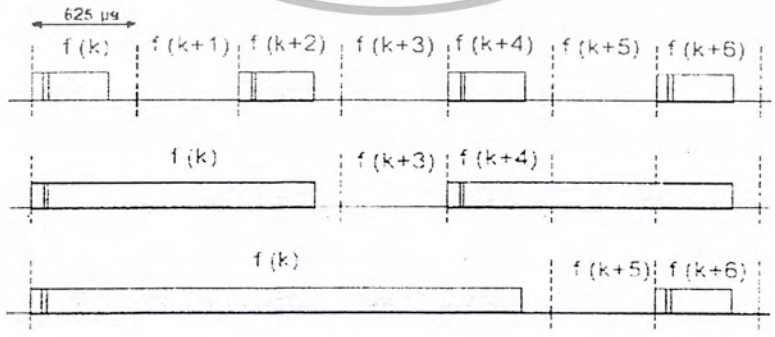
ในรูปแบบการเชื่อมต่อแต่ละแบบที่กล่าวมาถึงนี้ มีการใช้รูปแบบของ Packet ถึง 12 แบบ ประกอบด้วย 4 Packet ที่ใช้ควบคุมการเชื่อมต่อทั้ง 2 รูปแบบนี้



รูปที่ 3.9 แสดงลักษณะการเชื่อมต่อระหว่าง Master กับ Slave



รูปที่ 3.10 แสดงลักษณะของ Time slot ที่ทำงานแบบ Master และ Slave



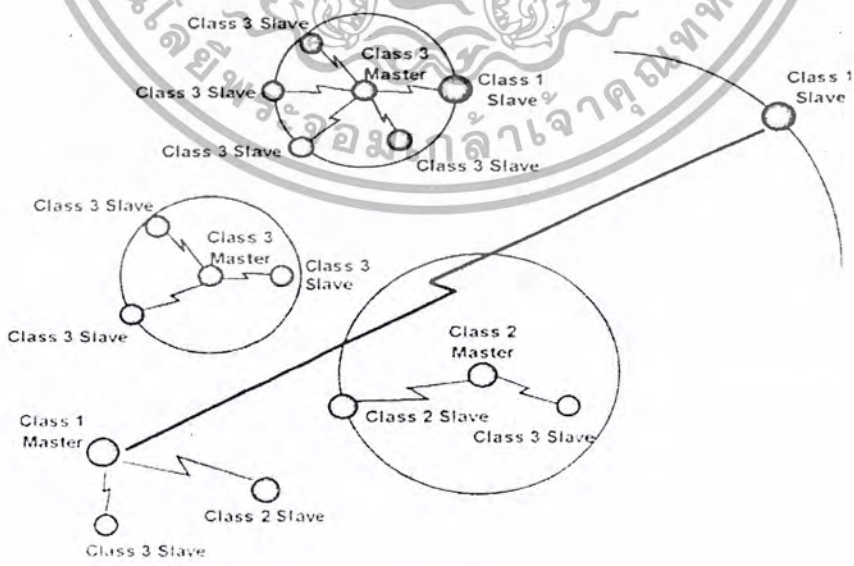
รูปที่ 3.11 แสดงขนาด Time Slot ที่มีขนาดใหญ่ และเล็กได้ ขึ้นอยู่กับรูปแบบการเชื่อมต่อทางกายภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Bluetoothใช้งาน Time Slot ในรูปแบบของ Frequency Hopping เนื่องจาก Bluetooth ถูกออกแบบมาเพื่อให้สามารถทำงานในสิ่งแวดล้อมของการกวนกันของคลื่นความถี่วิทยุ และมีการใช้การวิธีการตอบรับการติดต่อการส่งหรือรับข่าวสารที่รวดเร็ว

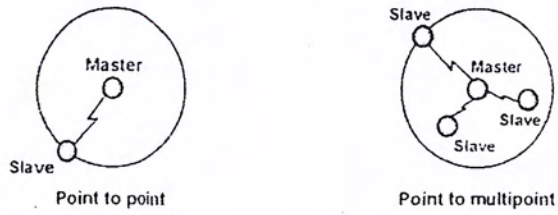
หากเปรียบเทียบกับระบบอื่น ที่อยู่ในย่านความถี่เดียวกันการกระโดดของ Bluetooth สามารถทำได้เร็วกว่า อีกทั้งมีการใช้ Packet ที่สั้นกว่า เนื่องจากการใช้ Packet ที่มีขนาดสั้นและการกระโดดที่รวดเร็ว ทำให้การทำงานมีความปลอดภัย และเสถียร สามารถรอดพ้นต่อภัยคุกคามของเครื่องใช้ไฟฟ้าความถี่สูงอย่างเตาอบไมโครเวฟ รวมทั้งแหล่งแพร่สัญญาณรบกวนอื่นๆ นอกจากนี้การใช้ระบบ Forward Error Correction (FEC) ทำให้สามารถลดปัญหาที่เกิดจากผลกระทบของสัญญาณรบกวนแบบเอาแน่นอนไม่ได้ที่มาจากระยะทางไกลออกไป

ถึงแม้ว่า Bluetooth มีการทำงานแบบ Ac-hoc ก็ตาม แต่การสื่อสารทั้งหมดจะต้องกระทำผ่าน ส่วนที่เรียกว่า **"Master"** โดยสถานี Slave ไม่สามารถสื่อสารกันเองได้ โดยปราศจาก Master แต่ก็ไม่ได้หมายความว่าเครื่องลูกข่ายด้วยกันจะไม่สามารถสื่อสารกันเองได้โดยอิสระเพียงแต่ว่าเครื่อง Slave ทั้งหลาย หากต้องการจะสื่อสารระหว่างกันมันจะต้องก่อตั้งเครือข่ายเล็กๆ ที่เรียกว่า**"Piconet"**โดยภายใต้ Piconet นี้ เครื่องที่เป็น Slave จะต้องกำหนดตัวเองขึ้นเป็น Master สักตัวหนึ่ง ลักษณะนี้จะเห็นได้ว่า การเปลี่ยนแปลง Configuration หรือรูปแบบการเชื่อมต่อของ Bluetooth สามารถกระทำได้แบบพลวัตร หรือยืดหยุ่นได้ อีกทั้งทำได้อย่างรวดเร็ว



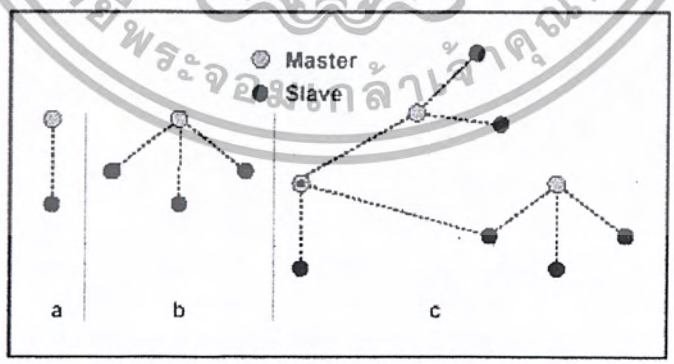
รูปที่ 3.12 แสดงการทำงานของ Master กับ Slave

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



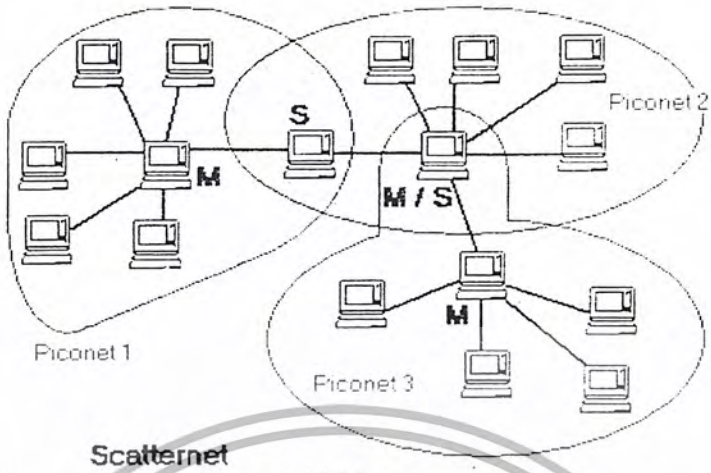
รูปที่ 3.13 แสดงผังการเชื่อมต่อระหว่าง Master และ Slave บนความถี่ระดับClassต่างๆ

รูปที่ 3.14 แสดงรูปแบบการเชื่อมต่อแบบจุดต่อจุด และแบบหนึ่งจุดกับหลายจุด



รูปที่ 3.15 แสดงรูปแบบการเชื่อมต่อระหว่างกลุ่มของ Slave และ Master

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.16 แสดงรูปแบบ Piconet ระหว่างเครื่อง Slave

ระดับกำลังงานของการเชื่อมต่อ

ศักยภาพทางคลื่นวิทยุของ Bluetooth ถูกสร้างให้มีขนาดที่เล็ก ขนาดเท่าไมโครชิป และทำงานบนย่านความถี่ มาตรฐานที่สำรองไว้ให้ใช้ทั่วโลกได้ เพื่อให้สามารถสื่อสารระหว่างกัน มีความเข้ากันได้ดี ดังนั้นจึงได้กำหนดให้มีระดับกำลังส่ง 2 ระดับ ได้แก่

1. ระดับกำลังส่งที่ต่ำที่ครอบคลุมพื้นที่ขนาดเล็ก เหมาะสำหรับเป็นเครือข่ายส่วนตัว
2. ระดับกำลังส่งสูง ที่ครอบคลุม ระยะทางขนาดปานกลาง อย่างเช่น ระยะทางที่ครอบคลุมรัศมีภายในบ้าน

โปรโตคอลการทำงานแบบ Baseband ของ Bluetooth เป็นการทำงานอย่างผสมผสานกัน ระหว่างวงจรทาง Hardware และ Packet Switch การกำหนดขนาดของ Time Slot ขึ้นอยู่กับการส่งข้อมูลบนช่องสัญญาณแบบ Synchronous โดยหนึ่ง Packet ของข้อมูลอาจใช้หนึ่ง Time Slot แต่ก็สามารถยืดหยุ่นออกจนสามารถครอบคลุมได้มากถึง 5 Slot ต่อหนึ่ง Packet

Bluetooth สามารถให้การสนับสนุนช่องสัญญาณแบบใด

- ช่องสัญญาณข้อมูลแบบ Asynchronous
- 3 ช่องสัญญาณเสียงที่ทำงานแบบ Synchronous พร้อมกัน
- ช่องสัญญาณหนึ่งช่องที่สามารถสนับสนุนข่าวสารแบบ Asynchronous และ Voice แบบ Synchronous พร้อมกันทีเดียวทั้งสองช่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เปรียบเทียบการทำงานของ Bluetooth กับระบบ IEEE 802.11

ขอบข่ายการทำงาน	Bluetooth	802.11
อัตราการส่งของสัญญาลักษณ์	1 Mbps	เชื่อมต่อจุดต่อจุด
Spread Spectrum	FHSS	DSSS
รูปแบบการใช้งาน	แทบไม่จำกัดรูปแบบ	จำกัดการใช้งานเฉพาะ LAN Station หรือ access Point
ลักษณะการแพร่กระจายของข่าวสาร	ไม่มีข้อห้าม	เฉพาะ Access point เท่านั้น
อัตราการกินกระแสไฟ	60 mA	300 mA
Audio	ช่องสัญญาณ PCM	สัญญาณ voice ภายใต้ 802.3
ใช้เพื่อแทนที่สายสัญญาณ	สาย Serial USB UART และ Audio	802.3
การจัดการในรูปแบบที่เคลื่อนที่ได้	Master	Mobil Station
ค่าใช้จ่ายวงจร	11 ดอลลาร์	46 ดอลลาร์

ตาราง 3.2 แสดงการเปรียบเทียบการทำงานระหว่าง Bluetooth กับ IEEE 802.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะพิเศษ	Bluetooth	802.11	802.11b	802.11a	HaperLan/1	HaperLan/2
Spectrum	2.4 GHz	2.4 GHz	2.4 GHz	5 GHz	5 GHz	5 GHz
อัตราการส่งข้อมูล	723 Kbps	1.2 Mbps	5 Mbps	32 Mbps	20 Mbps	32 Mbps
การเชื่อมต่อ	PTMP	PTP	PTP	PTP	PTMP	PTMP
การเลือกความถี่	FHSS	FHSS DSSS	DSSS	OFDM	OFDM	ช่องสัญญาณย่อยแบบ OFDM UMTS
การตรวจสอบสิทธิ์	มี	ไม่มี	ไม่มี	ไม่มี	Address	Adress
เหมาะกับเครือข่าย	เครือข่ายใดๆก็ได้	Ethernet	Ethernet	Ethernet	Ethernet ATM Firewire	Ethernet ATM Firewire UMTS
CQDDR	Option	ไม่มี	ไม่มี	ไม่มี	ไม่มี	
วิธีการเข้ารหัส	40 บิต	40 บิต	40 บิต	40 บิต	DES	DES 3DES
ข้อมูล	RC 4	RC 4	RC 4	RC 4		

ตาราง 3.3 แสดงความแตกต่างระหว่าง Bluetooth กับผลิตภัณฑ์อื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 การสถาปนาการเชื่อมต่อเครือข่าย Bluetooth

การจัดตั้ง Piconet

Piconet สามารถถูกจัดตั้งด้วยวิธีใดวิธีหนึ่งใน 4 วิธี ดังนี้

1. Page ที่ Master ใช้เพื่อการเชื่อมต่อกับ Slave
2. วิธีการทาง Page Scan (หน่วยทำงานที่ใช้เพื่อฟัง Access Code)
3. การสลับตำแหน่งไปมาระหว่าง Master กับ Slave
4. หน่วยทำงานที่แสดงให้เห็นว่า มี Slave ที่ไม่ได้พร้อมทำงานในขณะนั้น

เพื่อให้การจัดตั้งเป็นไปอย่างมีประสิทธิภาพ และราบลื่น จึงมีการใช้กระบวนการที่เรียกว่า การสอบถาม (Inquiry) และการร้องเรียกแบบ Paging (การทำ Paging เป็นการส่งเลขหมาย ID ของอุปกรณ์ไปยังอุปกรณ์ที่ต้องการจะติดต่อสื่อสารด้วย) การสอบถามเป็นการส่งข่าวสารออกมาของอุปกรณ์ Bluetooth มีจุดประสงค์เพื่อการค้นหาดูว่า ในรัศมีใกล้เคียงกับมันมีอุปกรณ์ Bluetooth ใดบ้างที่กำลังทำงานอยู่ในบริเวณนั้น หากบริเวณนั้น มีอุปกรณ์ Bluetooth ทำงานอยู่มันจะตรวจพบทันที เนื่องจากอุปกรณ์ที่อยู่ในบริเวณนั้น จะส่งข่าวสารตอบรับกลับมา จากนั้น จะมีการส่ง Paging ออกไป เพื่อแสดงตนระหว่างกันก่อนที่ทำการสื่อสารระหว่างกันต่อไป

ความหมายของ Piconet

Piconet เป็นเครือข่ายเล็กๆ ที่ประกอบไปด้วยกลุ่มอุปกรณ์สื่อสาร และคอมพิวเตอร์ที่สื่อสารกันภายใต้ Bluetooth และเป็นรูปแบบการเชื่อมต่อกันลักษณะ Ad-hoc โดย Piconet อุปกรณ์สามารถมีได้ไม่เกิน 8 หน่วยนี่เป็นเหตุผลที่ว่า เพราะเหตุใด Address ของอุปกรณ์ต่าง ๆ ภายใต้ Piconet สามารถมีขนาดไม่เกิน 3 บิตเท่านั้น

อุปกรณ์ต่างๆ ภายใต้ Bluetooth ทุกชิ้นจะต้องมีเลขหมายหรือรหัสที่ใช้แสดงความมีตัวตนของมัน อย่างไรก็ดี เมื่อมีการจัดตั้ง Piconet ขึ้น จะต้องมีอุปกรณ์ตัวหนึ่งทำหน้าที่เป็น Master ซึ่งทำหน้าที่ประสานจังหวะการทำงานกับอุปกรณ์ที่เป็น Slave อื่น ๆ

ระบบรักษาความปลอดภัยของ Bluetooth

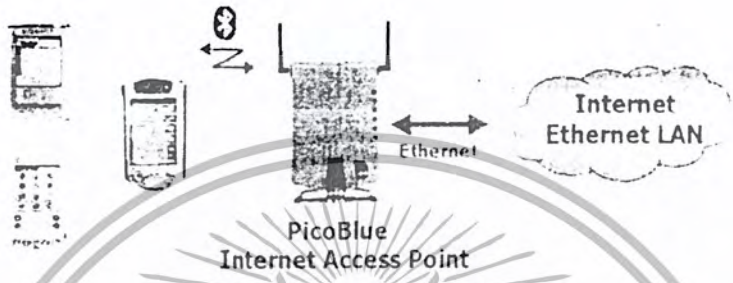
ระบบรักษาความปลอดภัย หมายถึง ขีดความสามารถของเครือข่ายจะต้องสามารถตอบสนองความต้องการ ดังต่อไปนี้

1. การรับส่งข้อมูลจะต้องราบลื่น ข้อมูลข่าวสารสามารถเดินทางไปถึงผู้รับได้อย่างปลอดภัย ไม่ตกหล่น แม้แต่นิดเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ข้อมูลข่าวสารที่เดินทางไปมาหาสู่ระหว่างกัน จะต้องไม่สามารถถูกดักฟังโดยบุคคลที่สาม ที่ไม่เกี่ยวข้อง

ความต้องการทั้งสองประการ ล้วนได้รับการตอบรับ โดยระบบของ Bluetooth



รูปที่ 3.17 แสดงลักษณะการเชื่อมต่อเครือข่าย Piconet

การสื่อสารข้อมูลที่ปลอดภัยกว่า

Bluetooth มีความเสถียรในการสื่อสารที่นับว่าอยู่ในเกณฑ์ดีมาก อีกทั้งมีระบบการเข้ารหัสข่าวสารที่น่าเชื่อถือ รวมทั้งมีกระบวนการพิสูจน์สิทธิ์ ที่จะช่วยเสริมความมั่นใจให้กับผู้ใช้งานมาก ยิ่งไปกว่านั้น Bluetooth ใช้เทคนิคที่เรียกว่า Frequency Hopping ที่มีการก้าวกระโดดไปมาระหว่างช่องสัญญาณที่รวดเร็ว และเร็วพอที่จะหลบสัญญาณรบกวนของอุปกรณ์คลื่นวิทยุ ที่อยู่ในบริเวณใกล้เคียง อัตราความเร็วของการกระโดดของ Frequency Hopping มีค่าเท่ากับ 1600 Hops ต่อวินาที ซึ่งเรียกว่า ระบบไร้สายอื่น ๆ ไม่เพียงเท่านี้ Bluetooth ยังมีระบบที่เรียกว่า **"Automatic Output Adaptation"** เพื่อให้มีความสามารถลดขนาดรัศมีทำการได้ตามที่ต้องการได้ จึงทำให้ Bluetooth เป็นระบบที่มีความยากยิ่งต่อการแอบดักฟัง

การตรวจสอบคุณภาพของข้อมูลข่าวสารบน Bluetooth

การตรวจสอบความสมบูรณ์ของข้อมูลข่าวสารเป็นเรื่องที่สำคัญมาก เนื่องจากเราไม่ต้องการให้ข้อมูลข่าวสารสามารถถูกดักฟังจากผู้ไม่เกี่ยวข้อง ดังนั้น ระบบ Bluetooth จึงต้องมีองค์ประกอบหลายประการเพื่อให้บรรลุจุดประสงค์ ดังกล่าว ประกอบด้วยส่วนประกอบดังนี้

- ระบบสร้างตัวเลขแบบสุ่ม (Random Number Generation) มีไว้เพื่อสร้างตัวเลขสำหรับการเข้ารหัสที่ไม่สามารถคาดเดาได้
- การเข้ารหัสข้อมูลข่าวสาร (Encryption)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การบริหารจัดการกับกุญแจที่ใช้เข้ารหัส
- การพิสูจน์สิทธิ์ (Authentication)

แบบแผนการตรวจสอบและแก้ไขข้อผิดพลาดของ Bluetooth

การใช้งาน Bluetooth ในบริเวณที่มีการกวนกันของสัญญาณเป็นเรื่องที่หลีกเลี่ยงไม่ได้ ดังนั้น เพื่อให้ข้อมูลข่าวสารมีความปลอดภัยไม่ผิดพลาด จึงจำเป็นต้องมีวิธีการตรวจสอบและแก้ไขข้อผิดพลาดของข่าวสาร วิธีการตรวจสอบหาความผิดพลาดของข้อมูลข่าวสาร ระบบ Bluetooth มีการใช้การคำนวณเพื่อหาค่าผลรวมหลายๆ รูปแบบ และเมื่อใดที่ตรวจพบความผิดพลาด ก็จะใช้แบบแผนการแก้ไขข้อผิดพลาดอย่างใดอย่างหนึ่งใน 3 วิธีการ ดังนี้

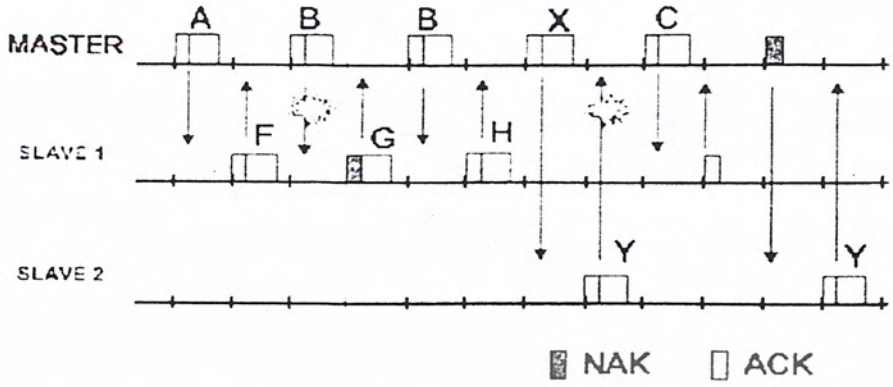
1. 1/3 Rate FEC (Forward Error Correction)
2. 2/3 Rate FEC
3. ARQ Unnumbered (Automatic Repeat For Request)

จุดประสงค์ของการใช้ FEC กับข้อมูลข่าวสารที่บรรทุกมากับ Packet ก็เพื่อที่จะลดจำนวนครั้งของการส่งข้อมูลซ้ำ เพราะการส่งข้อมูลซ้ำเกิดขึ้นบ่อย ๆ ไม่เป็นผลดีต่อเครือข่าย อีกทั้งเสี่ยงต่อความปลอดภัย แต่อย่างใดก็ตาม ในพื้นที่ ๆ ไม่มีสัญญาณรบกวน และปราศจากเหตุที่ทำให้เกิดข้อผิดพลาดของข้อมูลให้เลิกใช้ FEC เนื่องจากเป็นการเพิ่ม Overhead ให้กับการรับส่งข้อมูล ส่งผลให้ประสิทธิภาพการทำงานของเครือข่าย ถดถอยลงได้

การใช้แบบแผน ARQ

ARQ มาจากคำว่า **"Automatic Repeat For Request"** เป็นระบบควบคุมความผิดพลาดของการรับส่งข้อมูลอย่างง่ายๆ เมื่อใดที่ผู้รับได้ตรวจพบความผิดพลาดในการส่งข้อมูล ผู้รับจะส่งสัญญาณกลับมาให้เพื่อขอให้ส่งข้อมูลกันใหม่ ส่วนทางด้านของผู้ส่ง จะส่งข้อมูลซ้ำจนกระทั่งไม่มีข้อมูลผิดพลาด ซึ่งโดยทั่วไปผู้ส่งมักจะส่งข้อมูลซ้ำในส่วนที่ผิดพลาดเท่านั้น ถ้าจำนวนครั้ง ในการพยายามส่งข้อมูลไปให้ผู้รับมีจำนวนครั้งที่มากกว่าค่าสูงสุดที่ได้กำหนดตั้งไว้ โปรแกรมจะหยุดการติดต่อในทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.18 แสดงลักษณะการใช้ ARQ เพื่อควบคุมความผิดพลาดของข่าวสาร

จากรูปที่ 3.18 เป็นการแสดงเหตุการณ์ที่ต้องใช้ ARQ 2 เหตุการณ์กล่าวคือ เมื่อใดข่าวสารในรูปแบบของบล็อกของข้อมูลมีความเสียหายเกิดขึ้น ที่ผู้รับได้ตรวจพบ ดังนั้นในคราวต่อไปที่ผู้รับมีโอกาส สื่อสารกับผู้ส่ง ผู้รับจะส่งสัญญาณที่เรียกว่า "NAK" (Negative Acknowledge) ออกมา ซึ่งการกระทำนี้ จะทำให้ผู้ส่งทำการส่งข้อมูลซ้ำ เฉพาะในส่วนที่มีปัญหาออกมาให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

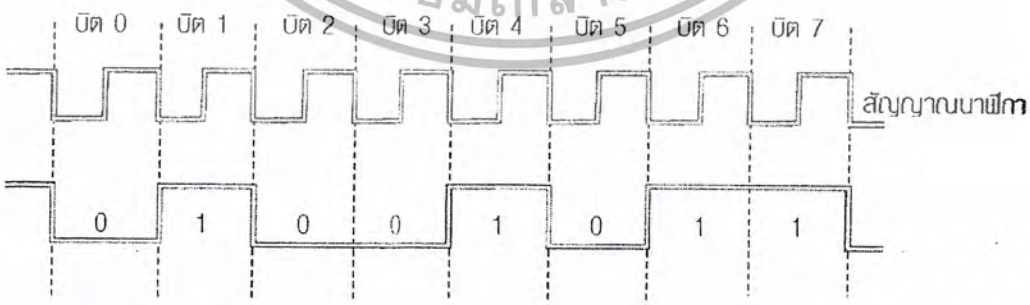
บทที่ 4
การสื่อสารข้อมูลแบบอนุกรม

การติดต่อข้อมูลผ่านทางพอร์ตอนุกรมจะช้ากว่าพอร์ตขนาน เนื่องจากพอร์ตขนานนั้นส่งข้อมูลที่ละหลายบิตและความเร็วในการสื่อสารก็เร็วกว่า ในขณะที่พอร์ตอนุกรมส่งข้อมูลที่ละ 1 บิต แต่การสื่อสารข้อมูลกลับส่งข้อมูลได้ไกลกว่าพอร์ตขนานและจำนวนสายที่ใช้ก็น้อยกว่า การสื่อสารข้อมูลของพอร์ตอนุกรมมีรูปแบบการส่งข้อมูลอยู่ด้วยกัน 3 แบบ คือ

1. Simplex สามารถส่งข้อมูลได้อย่างเดียว เป็นการสื่อสารแบบทางเดียว
2. Half - Duplex สามารถส่งข้อมูลไปยังปลายทางได้และยังรับข้อมูลจากปลายทางได้ แต่ไม่สามารถรับและส่งข้อมูลในเวลาเดียวกันได้
3. Full - Duplex สามารถรับและส่งข้อมูลในเวลาเดียวกันได้

4.1 รูปแบบของการสื่อสารข้อมูลผ่านพอร์ตอนุกรม

1. การสื่อสารแบบซิงโครนัส (Synchronous) สำหรับการสื่อสารข้อมูลแบบซิงโครนัสนี้จะใช้สัญญาณนาฬิกาในการรับส่งสัญญาณ เช่น สายคีย์บอร์ดคอมพิวเตอร์โดยจะมีสายสัญญาณเส้นหนึ่งเป็นสายสัญญาณนาฬิกา ส่วนสายอีกหนึ่งเป็นสายของข้อมูล (และจะมีสายกราวน์ด้วย) แต่การสื่อสารแบบซิงโครนัสเหมาะสำหรับการสื่อสารในระยะใกล้ข้อมูลที่จะส่งมีไม่มากนักเพราะถ้าระยะไกลมากสัญญาณนาฬิกาจะมีปัญหา และต้องใช้สายหลายเส้นทำให้สิ้นเปลืองสายมากซึ่งสามารถแสดงดังรูปที่ 4.1



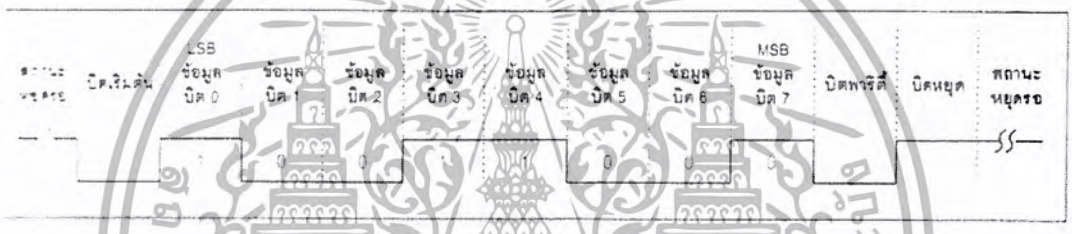
รูปที่ 4.1 ลักษณะสัญญาณของการสื่อสารแบบอะซิงโครนัส

2. การสื่อสารแบบอะซิงโครนัส (Asynchronous) สำหรับการสื่อสารแบบอะซิงโครนัสก็คือ การรับและส่งข้อมูลไปในสาย โดยไม่จำเป็นต้องมีสัญญาณนาฬิการ่วมด้วยเหมือนการส่งแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซิงโครนัส แต่มีการกำหนดสัญญาณนาฬิกาทั้งภาครับและภาคส่งให้มีค่าเท่ากันซึ่งจะเรียกสัญญาณที่ใช้ในการกำหนดค่าให้ภาครับ และภาคส่งนี้ว่า **"อัตราการถ่ายทอดข้อมูล"** หรือ **"บอดเรต"** (Baud rate) มีหน่วยเป็นบิตต่อวินาที (bit per second : bps) รูปแบบของข้อมูลที่ใช้ในการรับส่งแบบอะซิงโครนัสประกอบด้วย 4 ส่วนด้วยกัน คือ

1. บิตเริ่มต้น (start bit) จะมีขนาด 1 บิต
2. บิตข้อมูลแบบอนุกรมจะมีขนาด 5, 6, 7 หรือ 8 บิต
3. บิตตรวจสอบพาริตี้ (Parity Bit) จะมีขนาด 1 บิตหรือไม่มี
4. บิตปิดท้าย (Stop bit) 1, 1.5 หรือ 2 บิต



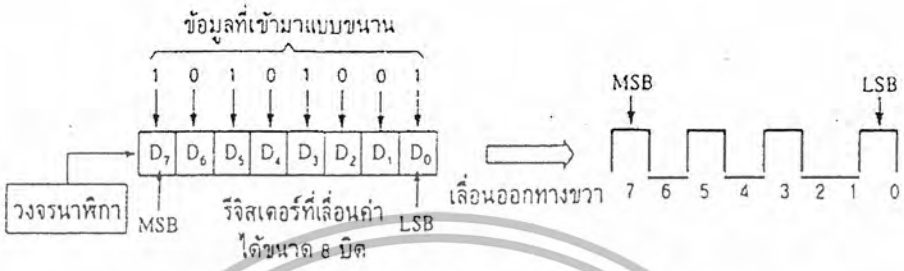
รูปที่ 4.2 แสดงรูปแบบของข้อมูลอนุกรมแบบอะซิงโครนัส

เมื่อไม่มีข้อมูลส่งมาที่ขาของ DATA จะมีสถานะลอจิกเป็น "1" ซึ่งจะเรียกสถานะนี้ว่า **"สถานะหยุดรอ"** (waiting stage) การเริ่มต้นส่งข้อมูลจะเริ่มจากให้ขา DATA มีลอจิก "0" ด้วยช่วงระยะเวลา 1 บิต เรียกบิตนี้ว่า **"บิตเริ่มต้น"** จากนั้นบิตข้อมูลจะถูกส่งออกไป โดยเริ่มจากบิตที่มีนัยสำคัญต่ำสุด (LSB) ก่อน ซึ่งข้อมูลภายในไบต์ที่ส่งอาจจะมีจำนวนบิต 5, 6, 7 หรือ 8 บิตก็ได้ จากนั้นก็จะตามด้วย **"บิตพาริตี้"** ซึ่งจะใช้ในการที่จะตรวจสอบความผิดพลาดในการส่งข้อมูล บิตสุดท้ายที่ส่ง คือ บิตปิดท้ายซึ่งขา DATA จะมีลอจิกเป็น "1" อีกครั้ง โดยจะใช้ระยะเวลาอย่างน้อย 1 บิต, 1.5 บิต หรือ 2 บิต เพื่อที่จะแสดงว่าสิ้นสุดการส่งข้อมูลแล้ว

อุปกรณ์พิเศษที่ว่านั้นได้รับการออกแบบมานั้นใช้สำหรับการรับและส่งข้อมูลแบบอะซิงโครนัส เรียกว่า **"Universal Asynchronous Receiver / Transmitter หรือ UART"** โดยที่ UART ทำหน้าที่แปลงข้อมูลแบบขนานจากคอมพิวเตอร์ให้อยู่ในรูปแบบอนุกรมของอะซิงโครนัสแล้วส่งออกไป จากนั้นก็จะแปลงสัญญาณอนุกรมแบบอะซิงโครนัสที่ป้อนเข้ามายัง UART ให้เป็นแบบขนานก่อนที่จะส่งเข้าสู่คอมพิวเตอร์ นอกจาก UART จะส่งเข้าข้อมูลไปยังคอมพิวเตอร์แล้วยังทำหน้าที่แจ้งข้อมูลอื่นๆ ให้กับคอมพิวเตอร์ทราบอีกด้วย เช่น อัตราความเร็วในการรับส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูล (บอดเรต) รูปแบบการส่งข้อมูล ความผิดระหว่างการถ่ายทอดข้อมูล (ผิดพลาดจาก พาริตี้ เฟรมข้อมูล โอเวอร์รัน) ซึ่งจะสามารถแสดงให้เห็นจากรูปที่ 4.3



รูปที่ 4.3 แสดงการเปลี่ยนข้อมูลแบบขนานให้เป็นแบบข้อมูลแบบอนุกรม

ภายใน UART จะมีส่วนของวงจรสร้างบอดเรตที่สามารถโปรมแกรมได้ (programmable buadrate generators) โดยการกำหนดค่าตัวหารให้กับสัญญาณนาฬิกาของ UART โดยที่ตัวหารนี้จะมีขนาดเท่ากับ 16 บิต ดังนั้นจึงสามารถกำหนดตัวหารให้อยู่ในช่วง 1 - 65,535 UART ซึ่งสามารถทำให้ส่งข้อมูลทั้งแบบฮาล์ฟดูเพล็กซ์ (half duplex) และฟลูดูเพล็กซ์ (full duplex)

ชนิดของ UART ในเครื่องคอมพิวเตอร์ทั่วไป จะมี UART ที่ใช้งานกันอยู่ 2 เบอร์ คือ 8250 ซึ่งเป็น UART มาตราฐานที่ใช้กันมายาวนาน ใน UART เบอร์นี้จะมีบัฟเฟอร์สำหรับรับและส่งข้อมูลตำแหน่งเดียวกัน ทำให้การรับและส่งข้อมูลถูกจำกัดความเร็วอยู่ที่ 57.6 กิโลบิตต่อวินาทีเท่านั้น แต่ UART เบอร์นี้ถือเป็นต้นแบบของ UART ที่ใช้ในคอมพิวเตอร์ โดยคอมพิวเตอร์ทุกๆ รุ่นนั้นจะต้องสนับสนุนการทำงานตามรูปแบบของ UART เบอร์นี้ ส่วนอีกเบอร์ 16450 มีความสามารถในการรับส่งข้อมูลได้เร็ว 115,200 บิตต่อวินาที และเพิ่มรีจิสเตอร์สำหรับพักข้อมูล นอกจากนี้ยังมียังเพิ่มส่วนของชิฟต์จิสเตอร์แบบ FIFO (First In First Out) ขนาด 16 ไบต์ เข้าไปเป็นผลทำให้สามารถสนับสนุนความเร็วในการรับส่งข้อมูลที่ 256 กิโลบิตต่อวินาทีได้ โดยคอมพิวเตอร์ในปัจจุบันใช้ UART เบอร์นี้หรือใหม่กว่านี้ เช่น เบอร์ TL 16C750 ซึ่งมีรีจิสเตอร์แบบFIFO ขนาด 64 ไบต์ ทำงานได้ที่ระดับ +5v และ +3v มีหมวดประหยัดพลังงานสามารถรับส่งข้อมูลได้ที่ความเร็ว 1 เมกะบิตต่อวินาที เมื่อใช้สัญญาณนาฬิกาที่ 16 MHz อย่างไรก็ตามความเร็วในการส่งข้อมูลที่มากมายของ UART เบอร์ใหม่ๆ ไม่ได้ช่วยให้การรับส่งข้อมูลของคอมพิวเตอร์เร็วขึ้น เนื่องจากว่าคอมพิวเตอร์ยังใช้ที่ความถี่ของสัญญาณนาฬิกาในการแปลงข้อมูลเพียง 1.8432 MHz เท่านั้น อัตราความเร็วในการรับและส่งข้อมูลของการรับส่งข้อมูลแบบอะซิงโครนัส คือ **"ค่าบอดเรต"** ซึ่งก็ คือ จำนวนบิตต่อวินาทีที่ใช้ในการรับและส่งข้อมูล มาตราฐานบอดเรตที่ใช้สำหรับในพอร์ตอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RS-232 ได้แก่ 110, 150, 300, 600, 1200, 2400, 4800, 19200 บิตต่อวินาที และมีการเพิ่มค่ามากขึ้นตามเทคโนโลยีคอมพิวเตอร์ซึ่งการรับและส่งข้อมูล โดยจะไม่ผ่านโมเด็มสามารถกำหนดค่าบอดเรตได้สูงถึง 115200 บิตต่อวินาที เนื่องจากบอดเรต คือ จำนวนบิตของข้อมูลที่สามารถถ่ายทอดข้อมูลภายใน 1 วินาที

ตัวอย่าง ข้อมูลอนุกรมถูกส่งในลักษณะ 8 บิต ไม่มีการตรวจสอบพาริตี้ มีบิตเริ่มต้น 1 บิต และบิตปิดท้าย 1 บิต ความยาวของข้อมูลที่ส่ง 10 บิต เมื่อใช้บอดเรตในการส่งข้อมูลให้มีค่าเท่ากับ 9600 บิตต่อวินาที ก็จะสามารถรับส่งข้อมูลได้ด้วยความเร็ว 960 ไบต์ต่อวินาที ถ้ามีการใช้ค่าของพาริตี้อีก 1 บิต แล้วความเร็วในการรับส่งข้อมูลจะเหลือเป็น 872 ไบต์ต่อวินาที เนื่องจากข้อมูลจะเพิ่มจาก 10 บิต เป็น 11 บิต ซึ่งสามารถแสดงได้ดังตาราง 4.1

อัตราการถ่ายทอดข้อมูล	ไบต์ต่อวินาที	ช่วงเวลาของแต่ละบิต
110	10	9.91 ms
150	15	6.67 ms
300	30	3.33 ms
600	60	1.67 ms
1200	120	0.833 ms
2400	240	0.417 ms
4800	480	0.208 ms
9600	960	0.104 ms
19200	1920	0.052 ms
38400	3840	0.026 ms

ตารางที่ 4.1 อัตราการถ่ายทอดข้อมูล, ช่วงเวลาของแต่ละบิต และอัตราการถ่ายทอดข้อมูลที่ใช้ในการย้ายข้อมูลอนุกรม

แต่ในการส่งข้อมูลด้วยอัตรา 110 จะมีลักษณะที่แต่ต่างจากอัตราบอดอื่นๆ ซึ่งจะใช้บิตเริ่มต้น 1 บิต ข้อมูล 8 บิต และบิตสุดท้าย 2 บิต ซึ่งรวมแล้วจะเท่ากับ 11 บิต ส่วนอัตราบอดอื่นๆ จะใช้บิตข้อมูลเริ่มต้น 1 บิต, บิตข้อมูล 8 บิต และบิตสุดท้าย 1 บิต รวมแล้วจะเท่ากับ 10 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การตรวจสอบค่าพาริตี้สามารถกำหนดให้เป็นเลขคี่ (odd), เลขคู่ (even) หรือจะไม่มีการตรวจสอบพาริตี้ก็ได้ การตรวจสอบพาริตี้เป็นการตรวจจำนวนรวมของบิตที่เป็นลอจิก **"1"** ภายในข้อมูลที่ส่งไป 1 ไบต์ว่ามีจำนวนเป็นเลขคี่ หรือเลขคู่ด้วยที่ต้องรวมบิตพาริตี้เข้าไปด้วย

ตัวอย่าง ข้อมูลที่ใช้ส่งมีขนาด 8 บิต และมีค่าเท่ากับ 99 เลขฐานสิบหก หรือ 10011001 เลขฐานสองจะเห็นว่าข้อมูลในไบต์มีลอจิกเป็น **"1"** จำนวน 4 ตัวซึ่งเป็นเลขคู่ ดังนั้นถ้ากำหนดพาริตี้บิตให้เป็นเลขคู่ ค่าในบิตพาริตี้ จะต้องมีสถานะลอจิกเป็น **"0"** แต่ถ้าพาราริตี้บิตเป็นเลขคี่ ค่าที่บิตพาริตี้จะมีสถานะเป็น **"1"** เพื่อให้ข้อมูลใน 1 ไบต์ที่รวมบิตของพาริตี้รวมกันแล้วเป็นเลขคี่ซึ่งสามารถแสดงรูปแบบ ตามตาราง 4.2 เป็นการแสดงบิตพาริตี้ในการรับส่งข้อมูลของพอร์ตอนุกรม

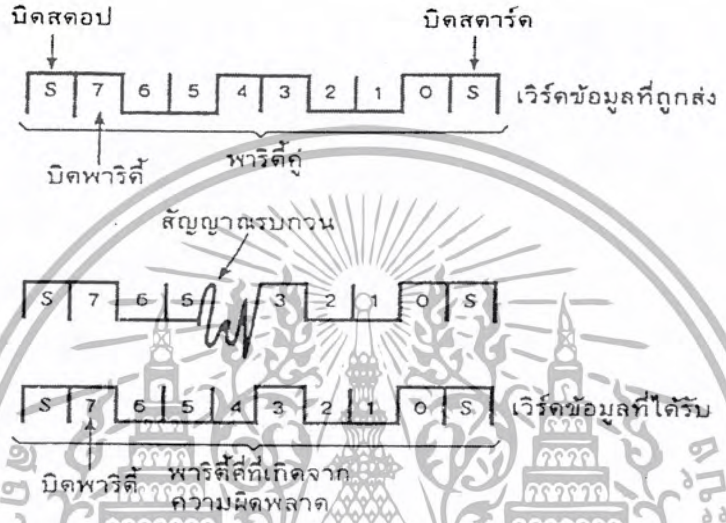
ข้อมูล	บิตพาริตี้คู่	บิตพาริตี้คี่
00000000	0	1
00000001	1	0
00000010	1	0
00000011	0	1
00000100	1	0
11111110	1	0
11111111	0	1

ตารางที่ 4.2 แสดงบิตพาริตี้ของข้อมูลพอร์ตอนุกรม

บิตพาริตี้ถูกสร้างขึ้นจากภาคส่งข้อมูลของ UART โดยภาครับจะต้องกำหนดคุณสมบัติการตรวจสอบพาริตี้ให้ตรงกันว่า เป็นคู่ หรือคี่ จากนั้นภาครับของรับ UART จะตรวจสอบค่าพาริตี้ที่เกิดขึ้นว่าเป็นคี่หรือคู่ โดยการนับจำนวนลอจิก **"1"** ทั้งหมดรวมบิตพาริตี้ด้วย ถ้ามีการกำหนดค่าพาริตี้ไว้เป็นเลขคู่ แต่อ่านค่าตัวเลขในการนับออกมาได้ตัวเลขเป็นคี่ ทางภาครับจะแจ้งข้อมูลให้ผู้ใช้ทราบนับเป็นการตรวจสอบความผิดพลาดของการถ่ายทอดข้อมูลที่ง่ายที่สุด แต่จะเชื่อถือได้เมื่อมีบิตข้อมูลที่ทำการส่งผิดพลาดเพียงบิตเดียวเท่านั้น ถ้าข้อมูลที่ส่งมีที่ผิดพลาดมากกว่า 1 บิต การตรวจสอบโดยวิธีนี้จะไม่ได้ผล สำหรับการตั้งพาริตี้เป็น NONE ทั้งภาครับและภาคส่งจะไม่มีการตรวจสอบพาริตี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.4 แสดงการส่งข้อมูลอนุกรมที่มีพาริตี้ที่เปลี่ยนไป เนื่องจากสัญญาณรบกวน (Noise) โดยบิตที่ 4 ในรูปจะถูกสัญญาณรบกวนทำให้มีการเปลี่ยนแปลงค่าจาก 1 ไป 0 ซึ่งข้อมูลที่ได้รับนั้นไม่ถูกต้อง



รูปที่ 4.4 แสดงการใช้บิตพาริตี้ตรวจสอบความผิดพลาดในการโอนย้ายข้อมูลแบบอนุกรม

4.2 มาตราฐานพอร์ตอนุกรมแบบ RS-232

มาตราฐานการเชื่อมต่อแบบอนุกรม RS-232 เป็นมาตราฐานทางอุตสาหกรรมที่ได้ออกแบบมาเพื่อใช้ในการส่งข้อมูลอนุกรมแบบอะซิงโครนัส 2 ทิศทาง โดยมาตราฐาน RS-232 ในอดีตนั้นถูกออกแบบมาเพื่อส่งข้อมูลจากคอมพิวเตอร์ไปยังโมเด็มเพียงอย่างเดียว เพื่อจะนำข้อมูลจากโมเด็มสื่อสารผ่านสายโทรศัพท์ไปยังคอมพิวเตอร์อีกชุดซึ่งอยู่ห่างไกลกัน โดยคณะกรรมการที่เรียกว่า **" สมาคมอุตสาหกรรมอิเล็กทรอนิกส์ "** (Electronic Industrial Association : EIA) ได้วางมาตราฐานที่มีชื่อเรียกกันว่า "**EIA RS-232**" ขึ้นมา ในปี ค.ศ 1969 มาตราฐานนี้ในช่วงแรกจะใช้คอนเน็กเตอร์แบบ DB-25 โดยกำหนดความยาวสูงสุดของสายสัญญาณไว้ที่ 50 ฟุต มีระดับสัญญาณตั้งแต่ -3 ถึง -15 volt แสดงว่ามีข้อมูล (Mark) และ +3 ถึง +15 volt แสดงว่ามีช่องว่าง (Space)

ในยุคแรกๆ มาตราฐานของ RS-232 ได้กำหนดรูปแบบของอุปกรณ์ที่ใช้ในการเชื่อมต่อข้อมูล(Data Terminal Equipment : DTE) กับวงจรข้อมูลปลายทาง(Data Circuit Equipment : DCE) ไว้ว่าอุปกรณ์ DTE จะต้องเป็นอุปกรณ์ที่สามารถประมวลผลในตัวเอง เช่น ไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรือไมโครคอมพิวเตอร์ซึ่งมีความสามารถในการสร้างข้อมูลอนุกรมได้ ส่วนอุปกรณ์ DCE ทำหน้าที่เป็นเพียงตัวรับข้อมูลที่ส่งมาจาก DTE เท่านั้น ทั้งนี้ก็เพื่อป้องกันไม่ให้เกิดการส่งข้อมูลบนสายเดียวโดยการรับส่งรับข้อมูลของอุปกรณ์ทั้งสองจะกระทำผ่านมาตราฐาน RS-232 ซึ่งแบ่งออกเป็น 2 ชนิดประเภท ซึ่งประกอบด้วย

1. อุปกรณ์ DTE (Data Terminal Equipment) เป็นอุปกรณ์สำหรับส่งข้อมูล (เอาท์พุต)
2. อุปกรณ์ DCE (Data Circuit Equipment) เป็นอุปกรณ์สำหรับรับข้อมูล (อินพุต)

ข้อแตกต่างของอุปกรณ์ DTE และDCE อย่างหนึ่งที่เห็นได้อย่างชัดเจน คือ คอนเน็กเตอร์DTE จะเป็นตัวผู้ ส่วนคอนเน็กเตอร์ของ DCE จะเป็นตัวเมีย ซึ่งพอร์ตอนุกรมของคอมพิวเตอร์ที่ใช้กันอยู่อย่างทั่วไปเป็นแบบ DTE ส่วนคอนเน็กเตอร์ของโมเด็มจะเป็นแบบ DCE สำหรับการใช้งานเครื่องคอมพิวเตอร์พอร์ต RS-232 มักถูกใช้เชื่อมต่อกับโมเด็ม คีย์บอร์ด หรือ เมาส์ โดยสามารถรับส่งข้อมูลได้ที่ความยาวของสายสัญญาณสูงสุดถึง 20 เมตร

โดยที่คุณลักษณะ RS-232 มีลักษณะดังนี้

อัตรารับส่งข้อมูล	:	0 – 20000 บิต/วินาที
ระดับแรงดันเอาท์พุตสูงสุดในสภาวะไม่มีโหลด	:	-25 โวลท์ (ลอจิก)
	:	+25 โวลท์ (ลอจิก)
ระดับแรงดันเอาท์พุตสำหรับโหลด 3-7 กิโลโอห์ม		
	: ลอจิก “1”	- 15 โวลท์ (7 กิโลโอห์ม)
		- 5 โวลท์ (3 กิโลโอห์ม)
	: ลอจิก “0”	+ 15 โวลท์ (7 กิโลโอห์ม)
		+ 5 โวลท์ (3 กิโลโอห์ม)
กระแสเอาท์พุตเมื่อลัดวงจร	:	สูงสุด 500 มิลลิแอมป์
เอาท์พุตของอิมพีแดนซ์เมื่อไม่จ่ายไฟเลี้ยง	:	ต่ำสุด300 โอห์ม
สลูว์เรตทางเอาท์พุตสูงสุด	:	30 โวลท์/ไมโครวินาที
ความต้านทานอินพุตของภาครับ	:	สูงสุด 7 กิโลโอห์ม
	:	ต่ำสุด 3 กิโลโอห์ม
ค่าความจุอินพุตของภาครับ	:	สูงสุด 2500 กิโลฟารัด
ย่านแรงดันอินพุตของภาครับ	:	-25v ถึง +25v

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ลักษณะการเชื่อมต่อคอนเน็กเตอร์ของพอร์ต RS-232

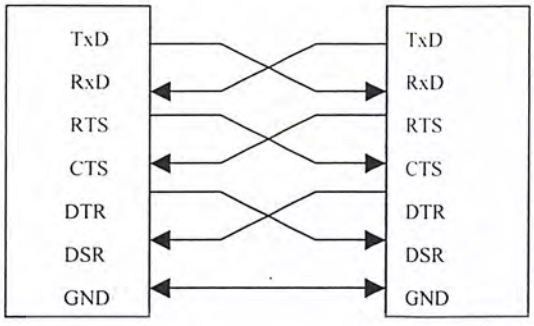
มาตรฐานการเชื่อมต่อแบบ RS-232 จะใช้คอนเน็กเตอร์แบบ DB-25 ตัวผู้ หรือ DB-9 ตัวผู้ซึ่งคอนเน็กเตอร์แบบ DB-25 จะมีขาต่อใช้งานเพียง 9 เส้นเช่นเดียวกันแบบ DB-9 เนื่องจากขาอื่นๆ ที่เคยใช้งานอดีต ปัจจุบันมีการใช้งานไม่มากนักจึงได้ถูกยกเลิกไป โดยแสดงได้ดังรูปที่ 4.5 แสดงรูปร่างและตำแหน่งขาของอุปกรณ์

คอนเน็กเตอร์ DB-9	คอนเน็กเตอร์ DB-25	ชื่อของสายสัญญาณ	สัญลักษณ์	ชนิดของสายสัญญาณ
1	8	Data Carrier Detecter	DCD	อินพุท
2	3	Received Data	RxD	อินพุท
3	2	Transmitted Data	TxD	เอาท์พุต
4	20	Data Teminal Ready	DTR	เอาท์พุต
5	7	Signal Ground	GND	-
6	6	Data Set Ready	DSR	อินพุต
7	4	Request To Send	RTS	เอาท์พุต
8	5	Clear To Send	CTS	อินพุต
9	22	Ring Indicator	RI	อินพุต

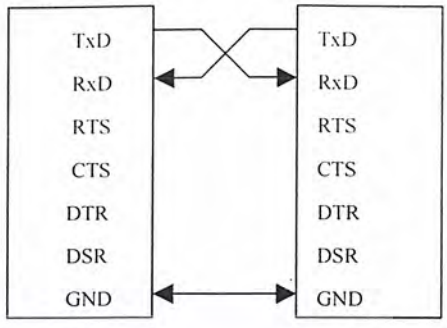
รูปที่ 4.5 รูปแบบการเชื่อมต่อขาพอร์ตอนุกรมมาตราฐาน RS-232 ของ DB-9 และ DB-25

สำหรับการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอก แสดงได้ดังรูปที่ 4.6 ลูกศรในรูปแสดงถึงทิศทางของข้อมูล ในรูปที่ 4.6 (ก) เป็นการเชื่อมต่อแบบ Null Modem หรือเป็นการเชื่อมต่อโดยไม่ผ่านโมเด็มโดยมีการตรวจสอบแฮนด์เช็กเต็มรูปแบบ ส่วนในรูปที่ 4.6 (ข) เป็นการเชื่อมต่อแบบ Null Modem ในลักษณะที่ใช้สายสัญญาณเพียง 3 เส้น โดยเส้นหนึ่งเป็นเส้นสำหรับส่งข้อมูล ส่วนอีกเส้นสำหรับรับข้อมูล และเส้นสุดท้ายเป็นกราวน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 (ก) แบบ Null Modem



รูปที่ 4.6 (ข) แบบอาศัยสัญญาณ 3 เส้น

รูปที่ 4.6 การต่ออุปกรณ์ภายนอกกับพอร์ตอนุกรม RS-232 ของคอมพิวเตอร์ในลักษณะต่างๆ

สำหรับรายละเอียดหน้าที่การทำงานในแต่ละขาของพอร์ตอนุกรม RS-232 มีดังนี้

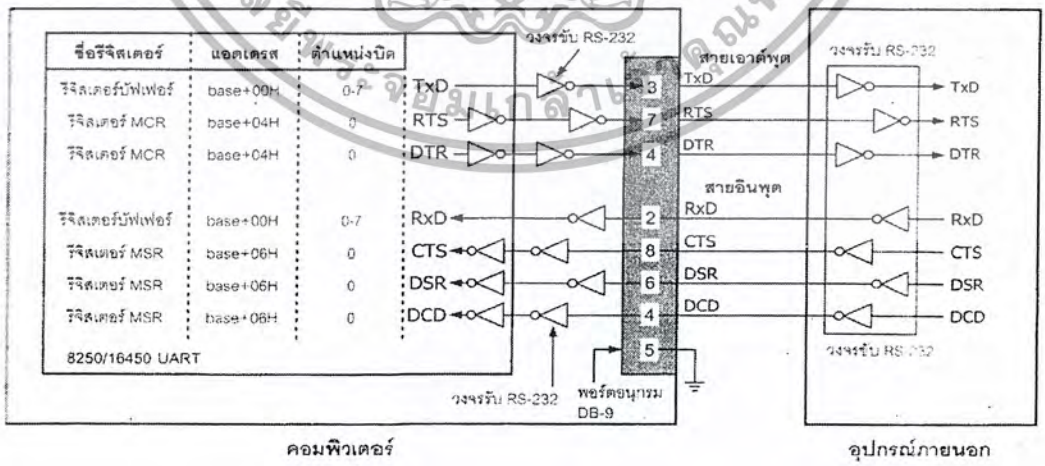
- Data Carrier Detect : DCD หรือเรียกว่าCarrier Detect : CD ขานี้จะแอกตีฟเมื่อมีการส่งสัญญาณพาห์จากอุปกรณ์สื่อสารข้อมูล เช่น โมเด็ม สำหรับการใช้งานปกติขานี้ไม่ได้ถูกใช้งานมากนัก
- Receive Data : RD หรือ RXD ขานี้ใช้เพื่อรับสัญญาณอนุกรมเข้ามายังคอมพิวเตอร์ โดยนำข้อมูลที่อ่านได้ไปเก็บไว้ในรีจิสเตอร์บัฟเฟอร์
- Transmitted Data : TD หรือTXD ใช้ส่งข้อมูลจากคอมพิวเตอร์เพื่อให้อุปกรณ์ปลายทางรับรู้ว่าต้องการติดต่อด้วยขา DTR นี้ จะต้องเชื่อมต่อกับ DSR ของอุปกรณ์ปลายทาง และขา DTR ของอุปกรณ์ปลายทางต้องเชื่อมต่อกับขา DSR ของคอมพิวเตอร์ ถ้าใช้ในการเชื่อมต่อแบบ Null Modem ซึ่งใช้สายในการเชื่อมต่อเพียง 3 เส้นจะต้องต่อขา DSR กับ DTR ของตัวมันเองเข้าด้วยกัน และจะต้องต่อขา DCD นี้เมื่อโปรแกรมที่ใช้ในการสื่อสารมีการตรวจจับสัญญาณพาห์
- Signal Ground : GND กราวด์ของระบบ
- Data Set Ready : DSR ขานี้จะใช้กับขา DTR เพื่อใช้ตรวจสอบการเชื่อมต่อระหว่างคอมพิวเตอร์กับอุปกรณ์ปลายทาง ซึ่งขานี้จะเป็นขาที่สำหรับรับข้อมูลจากภายนอกซึ่งถูกส่งมาจากขา DTR
- Request To Send : RTS เป็นขาสำหรับส่งสัญญาณข้อมูลให้ทางอุปกรณ์ปลายทางส่งข้อมูลกลับมายังคอมพิวเตอร์ โดยขาที่รับสัญญาณ RTS ก็คือขา CTS ในกรณีที่ใช้การเชื่อมต่อแบบ Null Modem 3 สาย จะต้องเชื่อมต่อขา RTS และ CTS ของตัวมันเองเข้าด้วยกัน เพื่อให้การรับและส่งข้อมูลสามารถเกิดขึ้นได้ตลอดเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Clear To Send : CTS จะคอยรับสัญญาณจากขา RTS เมื่อรับสัญญาณได้ ข้อมูลที่ขา TxD จะถูกส่งออกไป ดังนั้นขานี้จึงถูกใช้เพื่อตรวจสอบอุปกรณ์ต่อพ่วงว่าพร้อมที่จะรับข้อมูลหรือไม่
- Ring Indicator : RI ใช้แสดงสัญญาณเรียกสายโทรศัพท์ปกติในการสื่อสาร โดยทั่วไปสายนี้จะไม่ถูกใช้งานจะใช้งานก็ต่อเมื่อมีการเชื่อมต่อกับโมเด็ม และโปรแกรมมีการตรวจสอบสัญญาณนี้เท่านั้น

4.4 ลักษณะสัญญาณอินพุทและเอาต์พุทของพอร์ต RS-232

สัญญาณเอาต์พุทที่ใช้ในการควบคุม (RTS และ DTR) และสัญญาณแสดงสถานะอินพุท (CTS, DSR และDCD) ของพอร์ต RS-232 จะถูกกลับสถานะภายในตัวของ UART ส่วนสัญญาณข้อมูลทั้งภาคส่งและรับจะไม่ถูกกลับสถานะ UART จะให้ระดับของสัญญาณเอาต์พุทออกมาเป็นแบบทีทีแอลเท่านั้น ดังนั้นเมื่อสัญญาณถูกส่งออกมาจาก UART จึงต้องเข้าสู่วงจรขับเพื่อปรับระดับแรงดันให้ได้ระดับสัญญาณตามมาตราฐาน RS-232 ก่อนถูกส่งออกไปยังคอมพิวเตอร์ สำหรับอุปกรณ์ต่อเชื่อมปลายทางก็จะต้องมีวงจรขับในระดับนี้เช่นกัน เพื่อให้ได้ระดับสัญญาณในระดับเดียวกัน แต่วงจรขับที่ใช้ทั้งภายในคอมพิวเตอร์และอุปกรณ์ต่อเชื่อมปลายทางนั้นจะถูกกลับสถานะ ซึ่งสามารถแสดงได้ดังรูปที่ 4.7 โดยที่พอร์ตอนุกรมจะมีแอดเดรส อยู่ 4 ตำแหน่ง คือ COM 1 : 3F8H, COM 2 : 2E8H, COM 3 : 3E8H, COM 4 : 2E8H



รูปที่ 4.7 แสดงโครงสร้างทางฮาร์ดแวร์ของพอร์ตอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
ไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ (Microcontroller) เป็นชื่อของอุปกรณ์อิเล็กทรอนิกส์อย่างหนึ่ง ที่รวมเอาหน่วยประมวลผล หน่วยคำนวณทางคณิตศาสตร์ และลอจิก วงจรรับสัญญาณอินพุต วงจรขับสัญญาณเอาท์พุต หน่วยความจำ และวงจรกำเนิดสัญญาณนาฬิกาไว้ด้วยกัน ทำให้สามารถนำไปใช้งานแทนวงจรอิเล็กทรอนิกส์ที่ซับซ้อนได้เป็นอย่างดี ช่วยลดจำนวนอุปกรณ์ และขนาดของระบบในขณะที่ขีดความสามารถสูงขึ้นภายใต้งบประมาณที่เหมาะสม

ไมโครคอนโทรลเลอร์มาจากคำ 2 คำรวมกัน คือ "ไมโคร" (Micro) ซึ่งหมายถึง ไมโครโปรเซสเซอร์ (Microprocessor) ซึ่งเป็นอุปกรณ์ประมวลผลขนาดเล็ก ภายในประกอบด้วย หน่วยประกอบผลกลางหรือซีพียู (CPU : Central Processing Unit) หน่วยคำนวณทางคณิตศาสตร์และลอจิก (ALU : Arithmetic Logic Unit) วงจรเชื่อมต่อหน่วยความจำ และวงจรสัญญาณนาฬิกาอีกคำหนึ่ง คือคำว่า "คอนโทรลเลอร์" (Controller) หมายถึง อุปกรณ์ควบคุม ดังนั้นไมโคอนโทรลเลอร์ จึงเป็นอุปกรณ์ที่ใช้ในการควบคุม โดยสามารถเขียนโปรแกรมเพื่อที่จะกำหนดรูปแบบการควบคุมได้อย่างอิสระ

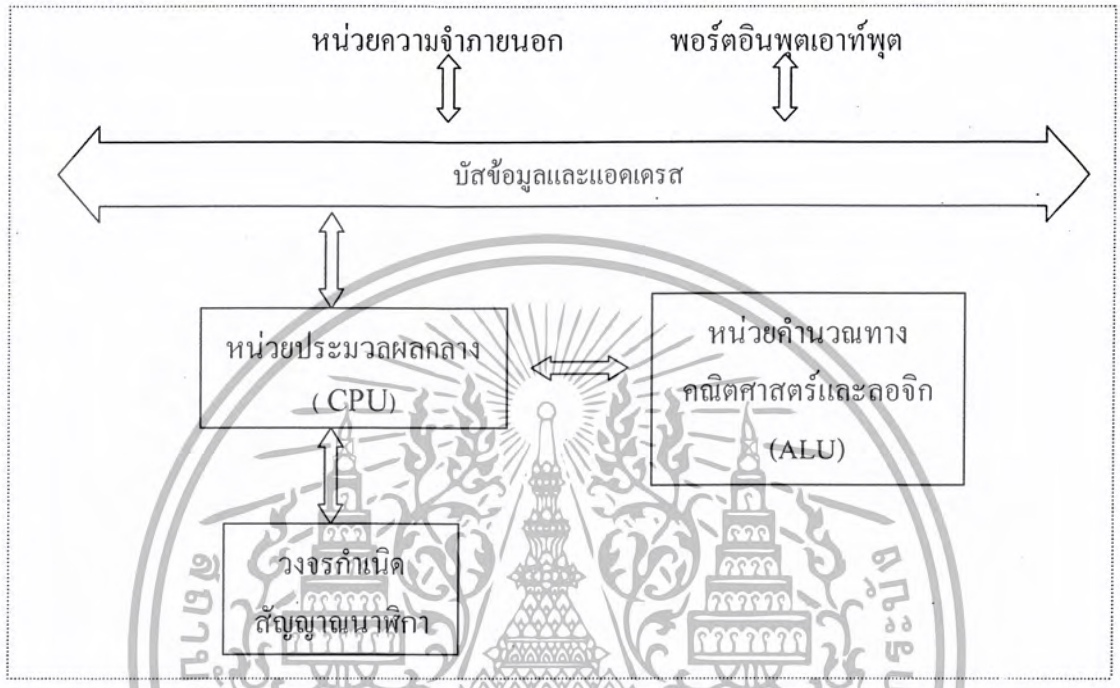
ความแตกต่างของไมโครโปรเซสเซอร์กับไมโครคอนโทรลเลอร์

ในรูปที่ 5.1 แสดงโครงสร้างของไมโครโปรเซสเซอร์ ซึ่งประกอบไปด้วยหน่วยประมวลผลกลาง หน่วยคำนวณทางคณิตศาสตร์ และลอจิก บัสข้อมูล และแอดเดรสที่ใช้สำหรับการติดต่อกับหน่วยความจำภายนอก และวงจรกำเนิดสัญญาณนาฬิกา นั้นหมายความว่า การใช้งานของไมโครโปรเซสเซอร์จะต้องเชื่อมต่อกับหน่วยจำโปรแกรมภายนอก และถ้าหากต้องการเชื่อมต่ออุปกรณ์อินพุท และเอาต์พุตต้องอาศัยอุปกรณ์ที่เรียกว่า " ไอซีขยายพอร์ต " (Port Expander) ทำให้การสร้างระบบควบคุมจึงต้องอาศัยอุปกรณ์จำนวนมากส่งผลให้ขนาดของระบบควบคุมใหญ่ขึ้น

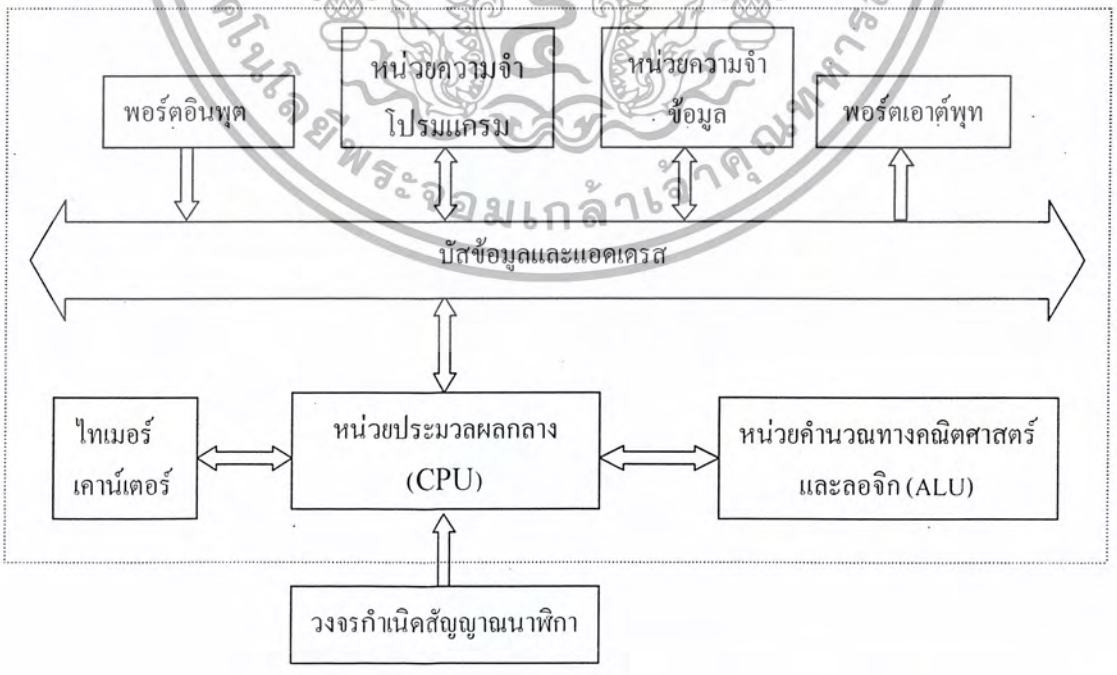
ในรูปที่ 5.2 แสดงโครงสร้างของไมโครคอนโทรลเลอร์ จะเห็นได้ว่าภายในของไมโครโทรลเลอร์มีอุปกรณ์พื้นฐานเหมือนไมโครโปรเซสเซอร์ เพียงแต่จะบรรจุหน่วยความจำโปรแกรม หน่วยความจำข้อมูล และพอร์ตอินพุตเอาต์พุตไว้ภายในพร้อมสรรพ ผู้ใช้เพียงแต่เขียนโปรแกรมลงบนตัวไมโครคอนโทรลเลอร์แล้วต่ออุปกรณ์วงจรกำเนิดสัญญาณนาฬิกา อาทิ คริสตอล ตัวเก็บประจุเป็นต้น สุดท้ายเชื่อมต่ออุปกรณ์อินพุตเอาต์พุตเข้ากับขาพอร์ตเพียงเท่านี้ก็สามารถใช้งานได้แล้วอย่างไรก็ตามไมโครคอนโทรลเลอร์ก็สามารถเชื่อมต่อกับหน่วยจำภายนอกได้ โดยพิจารณา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ให้หน่วยจำภายนอกนั้นเป็นอุปกรณ์อินพุตเอาต์พุตตัวหนึ่ง แล้วใช้ขาพอร์ตที่มีอยู่ทำการติดต่อตาม ตารางที่ 5.1 เป็นตารางที่แสดงความแตกต่างของไมโครคอนโทรลเลอร์ และไมโครโปรเซสเซอร์



รูปที่ 5.1 โครงสร้างพื้นฐานของไมโครโปรเซสเซอร์



รูปที่ 5.2 โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติ	ไมโครโปรเซสเซอร์	ไมโครคอนโทรลเลอร์
ขนาดของหน่วยประมวลผลกลาง	ไม่น้อยกว่า 8 บิต	ส่วนใหญ่ขนาด 8 บิต
หน่วยคำนวณทางคณิตศาสตร์และลอจิก	มีอยู่ภายใน	มีอยู่ภายใน
วงจรกำเนิดสัญญาณนาฬิกา	มีอยู่ภายใน	มีอยู่ภายใน
การเชื่อมต่อกับหน่วยความจำโปรมแกรม	เชื่อมต่อภายนอกเท่านั้น	ใช้ได้ทั้งภายนอกและภายใน
การเชื่อมต่อกับหน่วยความจำข้อมูล	เชื่อมต่อภายนอกเท่านั้น	ใช้ได้ทั้งภายนอกและภายใน
การเชื่อมต่อกับพอร์ตอินพุตเอาต์พุต	เชื่อมต่อภายนอกเท่านั้น	มีอยู่ภายในและสามารถขยายได้
ไทเมอร์ / เคาน์เตอร์	ไม่มีอยู่ในซีพียูขนาดเล็ก	มีอย่างน้อย 1 ตัว ขนาด 8-16 บิต
วอดช์ด็อกไทเมอร์	ไม่มีอยู่ในซีพียูขนาดเล็ก	มีอย่างน้อย 1 ตัว
จำนวนขาที่ใช้งาน	ไม่น้อยกว่า 40 ขา	มีตั้งแต่ 8 ขาขึ้นไป

ตารางที่ 5.1 เปรียบเทียบคุณสมบัติของไมโครคอนโทรลเลอร์ และไมโครโปรเซสเซอร์

ดังนั้นจึงสามารถสรุปได้ว่า ไมโครคอนโทรลเลอร์เหมาะสำหรับการสร้างระบบควบคุม ส่วนไมโครโปรเซสเซอร์เหมาะสำหรับการสร้างระบบประมวลผลข้อมูลความเร็วสูง และระบบควบคุมที่มีขนาดใหญ่มากๆ ถ้าต้องใช้กับข้อมูลที่มากกว่า 8 บิต ตลอดเวลา และต้องการความเร็วในการทำงานที่สูงๆ สามารถที่จะติดต่อกับหน่วยความจำได้เป็นจำนวนมากๆ ติดต่อกับอุปกรณ์อินพุตเอาต์พุตได้มากมายเวลาเดียวกันควรเลือกไมโครโปรเซสเซอร์ แต่ถ้าเป็นระบบควบคุมขนาด 8 บิตที่เชื่อมต่อกับอุปกรณ์ไม่มากนักควรใช้ไมโครคอนโทรลเลอร์

5.1 สถาปัตยกรรมและโครงสร้างของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

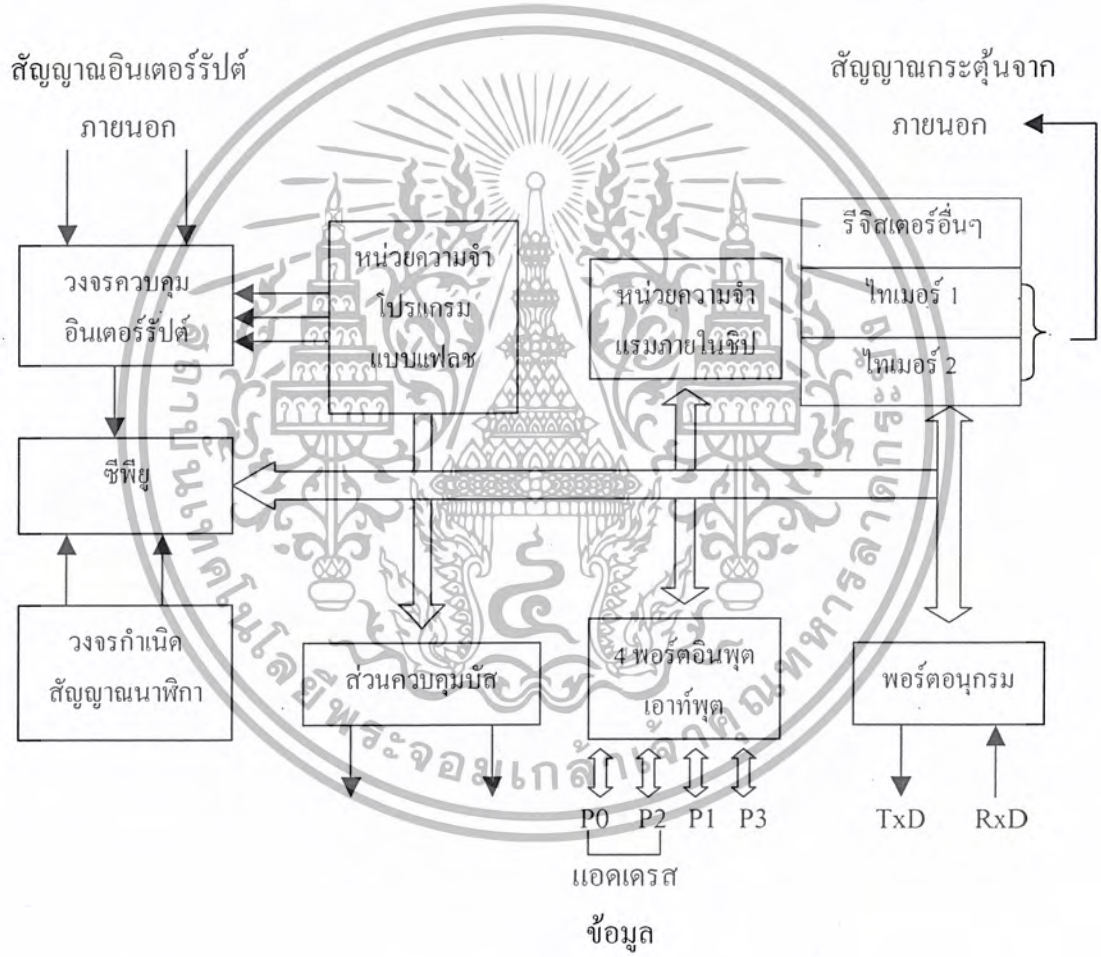
คุณสมบัติของไมโครคอนโทรลเลอร์ตระกูล MCS-51 อนุกรม AT 89Cxx

- เป็นไมโครคอนโทรลเลอร์ที่ใช้ซีพียูขนาด 8 บิต
- ภายในมีหน่วยความจำแบบแฟลชสามารถลบและเขียนใหม่ได้นับพันครั้ง
- ขาพอร์ตเป็นแบบสองทิศทาง สามารถใช้งานเป็นทั้งอินพุตและเอาท์พุต
- มีวงจรสื่อสารอนุกรมแบบฟลูดูเพล็กซ์
- มีไทเมอร์/เคาน์เตอร์ขนาด 16 บิตอย่างน้อย 2 ตัว
- สามารถรองรับแหล่งกำเนิดอินเตอร์รัปต์ได้ 6 ประเภท
- มีวงจรกำเนิดสัญญาณนาฬิกาอยู่ภายในชิป
- มีหน่วยความจำพื้นฐานเป็นหน่วยจำแบบแรม ในบางตัวมี EPROMเพิ่ม
- 1 Machine จะใช้เวลา 1 ไมโครวินาที โดยใช้ X-TAL 11.0592 MHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สามารถกำหนดการใช้งานพอร์ต I/O ได้ในระดับไบต์หรือบิตได้โดยตรง

ในรูปที่ 5.3 เป็นโครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ MCS - 51 แบบอนุกรม AT89Cxx จะเห็นได้ว่าโครงสร้างของ AT89Cxx จะเหมือนกับไมโครคอนเลอร์ตระกูล MCS-51 พื้นฐานหากแต่แตกต่างกันเฉพาะหน่วยจำโปรแกรมแบบแฟลชที่ได้เพิ่มเติมเข้ามา แต่ถ้าเป็นไมโครคอนโทรลเลอร์ในอนุกรม 87xx หน่วยความจำโปรแกรมภายในจะเป็นแบบอีพรอม และบางเบอร์สามารถโปรแกรมได้เพียงครั้งเดียว



รูปที่ 5.3 โครงสร้างของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชอนุกรมAT89Cxx

5.2 การจัดขาของไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์ MCS-51ทุกเบอร์จะมีสถาปัตยกรรมและขาใช้งานพื้นฐานเหมือนกันดังแสดงดังรูปที่ 5.3 และรูปที่ 5.4 โดยมีรายละเอียดขั้นต้น ดังนี้

ขา VCC ใช้สำหรับต่อไฟเลี้ยง +5v

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขา GND เป็นขากราวน์ สำหรับต่อกับกราวน์ของระบบ

ขาพอร์ต 0 (P0.0 – P0.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาท์พุตสำหรับใช้งานทั่วไป หากต้องการกำหนดให้ขาพอร์ต 0 ขาใดขาหนึ่งเป็นอินพุตสามารถทำได้โดยการเขียนข้อมูล **"1"** ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขานั้นมีสถานะปล่อยลอย (float) จึงมีอินพุตอิมพีแดนซ์สูงสามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ตนี้ยังจะถูกใช้งานติดต่อกับขาแอดเดรสไบต์ต่ำของหน่วยความจำภายนอก (A0-A7) และขาข้อมูล (D0-D7) โดยใช้กระบวนการมัลติเพล็กซ์เข้าช่วยเพื่อสลับการทำงานเป็นได้ทั้งขาติดต่อแอดเดรสและขาข้อมูล

ขาพอร์ต 1 (P1.0 – P1.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาท์พุตสำหรับใช้งานทั่วไปถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุตสามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย

ขาพอร์ต 2 (P2.0 – P2.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาท์พุตสำหรับใช้งานทั่วไปถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุตสามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาของพอร์ตมีสถานะปล่อยลอย (float) จึงมีอินพุตอิมพีแดนซ์สูง สามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ตนี้ยังถูกใช้งานติดต่อกับขาแอดเดรสไบต์สูงของหน่วยความจำภายนอก (A8 – A15)

ขาพอร์ต 3 (P3.0 –P3.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาท์พุตสำหรับใช้งานทั่วไปถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุตสามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องติดต่อส่งผลให้ขาของพอร์ตนั้นมีสถานะปล่อยลอย(flaot) จึงมีอินพุตอิมพีแดนซ์สูง สามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนี้ขาของพอร์ต 3 ยังเป็นขา ที่ทำหน้าที่การใช้งานพิเศษ ดังมีรายละเอียดขั้นต้นดังนี้

P3.0 ใช้เป็นขาอินพุตสำหรับรับข้อมูลจากการสื่อสารแบบอนุกรมหรือขา RxD

P3.1 ใช้เป็นขาอินพุตสำหรับส่งข้อมูลจากการสื่อสารแบบอนุกรมหรือขา TxD

P3.2 ใช้เป็นขาอินพุตรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 0 หรือขา INT0

P3.3 ใช้เป็นขาอินพุตรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 1 หรือขา INT1

P3.4 ใช้เป็นขาอินพุตสำหรับรับสัญญาณไทเมอร์จากภายนอกช่อง 0 หรือ ขา T0

P3.5 ใช้เป็นขาอินพุตสำหรับรับสัญญาณไทเมอร์จากภายนอกช่อง 1 หรือ ขา T1

P3.6 ใช้เป็นขาสัญญาณ WR ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก

P3.7 ใช้เป็นขาสัญญาณ RD ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขารีเซต (RESET) ใช้ในการรีเซตการทำงานของไมโครคอนโทรลเลอร์ โดยในการป้อนสัญญาณเพื่อรีเซตสถานะที่ขานี้ต้องอยู่ในระดับรีเซตอย่างน้อย2 แมชชีนไซเคิล โดยที่วงจรกำเนิดสัญญาณนาฬิกายังคงทำงานต่อเนื่องไปอย่างปกติ

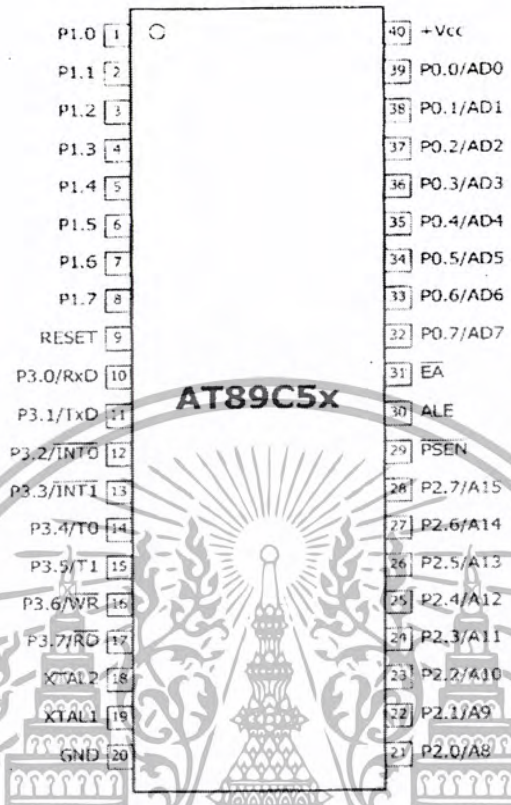
ขา ALE/PROG (Address Latch Enable/Program Pulse Input) เป็นขาที่ใช้ในการแลตซ์ของขาพอร์ต 0 เมื่อมีการใช้งานหน่วยความจำภายนอก นอกจากนี้ขานี้ยังใช้เป็นขาสำหรับรับพัลส์ของการโปรแกรมลงในไมโครคอนโทรลเลอร์ MCS-51 ในรุ่นที่มีการโปรแกรมเป็นแบบอีพรอม

ขา PSEN (Program Store Enable) ขานี้จะใช้ในการส่งสัญญาณเพื่อร้องขอการติดต่อกับหน่วยความจำโปรแกรมภายนอก เมื่อตัวไมโครคอนโทรลเลอร์ต้องการอ่านข้อมูลจากหน่วยความจำโปรมแกรมภายนอกตัวไมโครคอนโทรลเลอร์ตัวของไมโครคอนโทรลเลอร์ จะส่งสัญญาณออกมาที่ขานี้ 2 ครั้งในแต่ละแมชชีนไซเคิล แต่ถ้าต้องการติดต่อกับหน่วยความจำข้อมูลภายนอกขานี้จะไม่มีการส่งสัญญาณใดๆออกมา

ขา EA/VPP (External Access enable/Programming voltage Input) ใช้สำหรับเลือกติดต่อกับหน่วยความจำโปรมแกรมจากภายนอกหรือภายในตัวไมโครคอนโทรลเลอร์ ถ้าขานี้มีลอจิก **"0"** เป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำโปรมแกรมภายนอก แต่ถ้าขานี้มีลอจิก "1" เป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำภายในตัวของไมโครเอง นอกจากนี้ขานี้ยังใช้เป็นขาอินพุตสำหรับรับแรงดันไฟสูง สำหรับการโปรมแกรมหน่วยความจำภายในตัวของไมโครคอนโทรลเลอร์ สำหรับไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชต้องการแรงดันสำหรับการโปรแกรม +12 volt

ขา XTAL1 และ XTAL2 เป็นขาสำหรับต่อคริสตอลเพื่อสร้างสัญญาณนาฬิกาที่ใช้ในการกำหนดจังหวะการทำงานของไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



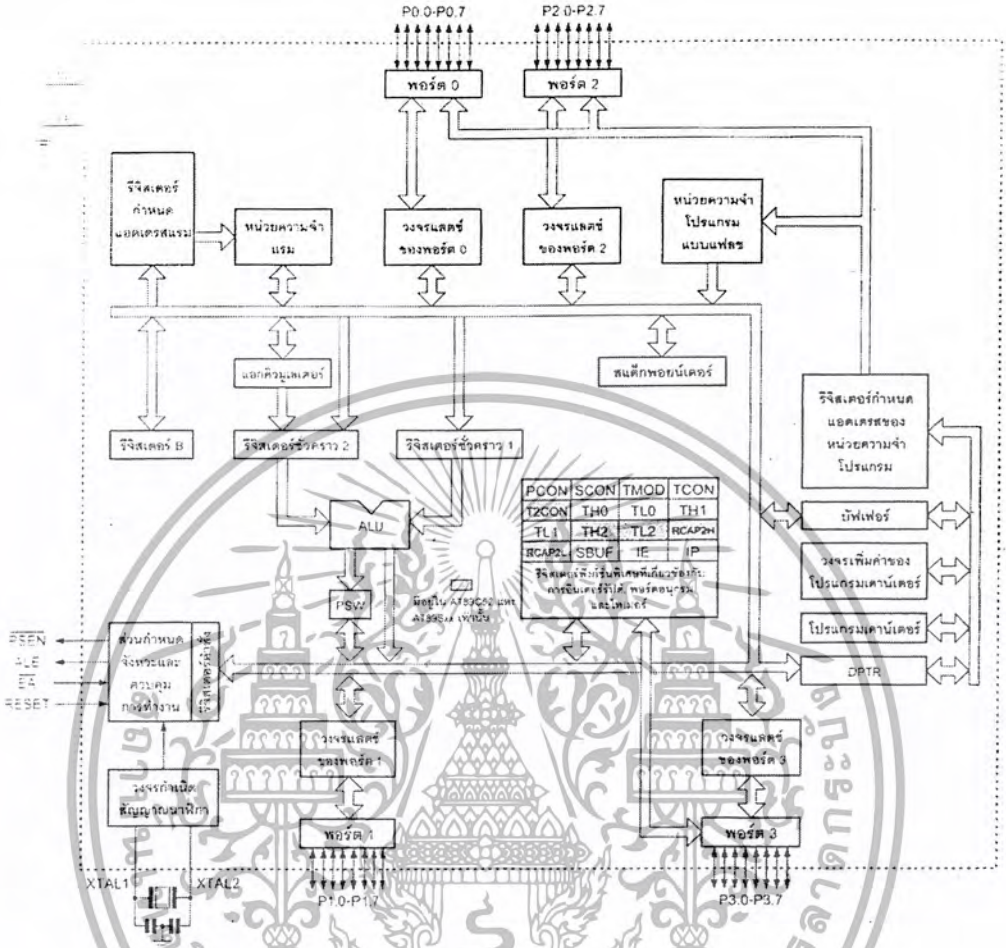
รูปที่ 5.4 การจัดขามาตราฐานของไมโครคอนโทรลเลอร์ MCS-51

5.3 โครงสร้าง และการทำงานของพอร์ต

ไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชมีพอร์ตให้ใช้งานทั้งสิ้น 4 พอร์ตคือ พอร์ต 0 ถึงพอร์ต 3 แต่ละพอร์ตมีขนาด 8 บิตเป็นพอร์ตแบบ 2 ทิศทาง กล่าวคือ สามารถเป็นได้ทั้งอินพุต สำหรับรับสัญญาณข้อมูลเข้า และเอาท์พุตสำหรับส่งสัญญาณข้อมูลออกทุกพอร์ตของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชจะมีวงจรแลตช์และวงจรขับตลอดจนบัฟเฟอร์อินพุตดังแสดงดังรูปที่ 5.5

ที่พอร์ต 0 และ พอร์ต 2 จะใช้เป็นพอร์ตอินพุต และเอาท์พุตสำหรับใช้งานทั่วไป และใช้ในการติดต่อกับหน่วยความจำภายนอก สำหรับพอร์ต 3 และพอร์ต 1 บางขานอกจากจะใช้เป็นขาพอร์ตอินพุตและเอาท์พุตตามปกติแล้ว ยังสามารถนำมาใช้งานเพื่อทำหน้าที่พิเศษได้

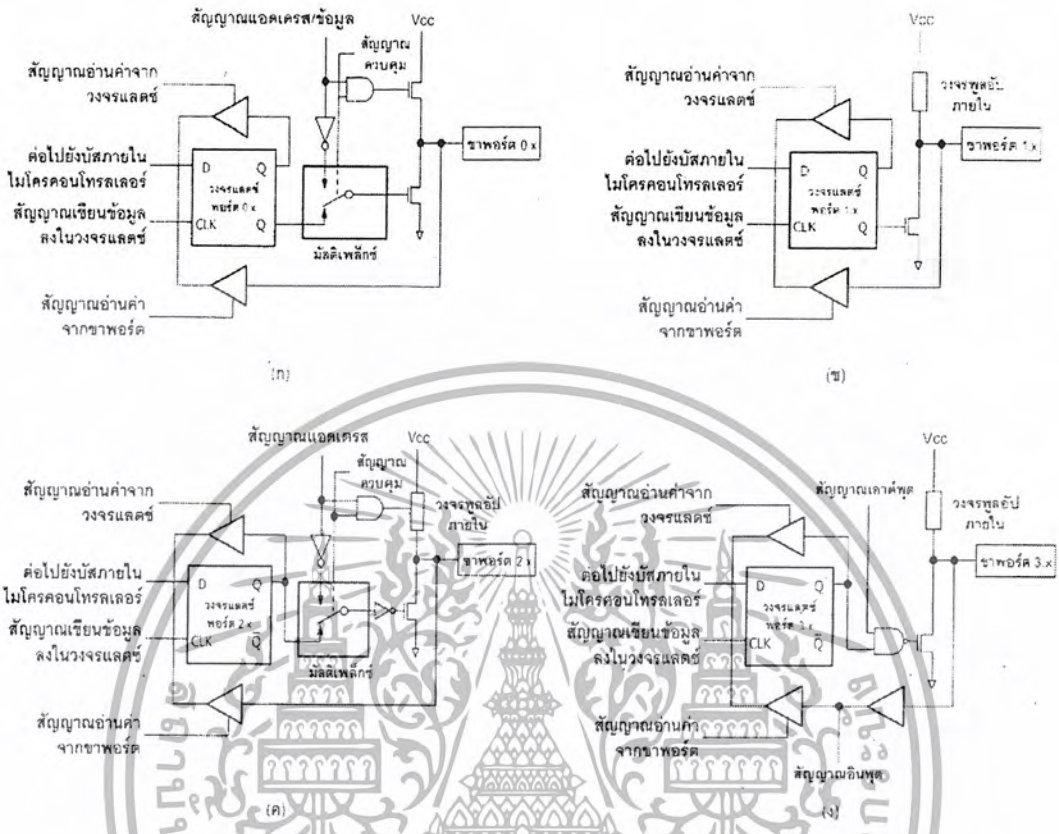
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.5 รายละเอียดโครงสร้างหลักของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

ในรูปที่ 5.6 (ก) วงจรของพอร์ต 0 วงจรแลตช์ของแต่ละบิตในแต่ละพอร์ต ก็คือ วงจรดีฟลิฟลอปนั่นเอง การอ่านค่าสถานะของพอร์ตและวงจรแลตช์สามารถกระทำได้อย่างอิสระด้วยซึ่งสัญญาณที่แยกจากกัน นั่นคือสัญญาณอ่านข้อมูลจากขาพอร์ต และสัญญาณอ่านข้อมูลจากวงจรแลตช์ ส่วนการเขียนข้อมูลมายังพอร์ตต้องส่งสัญญาณมายังขา CLK ของดีฟลิฟลอป ในขณะที่ข้อมูลจะผ่านมาทางบัสข้อมูลภายในเข้าสู่ขา D ของดีฟลิฟลอปที่พอร์ตนี้มีวงจรมัลติเพล็กซ์สำหรับกำหนดลักษณะการทำงานของพอร์ตว่าต้องการใช้งานเป็นขาพอร์ตอินพุทเอาท์พุตปกติ หรือใช้ติดต่อหน่วยความจำภายนอกไมโครคอนโทรลเลอร์ เนื่องจากที่ขาพอร์ต 0 ไม่มีวงจรฟูลอัปภายในหากมีการนำพอร์ต 0 ไปใช้งานเป็นพอร์ตอินพุทจะต้องต่อตัวต้านทานฟลูอัปภายนอกเข้าที่ขาพอร์ต 0 ทุกขา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

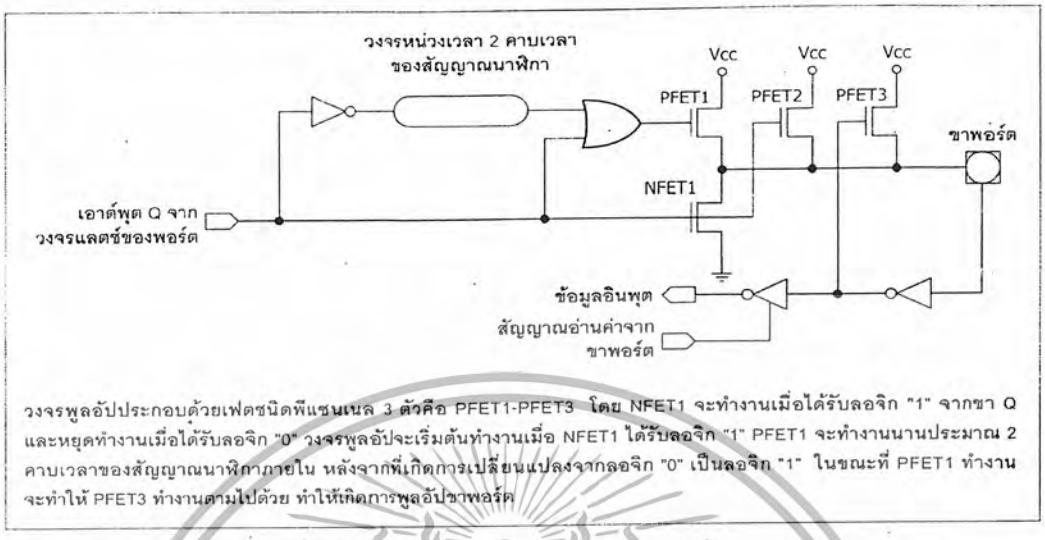


รูปที่ 5.6 แสดงวงจรภายในแต่ละพอร์ตของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

ในรูปที่ 5.6 (ข) เป็นวงจรของพอร์ต 1 ซึ่งมีลักษณะทั่วไปคล้ายกับพอร์ต 0 หากแต่ไม่มีวงจรมัลติเพล็กซ์ เนื่องจากพอร์ตนี้จะไม่ใช้ติดต่อกับหน่วยความจำภายนอกแต่จะมีวงจรฟลูอัปในแต่ละพอร์ตของพอร์ตนี้แทน สำหรับรายละเอียดของวงจรฟูลอัปแสดงดังรูปที่ 5.7

ในรูปที่ 5.6 (ค) เป็นวงจรภายในของพอร์ต 2 จะคล้ายกับพอร์ต 0 มาก ต่างกันเพียงมีวงจรฟูลอัปเพิ่มเติมเข้ามา ส่วนในรูป 5.6 (ง) เป็นวงจรภายในของพอร์ต 3 จะเห็นได้ว่าคล้ายกับพอร์ต 1 มีการเพิ่มเติมวงจรบัฟเฟอร์ และวงจรอินพุทเอาท์พุต เมื่อทำงานในฟังก์ชั่นพิเศษเข้ามา เนื่องจากที่พอร์ต 3 สามารถนำไปใช้งานในหน้าที่พิเศษได้ทุกขา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.7 วงจรพูลอัปภายในพอร์ตของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

การใช้งานเป็นพอร์ตอินพุต

เนื่องจากพอร์ตทั้งหมดของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชสามารถเป็นได้ทั้งอินพุตและเอาท์พุต ดังนั้นจึงมีความจำเป็นอย่างยิ่งต้องความเข้าใจถึงการกำหนดลักษณะการทำงานให้แก่พอร์ตของไมโครคอนโทรลเลอร์ MCS-51แบบแฟลช

ในการกำหนดให้เป็นพอร์ตอินพุตต้องเริ่มต้นการเขียนข้อมูล **"1"** มายังแต่ละบิตที่ต้องการใช้การใช้เป็นอินพุต เพื่อหยุดการทำงานของเฟตที่ใช้ในการขับสัญญาณเอาท์พุตของบิตนั้นๆ ทำให้ขาสัญญาณของพอร์ตที่จะเชื่อมต่อเข้ากับวงจรฟลูอัปภายในโดยตรง ส่งให้ขาพอร์ตมีลอจิกเป็น "1" สามารถที่จะรับสัญญาณลอจิก **"0"** จากอุปกรณ์ภายนอกได้ สัญญาณข้อมูลจากอุปกรณ์ภายนอกจะถูกส่งเข้ามาและเก็บไว้ในวงจรบัฟเฟอร์ภายในพอร์ตแล้วรอให้ซีพียูมาอ่านค่าเข้าไปเมื่อเป็นเช่นนี้ อุปกรณ์ภายนอกที่เชื่อมต่อกับพอร์ตอินพุตของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชควรที่จะกำหนดให้ทำงานในสภาวะลอจิก **"0"** จะดี และสะดวกที่สุด (ซึ่งในปัจจุบันอุปกรณ์อินพุตที่เชื่อมต่อไมโครคอนโทรลเลอร์แทบทั้งหมดทำงานที่ลอจิก **"0"** แล้ว)

การใช้งานเป็นพอร์ตเอาท์พุต

โดยปกติแล้วขาพอร์ตจะกำหนดให้มีลักษณะเป็นเอาท์พุตอยู่แล้ว ดังนั้นจึงสามารถส่งข้อมูลออกไปได้อย่างง่ายดาย และตรงไปตรงมา กล่าวคือ เมื่อต้องการส่งข้อมูล **"0"** ไปยังวงจรแลตช์ ซึ่งก็ส่งผลต่อไปขับเฟตทำให้เฟตที่ขาพอร์ตจะทำงานเมื่อได้รับลอจิก **"0"** ขึ้น ในทางตรงข้ามหากต้องการส่งข้อมูล **"1"** ออกไปก็ให้เขียนข้อมูล **"1"** ไปยังวงจรแลตช์ วงจรขับก็จะหยุดการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำให้ขาพอร์ตที่เชื่อมต่อกับวงจรฟลูอัปภายในเกิดเป็นลอจิก **"1"** ที่ขาพอร์ตนั้นซึ่งจะคล้ายกับเป็นการกำหนดให้เป็นขาอินพุตมากเพียงแต่แตกต่างกันตรงที่กระบวนการในการเคลื่อนย้ายข้อมูล โดยถ้าเป็นอินพุตจะมีสัญญาณมาอ่านข้อมูลที่บัฟเฟอร์ แต่ถ้าเป็นเอาท์พุตจะไม่มีการอ่านข้อมูลที่บัฟเฟอร์แต่อย่างใดเว้นแต่ในกรณีที่ต้องการตรวจสอบข้อมูลที่ส่งออกมาทางเอาท์พุต

เมื่อใช้งานเป็นพอร์ตเอาท์พุตแต่ละขา (หรือแต่ละบิต)ของแต่ละพอร์ตจะมีความสามารถในการจ่ายกระแสหรือกระแสซอร์ส (Source Current) ได้สูงสุด 10 mA และทุกขารวมกันแต่ละพอร์ต (ทั้ง 8 บิต) สูงสุด 26 mA สำหรับพอร์ต 0 และ 15mA สำหรับพอร์ต 1, 3 ในกรณีที่ใช้งานทุกพอร์ต เอาท์พุตจะสามารถจ่ายกระแสได้รวมกันสูงสุด 71 mA ดังนั้นการใช้งานเป็นพอร์ตเอาท์พุต เพื่อไม่ให้เกิดปัญหาเกี่ยวกับความสามารถในการจ่ายกระแสจึงควรจะต่อวงจรบัฟเฟอร์ของเอาท์พุต เพื่อช่วยในการขับกระแสอีกทางหนึ่ง

การอ่านค่าลอจิกจากพอร์ต

ในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช สามารถอ่านค่าลอจิกจากพอร์ตได้ทั้ง 2 ลักษณะคือ อ่านค่าจากพอร์ตโดยตรงและอ่านค่าจากวงจรแลตช์ของแต่ละพอร์ต ในกรณีที่ขาเบสของทรานซิสเตอร์ชนิด NPN และขาอิมิตเตอร์ของทรานซิสเตอร์ตัวนั้นต่อลงกราวน์ทำให้หากอ่านค่าลอจิกที่ขาพอร์ตจะได้ผลตรงข้ามกับที่ส่งออกมา แต่หากทำงานโดยอ่านค่าลอจิกที่วงจรแลตช์จะได้ค่าที่ตรงกับค่าที่ต้องการส่งจริง ดังนั้นในการค่าลอจิกจากพอร์ตจึงควรเลือกวิธีการให้เหมาะสมกับอุปกรณ์ที่นำมาต่อด้วย

5.4 หน่วยความจำของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

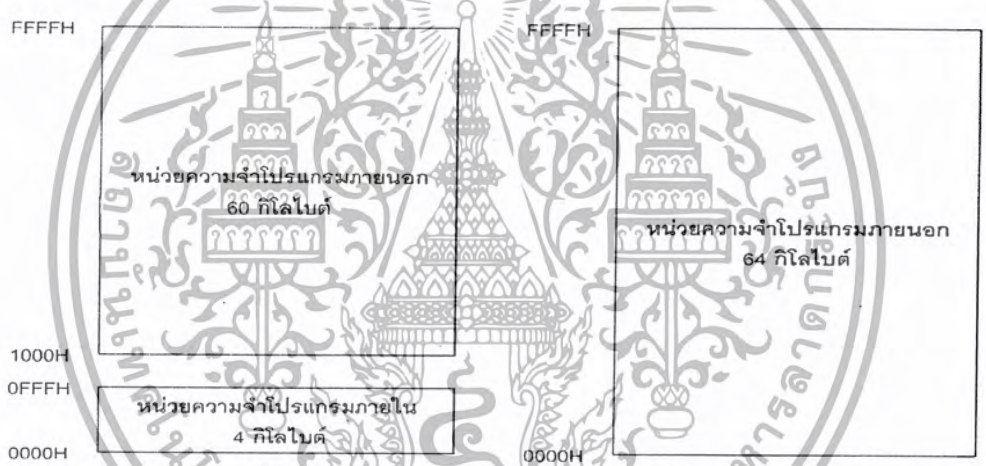
ในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชมีหน่วยความจำภายในหลักๆ อยู่ 2 ส่วน คือ หน่วยความจำโปรแกรมและหน่วยความจำข้อมูล ซึ่งก็มีขนาดและการจัดสรรแตกต่างกันไปในแต่ละเบอร์ที่รวมถึงการเชื่อมต่อกับหน่วยความจำภายนอก และข้อมูลเบื้องต้นของรีจิสเตอร์ฟังก์ชั่นพิเศษที่ใช้ในการควบคุมการทำงานของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

หน่วยความจำโปรแกรม (Program memory)

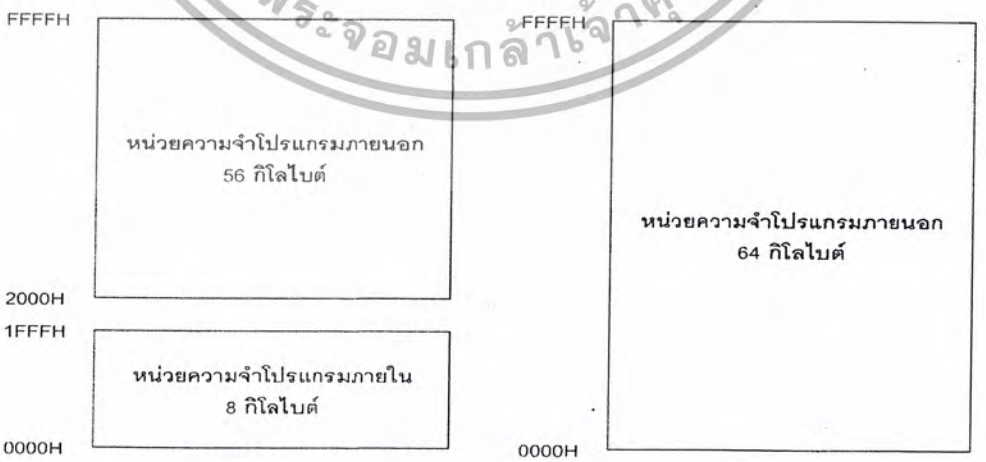
ในรูปที่ 5.8 แสดงการจัดหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช ในเบอร์ต่างๆที่นิยมใช้งานซึ่งประกอบด้วยเบอร์ AT89C51 และ AT89C52 จะเห็นว่าทั้ง 2 เบอร์สามารถติดต่อกับหน่วยความจำโปรแกรมสูงสุด 64 กิโลไบต์ โดยสามารถที่จะเลือกใช้หน่วยจำโปรแกรมภายในอย่างเดียว หรือรวมกับภายนอกหรือจะเลือกใช้หน่วยความจำภายนอกอย่างเดียวก็ได้ดังรูปที่ 5.8 (ก) โดยภายใน AT89C51 จะมีหน่วยความจำโปรแกรมภายใน 4 กิโลไบต์ ในขณะที่เบอร์ AT89C52 จะมีขนาด 8 กิโลไบต์ ในขณะที่ใช้หน่วยความจำภายใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รวมกับภายนอกเบอร์AT89C51จะสามารถติดต่อกับหน่วยความจำภายนอกได้ 60 กิโลไบต์แต่เบอร์ AT89C52จะสามารถ ติดต่อกับหน่วยความจำโปรแกรมภายนอกได้ 56 กิโลไบต์ ซึ่งหน่วยความจำโปรแกรมที่ใช้เก็บข้อมูลโปรแกรมเพื่อควบคุมการทำงานของไมโครคอนโทรลเลอร์นั้น เรียกว่า **"โปรมแกรมมอนิเตอร์"** (monitor program) หากใช้หน่วยความจำภายนอกมักจะบรรจุชนิดอีพรอม (EPROM : Erasable Programmable Read -Only Memory) ซึ่งสามารถทำการอ่านอย่างเดียว หน่วยความจำโปรแกรม จะมีแอดเดรสเริ่มต้นที่ 0000H เมื่อซีพียูได้รับการรีเซตให้เริ่มต้นทำงานจะต้องมาเริ่มต้นที่แอดเดรส 0000H นี้เสมอ อย่างไรก็ตามในพื้นทำงานของหน่วยความจำโปรแกรมไม่ว่าจะใช้งานจากภายนอก หรือภายในต้องมีการสงวนพื้นที่บางตำแหน่งเอาไว้สำหรับการอินเตอร์รัปต์ 6 ประเภท ประเภทละ 8 ไบต์



(ก) การจัดสรรหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์เบอร์ AT89C51



(ข) การจัดสรรหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์เบอร์ AT89C52

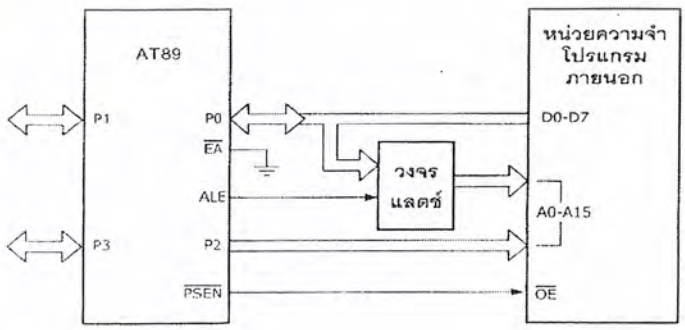
รูปที่ 5.8 การจัดหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พื้นที่สำหรับบริการอินเตอร์รัปต์ 0 จากภายนอก	กำหนดไว้ที่แอดเดรส 0003H
พื้นที่สำหรับบริการอินเตอร์รัปต์จากไทเมอร์ 0	กำหนดไว้ที่แอดเดรส 000BH
พื้นที่สำหรับบริการอินเตอร์รัปต์ 1 จากภายนอก	กำหนดไว้ที่แอดเดรส 0031H
พื้นที่สำหรับบริการอินเตอร์รัปต์จากไทเมอร์ 1	กำหนดไว้ที่แอดเดรส 001BH
พื้นที่สำหรับบริการอินเตอร์รัปต์ของการสื่อสารอนุกรม	กำหนดไว้ที่แอดเดรส 0023H
พื้นที่สำหรับบริการอินเตอร์รัปต์จากไทเมอร์ 2	กำหนดไว้ที่แอดเดรส 002BH

กรณีที่ใช้ไมโครคอนโทรลเลอร์ MCS-51แบบแฟลชที่หน่วยความจำโปรแกรมภายใน แต่ต้องการติดต่อกับหน่วยความจำโปรแกรมภายนอกด้วย สามารถทำได้โดยกำหนดแอดเดรสของหน่วยความจำโปรแกรมให้ต่อแอดเดรสสุดท้ายของหน่วยความจำโปรแกรมภายในตัวไมโครคอนโทรลเลอร์

ตัวอย่าง ไมโครคอนโทรลเลอร์ AT 89C51 มีหน่วยความจำโปรแกรมขนาด 4 กิโลไบต์ มีแอดเดรสอยู่ระหว่าง 0000H – 0FFFH เมื่อต่อหน่วยความจำโปรแกรมภายนอกต้องกำหนดให้แอดเดรสอยู่ในช่วง 1000H – FFFFH การต่อหน่วยความจำภายนอกแสดงดังรูปที่ 5.9 ขาพอร์ต P0.0 – P0.7 ใช้เป็นขาข้อมูล D0-D7 และขาแอดเดรสไบต์ต่ำโดยผ่านวงจรแลตช์ซึ่งปกติใช้ไอซีเบอร์ 74HC53 และใช้สัญญาณ ALE และ PSEN ในการเลือกใช้งานขา P0.0-P0.7 เพื่อเป็นขาข้อมูล หรือขาแอดเดรส ใน ขณะที่ขา P2.0-P2.7 ใช้ในการเชื่อมต่อกับขาแอดเดรสไบต์สูง A8-A15 ดังนั้นเมื่อมีการเชื่อมต่อกับหน่วยความจำหน่วยความจำภายนอกไมโครคอนโทรลเลอร์ จะเลือกขาพอร์ตเพียง 16 บิต คือ ขาที่พอร์ต P1.0 -P1.7 และ P3.0 - P3.7

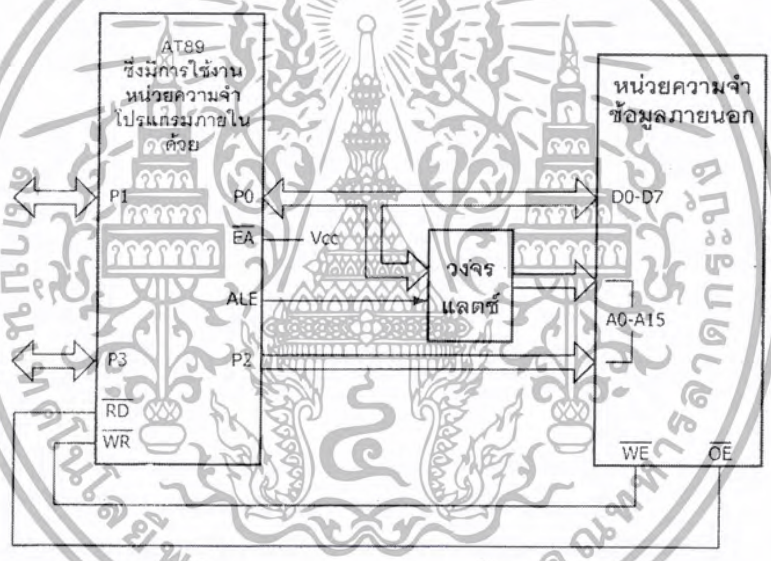


รูปที่ 5.9 การเชื่อมต่อหน่วยความจำโปรแกรมภายนอกของไมโครคอนโทรลเลอร์MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยความจำข้อมูล (DATA memory)

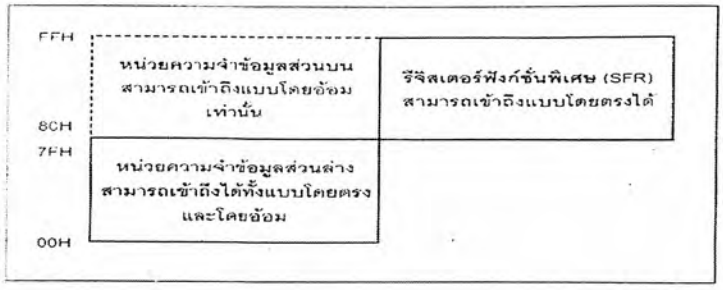
มีด้วยกัน 2 แบบ คือ หน่วยความจำข้อมูลภายใน และภายนอก โดยไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช ในอนุกรม AT89 สามารถที่จะติดต่อกับหน่วยความจำข้อมูลภายนอกได้สูงสุด 64 กิโลไบต์ โดยการใช้คำสั่ง MOVX ในการติดต่อกับหน่วยความจำข้อมูลภายนอกการติดต่อกับหน่วยความจำข้อมูลภายนอกของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช ดังแสดงในรูปที่ 5.10 จะเห็นได้ว่ามีลักษณะคล้ายกับการติดต่อกับหน่วยความจำโปรแกรมภายนอก แตกต่างกันที่มีสัญญาณที่ใช้สำหรับการอ่าน และการเขียนหน่วยความจำข้อมูลภายนอก นั่นคือ ขา RD และ WR



รูปที่ 5.10 การเชื่อมต่อหน่วยความจำข้อมูลภายนอกของไมโครคอนโทรลเลอร์ MCS-51

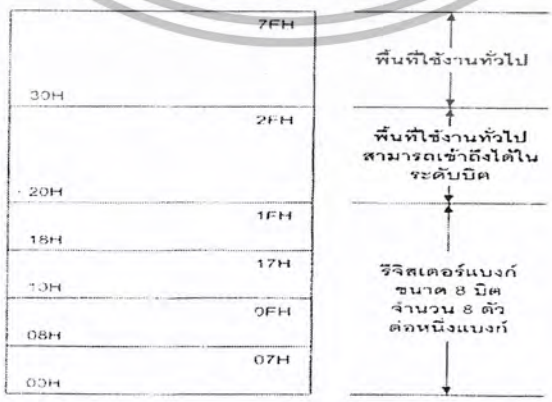
สำหรับไมโครคอนโทรลเลอร์ MCS-51แบบแฟลชในอนุกรม AT89 ทุกเบอร์จะมีหน่วยความจำข้อมูลภายในเป็นแบบแรม (RAM : Random Access Memory) โดยแต่ละเบอร์นั้นจะมีขนาดที่แต่ต่างกันไปในเบอร์ AT89C51 มีหน่วยความจำข้อมูลขนาด 128 ไบต์ ในขณะที่เบอร์ของ AT89C52 มีขนาด 256 ไบต์ สำหรับการจัดหน่วยความจำภายในแบ่งเป็น 3 ส่วนคือ หน่วยความจำข้อมูลส่วนล่าง (Lower) ส่วนบน (Upper) และรีจิสเตอร์พิเศษ (SFR : Special Function Register) แต่ละส่วนมีขนาด 128 ไบต์ ดังแสดงการจัดสรรในรูปที่ 5.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.11 การจัดสรรพื้นที่ของหน่วยความจำข้อมูลภายในไมโครคอนโทรลเลอร์MCS-51

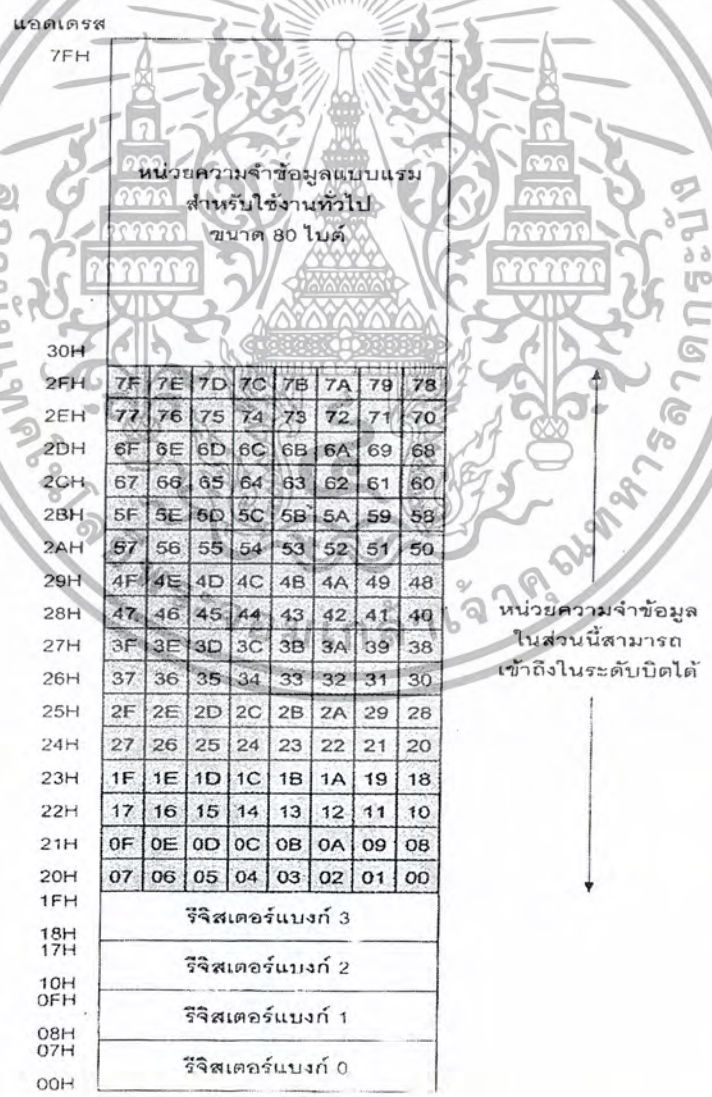
จะเห็นได้ว่าหน่วยความจำส่วนบนและรีจิสเตอร์ฟังก์ชั่นพิเศษมีตำแหน่งที่ทับซ้อนกัน แต่จะใช้การติดต่อที่แตกต่างกัน และไมโครคอนโทรลเลอร์ MCS-51 บางเบอร์จะไม่มีหน่วยความจำข้อมูลส่วนบนขนาดข้อมูลของหน่วยความจำของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช โดยแท้จริงมี 256 ไบต์ แต่ด้วยการเข้าถึงที่แตกต่างกันจึงดูเสมือนว่าไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชมีหน่วยความจำข้อมูลภายในสูงสุดถึง 384 ไบต์ โดยในหน่วยความจำข้อมูลส่วนล่างขนาด 128 ไบต์ มีแอดเดรสอยู่ที่ 00H-7FH สามารถเข้าถึงข้อมูลได้ทางตรงและทางอ้อม ส่วนหน่วยความจำข้อมูลส่วนบนขนาด 128 ไบต์ มีแอดเดรสอยู่ที่ 80H-FFH สามารถเข้าถึงข้อมูลทางอ้อมเท่านั้น ในขณะที่รีจิสเตอร์SFR มีแอดเดรสอยู่ที่ 80H-FFH เหมือนหน่วยความจำส่วนบน แต่สำหรับรีจิสเตอร์ SFR ใช้การเข้าถึงข้อมูลโดยตรง ดังนั้นเพื่อความสะดวกและง่ายตลอดจนป้องกันความสับสนในการเขียนโปรแกรมจึงควรใช้หน่วยความจำภายในเพียง 128 ไบต์ จากหน่วยความจำข้อมูลส่วนล่างร่วมกันกับรีจิสเตอร์SFR



รูปที่ 5.12 เป็นการจัดสรรพื้นที่ของหน่วยความจำข้อมูลภายในส่วนล่างของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 5.12 การแสดงการจัดสรรหน่วยความจำข้อมูลส่วนล่าง หน่วยความจำ 32 ไบต์ต่ำสุดที่แอดเดรส 00H–1FH แบ่งได้เป็น 4 กลุ่ม เรียกว่า 4 แบงก์ (bank) แต่ละแบงก์จะมีรีจิสเตอร์ 8 ตัว คือ R0–R7 การติดต่อกับหน่วยความจำในแบงก์ใดให้กำหนดที่รีจิสเตอร์ PSW (Program Status Word Register) หน่วยความจำข้อมูลขนาด 16 ไบต์ถัดมาที่แอดเดรส 20H – 2FH เป็นพื้นที่สำหรับใช้งานทั่วไปสามารถเข้าถึงข้อมูลได้ระดับบิต (Bit Addressable) และหน่วยความจำที่เหลือ 80 ไบต์ จะต้องแบ่งส่วนหนึ่งสำรองไว้เป็นพื้นที่ของสแต็ก (Stack : ที่พักข้อมูลช่วงคราวในกรณีที่ซีพียูกระโดดไปทำงานในโปรมแกรมย่อย) การเข้าถึงหน่วยความจำส่วนนี้ต้องใช้การเข้าถึงในระดับไบต์



รูปที่ 5.13 แสดงโครงสร้างหน่วยความจำข้อมูลภายในส่วนบนของไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 5.13 แสดงโครงสร้างของหน่วยความจำข้อมูลส่วนบน ซึ่งจะมีลักษณะคล้ายกับหน่วยความจำข้อมูลส่วนล่าง หากแต่ใน 80 ไบต์บนไม่จำเป็นต้องสำรองไว้สแต็ก และต้องใช้การเข้าถึงในลักษณะโดยอ้อมเท่านั้น

รีจิสเตอร์ฟังก์ชั่นพิเศษ (Special Function Register: SFR)

เป็นรีจิสเตอร์ที่ใช้ในการควบคุมการทำงานของไมโครคอนโทรลเลอร์ MCS - 51 มีด้วยกัน 22 ตัวสำหรับเบอร์ AT89C51 28 ตัว สำหรับ AT89C52 และอนุกรม AT89Sxx ทั้งนี้เนื่องจากใน AT89C52 และAT89Sxx มีจำนวนไทเมอร์เคาน์เตอร์มากกว่า AT89C51 รีจิสเตอร์SFR มีแอดเดรสอยู่ระหว่าง 80H – FFH ในพื้นที่ของหน่วยความจำส่วนบนสามารถเข้าถึงโดยตรง (Direct Addressing) ดังรูปที่ 5.14

แอดเดรส	บิต								
FFH									
F0H	B7	B6	B5	B4	B3	B2	B1	B0	รีจิสเตอร์ B
E0H	A7	A6	A5	A4	A3	A2	A1	A0	รีจิสเตอร์ ACC
D0H	D7	D6	D5	D4	D3	D2	-	D0	รีจิสเตอร์ PSW
B8H	-	-	-	D4	D3	D2	D1	D0	รีจิสเตอร์ IP
B0H	3.7	3.6	3.5	3.4	3.3	3.2	3.1	3.0	รีจิสเตอร์ P3
A8H	D7	-	-	D4	D3	D2	D1	D0	รีจิสเตอร์ IE
A0H	2.7	2.6	2.5	2.4	2.3	2.2	2.1	2.0	รีจิสเตอร์ P2
99H	ไม่สามารถเข้าถึงระดับบิตได้								รีจิสเตอร์ SBUF
98H	S7	S6	S5	S4	S3	S2	S1	S0	รีจิสเตอร์ SCON
90H	1.7	1.6	1.5	1.4	1.3	1.2	1.1	1.0	รีจิสเตอร์ P1
8DH	ไม่สามารถเข้าถึงระดับบิตได้								รีจิสเตอร์ TH1
8CH	ไม่สามารถเข้าถึงระดับบิตได้								รีจิสเตอร์ TH0
8BH	ไม่สามารถเข้าถึงระดับบิตได้								รีจิสเตอร์ TL1
8AH	ไม่สามารถเข้าถึงระดับบิตได้								รีจิสเตอร์ TL0
89H	ไม่สามารถเข้าถึงระดับบิตได้								รีจิสเตอร์ TMOD
88H	T7	T6	T5	T4	T3	T2	T1	T0	รีจิสเตอร์ TCON
87H	ไม่สามารถเข้าถึงระดับบิตได้								รีจิสเตอร์ PCON
83H	ไม่สามารถเข้าถึงระดับบิตได้								รีจิสเตอร์ DPH
82H	ไม่สามารถเข้าถึงระดับบิตได้								รีจิสเตอร์ DPL
81H	ไม่สามารถเข้าถึงระดับบิตได้								รีจิสเตอร์ SP
80H	0.7	0.6	0.5	0.4	0.3	0.2	0.1	0.0	รีจิสเตอร์ P0

หมายเหตุ : ชื่อของแต่ละบิตที่กำหนดในรูปเป็นการกำหนดให้เห็นว่ามีการเรียงลำดับนัยสำคัญของรีจิสเตอร์แต่ละตัว โดยเรียงจากบิตสูงมายังบิตต่ำ สำหรับชื่อที่แท้จริงของแต่ละบิต ให้ตรวจสอบกับรายละเอียดของรีจิสเตอร์ตัวนั้น ๆ ต่อไป!

รูปที่ 5.14 การจัดสรรพื้นที่ของรีจิสเตอร์ฟังก์ชั่นพิเศษ (SFR)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับรายละเอียดของรีจิสเตอร์ของSFR มีดังนี้

รีจิสเตอร์แสดงสถานะของโปรแกรม(Program Status Word : PSW)

เป็นรีจิสเตอร์ขนาด 8 บิต สามารถเข้าถึงข้อมูลได้ในระดับบิตจึงสามารถกำหนดค่าในแต่ละบิตของรีจิสเตอร์ตัวนี้ได้อย่างอิสระมีแอดเดรสอยู่ที่D0H ทำหน้าที่เก็บสถานะการทำงานของโปรแกรมในขณะนั้นจะเรียกสถานะต่างๆของโปรแกรมว่า **"แฟลก"** (Flag) เมื่อซีพียูกระทำคำสั่งทางคณิตศาสตร์และลอจิก แล้วเกิดการเปลี่ยนแปลงสถานะขึ้นผลของการเปลี่ยนแปลงนั้นจะมาปรากฏที่บิตต่างๆ ของรีจิสเตอร์ PSW รายละเอียดของแต่ละบิตในรีจิสเตอร์ PSW

บิต 7	บิต 6	บิต5	บิต 4	บิต3	บิต2	บิต1	บิต0
CY	AC	F0	RS1	RS0	OV	-	P

CY : แฟลกทด (Carry Flag) เป็น "1" เมื่อมีการกระทำทางคณิตศาสตร์และลอจิกแล้วค่าของแอกคิวมูเลเตอร์เกิน 255 (ฐานสิบ) หรือ FFH

AC : แฟลกทดเสริม(Auxiliary Carry flag) เป็น "1" เมื่อมีการกระทำทางคณิตศาสตร์แล้วทำให้เกิดการทดข้ามจากบิต 3 มายังบิต 4

F0 : แฟลกใช้งานทั่วไป เมื่อผู้เขียนโปรแกรมกำหนดค่าที่บิตนี้แล้ว ไม่ว่าจะกระทำคำสั่งใดๆที่บิตไม่มีการเปลี่ยนแปลง

RS1 : บิตเลือกรีจิสเตอร์แบงก์ (Register Select 1) ใช้งานร่วมกับบิต RS0 เพื่อเลือกแบงก์ของรีจิสเตอร์ R0 – R7

RS0 : บิตเลือกรีจิสเตอร์แบงก์(Register Select 0) ใช้งานร่วมกับบิต RS1 เพื่อเลือกแบงก์ของรีจิสเตอร์ R0 – R7

OV : บิตเกิน (Overflow) เป็น **"1"** เมื่อมีการกระทำทางคณิตศาสตร์และลอจิกแล้ว ทำให้เกิดการทดข้ามจากบิต 6 มายังบิต 7 ของแอกคิวมูเลเตอร์ หรือแอกคิวมูเลเตอร์มีค่าเกิน 127 (เลขฐานสิบ) นอกจากนั้นยังใช้เป็นการแสดงค่าลบด้วย

\- : บิตนี้สามารถใช้งานได้อย่างอิสระ

P : บิตพาริตี้ (Parity) ใช้ในการตรวจสอบจำนวนค่า **"1"** ภายในแอกคิวมูเลเตอร์ ถ้าหากในแอกคิวมูเลเตอร์มีจำนวนบิตที่เป็น **"1"** รวมกันเป็นเลขคู่ บิตนี้จะเป็น **"0"** ถ้ารวมเป็นเลขคี่ บิตนี้จะเป็น **"1"**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเห็นได้ว่านอกจากรีจิสเตอร์PSW ถูกใช้ในการเก็บสถานะของโปรแกรมแล้วที่บิต RS0 และRS1 ยังใช้ในการเลือกแบงค์ของหน่วยความจำส่วนล่างซึ่งเป็นพื้นที่ของรีจิสเตอร์ R0 – R7 ด้วยดังมีรายละเอียดแสดงในตาราง 5.2

RS0	RS1	แบงค์ของรีจิสเตอร์	ช่วงของแอดเดรส
0	0	แบงก์0	00H - 07H
0	1	แบงก์1	08H - 0FH
1	0	แบงก์2	10H - 17H
1	1	แบงก์3	18H- 1FH

ตาราง 5.2 แสดงการเลือกแบงก์ของหน่วยความจำส่วนล่างเพื่อติดต่อรีจิสเตอร์แบงก์R0-R7

โดยปกติแล้วในการใช้งานรีจิสเตอร์ R0 – R7 มักนิยมใช้แบงก์ 0 เป็นลำดับแรก หากไม่เพียงพอจึงเลือกในแบงก์อื่นๆมาใช้ แต่ต้องระมัดระวังในการกำหนดค่าและลำดับการติดต่อให้ดี มิฉะนั้นอาจให้การเขียนโปรแกรมเกิดความสับสนดังนั้นในการใช้งานควรเลือกใช้รีจิสเตอร์R0 – R7ในแบงก์ 0

แอกคิวมูเลเตอร์ (Accumulator: ACC)

เป็นรีจิสเตอร์ขนาด8 บิตมีแอดเดรสอยู่ที่ตำแหน่ง E0H เป็นรีจิสเตอร์ที่ใช้สำหรับเก็บข้อมูล หรือผลลัพธ์ที่ได้จากการทำงานของไมโครคอนโทรลเลอร์ โดยเฉพาะอย่างยิ่งการคำนวณทางคณิตศาสตร์ และถอจิกก่อนที่จะส่งข้อมูล หรือผลลัพธ์ที่ได้ให้แก่ซีพียู เพื่อทำการประมวลผลต่อไป อาจเรียกสั้นว่าๆ รีจิสเตอร์A หรือACC สามารถเข้าถึงข้อมูลในระดับบิตได้

รีจิสเตอร์B

มีขนาด 8 บิต มีแอดเดรสอยู่ที่ F0H มีหน้าที่พิเศษ คือหากต้องการคูณหรือหารทางคณิตศาสตร์ต้องนำข้อมูลที่ต้องการหารหรือคูณมาเก็บไว้ในรีจิสเตอร์B จึงกระทำคำสั่งการคูณหรือหารกับค่าในรีจิสเตอร์A ต่อไป ในกรณีไม่ต้องการคูณ หรือหารข้อมูลสามารถใช้รีจิสเตอร์B นี้ในการเก็บข้อมูลทั่วไปได้เหมือนกับรีจิสเตอร์ปกติ และสามารถเข้าถึงระดับบิตได้

โปรแกรมเคาน์เตอร์ (Program Counter: PC)

เป็นรีจิสเตอร์ขนาด16 บิต มีหน่วยที่ในการแจ้งแอดเดรสหน่วยความจำโปรแกรมในตำแหน่งถัดไปที่ซีพียูจะต้องไปใช้งาน รีจิสเตอร์PC เป็นรีจิสเตอร์ตัวเดียวที่ไม่ได้รวมกับรีจิสเตอร์ SFR ตัวอื่นๆ การเปลี่ยนแปลงค่าของรีจิสเตอร์PC จะขึ้นอยู่กับผลของการกระทำคำสั่งของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยความจำโปรมแกรมที่ถูกเขียนขึ้นมารีจิสเตอร์ PC มีความสำคัญมาก โดยเฉพาะอย่างยิ่งในการตรวจสอบการทำงานของโปรแกรมว่าดำเนินไปตามขั้นตอนตามที่กำหนดไว้หรือไม่

สแต็กพอยน์เตอร์ (Stack Pointer: SP)

รีจิสเตอร์ตัวชี้สแต็กพอยน์ มีขนาด 8 บิตมีแอดเดรสอยู่ที่ 81H ใช้ในการเก็บค่าตัวชี้สแต็กซึ่งสามารถเปลี่ยนแปลงได้เมื่อซีพียูมีการกระโดดไปทำงานที่โปรแกรมย่อย หรือกระโดดโปรแกรมย่อยกลับมายังโปรแกรมหลักเมื่อมีการรีเซตเกิดขึ้น (รีเซต : การกระโดดที่ส่งผลให้ซีพียูต้องเริ่มต้นการทำงานใหม่ตั้งแต่ต้น) ค่าของรีจิสเตอร์ SP จะเท่ากับ 07H ดังนั้นแอดเดรสแรกที่สำรองไว้ทำหน้าที่เป็นสแต็กจะเท่ากับ 08H

รีจิสเตอร์ชี้ข้อมูล หรือดาต้าพอยน์เตอร์ (Data Pointer: DPTR)

มีขนาด 16 บิต โดยแบ่งเป็นรีจิสเตอร์ชี้ข้อมูลไบต์สูง (DPH) และรีจิสเตอร์ชี้ข้อมูลไบต์ต่ำ (DPL) แต่ละตัวมีขนาด 8 บิต มีแอดเดรสอยู่ที่ 82H สำหรับ DPL และ 83H สำหรับ DPH รีจิสเตอร์ DPTR ใช้ในการเก็บค่าแอดเดรสหน่วยความจำ หรืออุปกรณ์ภายนอกที่ไมโครคอนโทรล เลอร์ต้องการติดต่อด้วย

รีจิสเตอร์พอร์ต (Port Register)

เป็นรีจิสเตอร์ 8 บิตที่ใช้ในการเก็บข้อมูลแต่ละพอร์ตของไมโครคอนโทรลเลอร์ MCS – 51 มี 4 ตัว คือ รีจิสเตอร์พอร์ต0 หรือ P0 มีแอดเดรสอยู่ที่ 80H รีจิสเตอร์พอร์ต 1 หรือ P1 มีแอดเดรสอยู่ที่ 90H รีจิสเตอร์พอร์ต 2 หรือ P2 มีแอดเดรสอยู่ที่ A0H และรีจิสเตอร์ 3 หรือP3 มีแอดเดรสอยู่ที่ B0H รีจิสเตอร์ทุกตัวสามารถเข้าถึงได้ในระดับบิต เมื่อต้องการอ่านหรือเขียนข้อมูลไปยังพอร์ตของไมโครคอนโทรลเลอร์จะต้องกระทำผ่านรีจิสเตอร์นี้ทุกครั้ง

รีจิสเตอร์บัฟเฟอร์ข้อมูลอนุกรม (Serial Data Buffer: SBUF)

เป็นรีจิสเตอร์ขนาด 8 บิต ที่ใช้ในการเก็บข้อมูลที่ส่งออกหรือรับเข้าของวงจรสื่อสารอนุกรมที่มีอยู่ในไมโครคอนโทรลเลอร์ โดยภายในรีจิสเตอร์ SBUF จะแบ่งเป็น 2 ส่วนคือ รีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูล (Transmit Buffer Register) และรีจิสเตอร์บัฟเฟอร์สำหรับรับข้อมูล (Receiver Buffer Register) เมื่อมีการเขียนข้อมูลมายังรีจิสเตอร์ SBUF ข้อมูลนั้นจะถูกส่งต่อไปยังบัฟเฟอร์ สำหรับส่งข้อมูลเพื่อส่งออกจากไมโครคอนโทรลเลอร์ผ่านทางขา TxD หรือขา P3.1 ในกรณีที่มีการอ่านข้อมูลจากรีจิสเตอร์ SBUF ข้อมูลจะถูกส่งออกไปยังรีจิสเตอร์บัฟเฟอร์รับข้อมูลเพื่อส่งไปยังไมโครคอนโทรลเลอร์ต่อไป สำหรับการรับข้อมูลอนุกรมจากภายนอกนั้นจะผ่านมาทางขา RxD หรือ P3.0 ของไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รีจิสเตอร์ไทเมอร์ (Timer Register)

เป็นรีจิสเตอร์ขนาด 16 บิตแบ่งเป็นไบต์สูงและไบต์ต่ำ รีจิสเตอร์ไทเมอร์ใช้ในการเก็บค่าตัวนับหรือเคาน์เตอร์ (Counter) ภายในไมโครคอนโทรลเลอร์ เพื่อใช้ในการสร้างฐานเวลา จับเวลา หรือนับจำนวนพัลส์สัญญาณนาฬิกาภายใน บางที่เรียกรีจิสเตอร์ตัวนี้ว่า **"รีจิสเตอร์ไทเมอร์ / เคาน์เตอร์"** ในไมโครคอนโทรลเลอร์เบอร์ AT89C51 มีรีจิสเตอร์ไทเมอร์/ เคาน์เตอร์ 2 ตัว ได้แก่ T0 และ T1 ในรีจิสเตอร์ยังแบ่งเป็นรีจิสเตอร์ไทเมอร์ไบต์ต่ำ(TL) และรีจิสเตอร์ไทเมอร์ไบต์สูง (TH) โดยรีจิสเตอร์ TL0 มีแอดเดรสอยู่ที่ 8AH รีจิสเตอร์TH0 มีแอดเดรสอยู่ที่ 8BH ในขณะที่ TL1 และ TH1 มีแอดเดรสอยู่ที่ 8CH และ 8DH

รีจิสเตอร์ที่ใช้ในการควบคุม (Control Register)

รีจิสเตอร์ PCONเป็นรีจิสเตอร์ที่เกี่ยวข้องกับการกำหนดอัตราการรับส่งข้อมูลของวงจรสื่อสารอนุกรม และกำหนดการทำงานในหมวดประหยัดพลังงานของไมโครคอนโทรลเลอร์ มีแอดเดรสอยู่ที่ 87H เข้าถึงข้อมูลได้แบบไบต์อย่างเดียวเท่านั้น

รีจิสเตอร์ SCON เป็นรีจิสเตอร์ที่ใช้ในการควบคุมการทำงานวงจรสื่อสารอนุกรมภายในไมโครคอนโทรลเลอร์

รีจิสเตอร์ TCON และT2CON เป็นรีจิสเตอร์ที่ใช้ในการควบคุมการทำงานของไทเมอร์ / เคาน์เตอร์ภายในไมโครคอนโทรลเลอร์ โดย T2 CON ใช้สำหรับไทเมอร์ / เคาน์เตอร์ 2 ของไมโครคอนโทรลเลอร์ AT89C52

รีจิสเตอร์ TMOD และT2MOD เป็นรีจิสเตอร์ที่ใช้ในการกำหนดโหมด หรือในลักษณะในการทำงานของไทเมอร์/เคาน์เตอร์ภายในไมโครคอนโทรลเลอร์ โดยT2MODใช้สำหรับไทเมอร์/เคาน์เตอร์ 2 ของไมโครคอนโทรลเลอร์เบอร์ AT89C52

รีจิสเตอร์ IE และIP เป็นรีจิสเตอร์ที่เกี่ยวข้องกับการตอบสนองการอินเตอร์รัปต์ (Interrupt : การขัดจังหวะการทำงานปกติของซีพียู) โดย IE ใช้ในการกำหนดลักษณะของการตอบสนองการอินเตอร์รัปต์ ในขณะที่ IP ใช้กำหนดลำดับความสำคัญของการตอบสนองการอินเตอร์รัปต์ว่าจะให้ซีพียูตอบสนองการอินเตอร์รัปต์ในลักษณะใดก่อน หรือหลัง

5.5 พอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์ MCS-51 มีวงจรสื่อสารอนุกรมแบบฟูลดูเพล็กซ์ 1 ชุด โดยใช้พอร์ต 3 ได้แก่ ขาP 3.0 ซึ่งเป็นขารับข้อมูลเข้ามา และขาP 3.1 เป็นขาที่ใช้ส่งข้อมูลออกมา โดยการสื่อสารข้อมูลแบบอนุกรมของไมโครคอนโทรลเลอร์จะเป็นแบบอะซิงโครนัส จะใช้ในการติดต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กับพอร์ตอนุกรมของคอมพิวเตอร์ โดยใช้มาตรฐาน RS –232 รีจิสเตอร์ที่เกี่ยวข้องกับพอร์ตอนุกรมใน MCS-51 มีด้วยกัน 2 ตัว คือ

รีจิสเตอร์บัฟเฟอร์ของพอร์ตอนุกรม หรือ SBUF (Serial Data buffer register)

เป็นรีจิสเตอร์ขนาด 8 บิต อยู่ที่แอดเดรส 99H ใช้ในการรับและส่งข้อมูล โดยในการรับจะใช้ขาP 3.0 ส่วนส่งจะใช้ขาP 3.1

รีจิสเตอร์ควบคุมการทำงานพอร์ตอนุกรม หรือSCON (Serial Port Control Register)

เป็นรีจิสเตอร์ขนาด 8 บิต มีแอดเดรสอยู่ที่ 98H ในพื้นที่ของรีจิสเตอร์ SFR สามารถเข้าถึงข้อมูลในระดับบิต มีรายละเอียดการทำงานดังนี้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SM0 - SM1 (Serial Port Mode bit 0-1) : ในการเลือกโหมดการทำงานของพอร์ตอนุกรมภายในไมโครคอนโทรลเลอร์ MCS-51 มีด้วยกัน 4 โหมด ดังแสดงตามตาราง 5.3

SM0	SM1	โหมด	การทำงาน	อัตราการรับ- ส่ง
0	0	0	Shift Register	F_{osc} / 12
0	1	1	8 bit UART	Variable
1	0	2	9 bit UART	F_{osc}/32หรือF_{osc}/64
1	1	3	9 bit UART	Variable

ตาราง 5.3 แสดงการกำหนดโหมดทำงานของพอร์ตอนุกรม ด้วย บิต SM0 และSM1

SM2 : ในการเอ็นเอเบิลการสื่อสารแบบมัลติโปรเซสเซอร์ (multiprocessors) ในการทำงานของโหมด 2 และ 3 ของพอร์ตอนุกรม ในไมโครคอนโทรลเลอร์ MCS-51 ถ้าบิตนี้เป็น **"1"** บิต RI จะไม่แอกทีฟ ถ้าบิตที่9 รับเข้ามาเป็น **"0"** (ข้อมูลบิตที่ 9 เก็บไว้ที่บิต RB 8) ในการทำงานของโหมด1 ถ้าบิตนี้เซต บิต RI จะไม่แอกตีฟ ถ้ายังไม่ได้รับบิตหยุด ส่วนในโหมด 0 บิตนี้ไม่มีการใช้งาน

REN (Enable Serial Reception) : ใช้ในการเอ็นเอเบิลของพอร์ตอนุกรม ทำการเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์ ถ้าต้องการรับข้อมูลต้องเซตบิตนี้ให้เป็น **"1"**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TB8 : ใช้สำหรับเก็บข้อมูลในบิตที่ 9 ที่ต้องการส่งออกไปในการทำงานโหมด 2 และ 3 ของพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51 สามารถเซตและเคลียร์โดยกระบวนการทางซอฟต์แวร์

RB8 : ใช้สำหรับข้อมูลบิตที่ 9 ที่เข้ามาทำงานในโหมด 2 และ 3 ของพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ แต่หากพอร์ตอนุกรมทำงานอยู่ในโหมด 1 และบิต SM2 เป็น **"0"** ข้อมูลของบิต RB 8 คือ ข้อมูลบิตหยุด (Stop Bit) สำหรับในการทำงานโหมด 0 บิตนี้จะไม่ใช้งาน บิต RB 8 นี้สามารถเซตและเคลียร์โดยกระบวนการทางซอฟต์แวร์

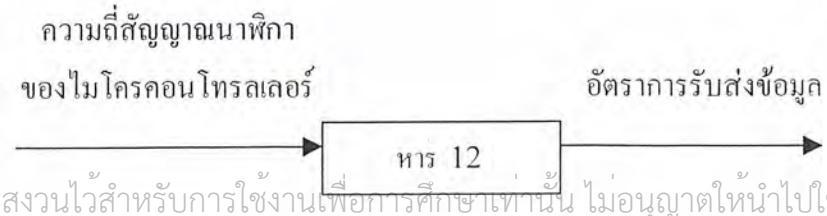
TI (Transmit Interrupt Flag) : ใช้ในการแสดงการเกิดอินเตอร์รัปต์เมื่อมีการส่งข้อมูลออกจากพอร์ตอนุกรมไมโครคอนโทรลเลอร์ MCS-51 สามารถเซตได้ด้วยกระบวนการทางฮาร์ดแวร์เมื่อทำการส่งข้อมูลที่ 8 ไปเรียบร้อยแล้วในการทำงานโหมด 0 ส่วนในโหมดอื่นๆ บิตนี้จะเซตเมื่อมีการเริ่มต้นส่งบิตหยุดออกไป การเคลียร์บิตนี้ต้องใช้กระบวนการทางซอฟต์แวร์เท่านั้น

RI (Receiver Interrupt Flag) : ใช้ในการเกิดอินเตอร์รัปต์เมื่อมีการรับข้อมูลเข้าสู่พอร์ตอนุกรม สามารถเซตได้โดยกระบวนการทางฮาร์ดแวร์เมื่อมีการรับข้อมูลบิตที่ 8 เรียบร้อยแล้วในโหมด 0 ส่วนในการทำงานโหมดอื่น บิตนี้จะถูกเซตเมื่อสามารถรับบิตหยุดของข้อมูลอนุกรมไปได้ครึ่งทางแล้ว ยกเว้นในกรณีที่บิต SM2 มีการเซต บิตนี้จะเซตได้ก็ต่อเมื่อการรับบิตหยุดหรือบิตที่ 9 เกิดขึ้นอย่างสมบูรณ์แล้ว การเคลียร์บิตนี้สามารถกระทำได้โดยผ่านทางซอฟต์แวร์เท่านั้น

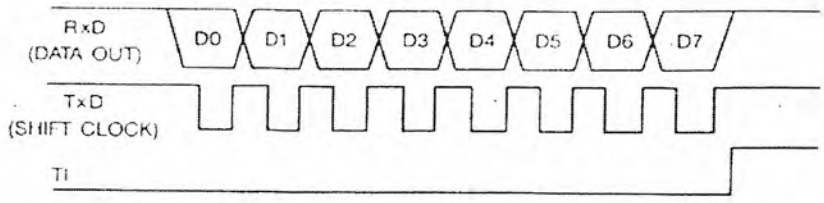
5.6 โหมดการทำงานของพอร์ตอนุกรมใน MCS-51

สามารถเลือกการทำงานได้ถึง 4 โหมด คือ

โหมด 0 เป็นการกำหนดให้พอร์ตอนุกรมทำงานในลักษณะชิฟต์รีจิสเตอร์ โดยที่ข้อมูลอนุกรมจะผ่านเข้าและออกทางขา3.0 (RxD) ส่วนขา3.1(TxD) เป็นตัวกำหนดสัญญาณนาฬิกาที่ใช้ในการเลื่อนข้อมูล (Shift Clock) เพื่อให้อุปกรณ์ภายนอกรับรู้จังหวะการรับส่งข้อมูลที่ขา 3.0 (RxD) ซึ่งจะเป็นข้อมูลขนาด 8 บิต โดยที่ข้อมูลจะเริ่มต้นจากบิตต่ำสุดก่อน(LSB: Least Significant Bit) แล้วจึงเรียงลำดับข้อมูลจนถึงบิตสูงสุด (MSB: Most Significant Bit) ซึ่งเป็นการสิ้นสุดข้อมูลในโหมด 0 อัตราการรับส่งข้อมูลจะมีค่าคงที่อยู่ที่ 1/12 ของความถี่สัญญาณนาฬิกา

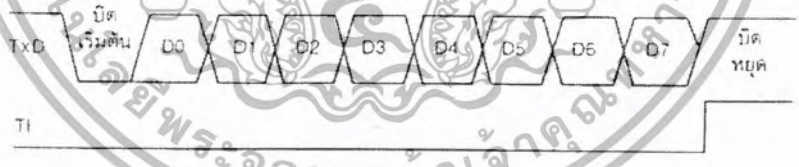


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.15 แสดงสัญญาณการรับ และส่งข้อมูลในโหมด 0

โหมด 1 เป็นการกำหนดให้พอร์ตอนุกรมทำงานในลักษณะ 8 bit UART ซึ่งจะมีข้อมูลขนาด 10 บิต โดยจะประกอบไปด้วยบิตเริ่มต้นของข้อมูล (Start Bit) จะมีค่าเป็น "0" จากนั้นก็จะตามด้วยข้อมูลอีก 8 บิต และบิตสุดท้าย(Stop bit) อีก 1 บิต มีค่าเป็น "1" การรับข้อมูลบิตจะนำมาเก็บไว้ที่บิต RB8 ของรีจิสเตอร์SCON โดยที่ข้อมูลจะส่งออกทางขา3.1 (TxD) และจะรับข้อมูลเข้ามาที่ขา 3.0 (RxD) อัตราการรับส่งข้อมูลจะถูกกำหนดโดยไทเมอร์ของไมโครคอนโทรลเลอร์ การทำงานในโหมดนี้จะได้รับความนิยมสูงสุด เนื่องจากมีกระบวนการที่ไม่ซับซ้อนและสามารถรับส่งข้อมูลกับคอมพิวเตอร์ได้อย่างมีประสิทธิภาพ



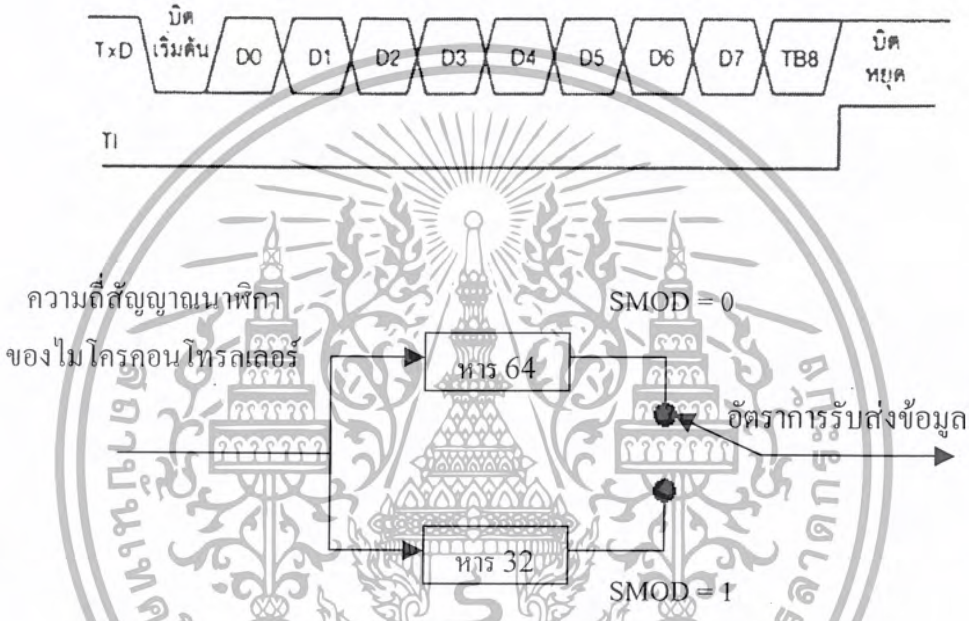
รูปที่ 5.16 แสดงสัญญาณการรับและส่งข้อมูล 8 Bit UART ในโหมด 1

โหมด 2 เป็นการกำหนดให้พอร์ตอนุกรมทำงานในลักษณะ 9 bit UART ซึ่งจะมีข้อมูลขนาด 11 บิต โดยประกอบไปด้วยบิตเริ่มต้น (Star Bit) 1 บิต มีค่าเป็น **"0"** จากนั้นก็ตามด้วยบิตข้อมูลอีก 8 บิต และมีเพิ่มขึ้นมาอีก 1 บิต คือ บิตที่ 9 (Parity Bit) ซึ่งเป็นบิตที่ตรวจสอบความผิดพลาดของข้อมูลที่ใช้ในการรับและส่งนั่นเอง ส่วนอีกบิตคือ บิตสุดท้าย (Stop Bit) มีค่าเป็น **"1"** เสมอ ในการส่งข้อมูลบิตที่ 9 จะนำไปเก็บไว้ในบิต TB8 ในรีจิสเตอร์SCON ส่วนในการรับข้อมูลบิตที่ 9 จะนำไปเก็บในบิต RB8 ของรีจิสเตอร์ SCON อัตราการรับส่งข้อมูลในโหมดนี้จะสามารถเลือกใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้ ที่ 2 ความถี่ คือ 1/32 or 1/64 ของ X-TAL ที่ใช้ โดยสามารถกำหนดได้ที่บิต SMOD ของรีจิสเตอร์ PCON

- ถ้าบิต SMOD เป็น 0 ค่า Baud rate เป็น 1/64 ของความถี่ X-TAL ที่ใช้
- ถ้าบิต SMOD เป็น 1 ค่า Baud rate เป็น 1/32 ของความถี่ X-TAL ที่ใช้



รูปที่ 5.17 แสดงสัญญาณการรับและส่งข้อมูลของ 9 Bit UART ในโหมด 2

โหมด 3 จะมีรูปแบบขนาดการรับและส่งข้อมูลที่เหมือนกับโหมด 2 แต่จะแตกต่างกันตรงที่อัตราการรับและส่งขึ้นอยู่กับไทเมอร์ของไมโครคอนโทรลเลอร์ซึ่งเหมือนโหมดที่ 1

ไทเมอร์ ในการกำหนดอัตราเร็วของการรับส่งข้อมูลที่อยู่ภายในไมโครคอนโทรลเลอร์ จะมี Timer 0 Timer1 และ Timer 2 ในบางเบอร์ของไมโครคอนโทรลเลอร์ โดยจะใช้ รีจิสเตอร์ TMOD (Timer / Counter Mode Control Register) ซึ่งเป็นรีจิสเตอร์ขนาด 8 บิต มีแอดเดรสอยู่ที่ 89 H ที่สามารถแบ่งออกเป็น 4 ส่วน คือ 4 บิตล่างใช้ในการเลือกโหมดการทำงานของ Timer 0 และ 4 บิตบนใช้ในการเลือกโหมดการทำงานของ Timer 1 ดังแสดงได้ดังรูป 5.17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0
ไทเมอร์ 1				ไทเมอร์ 0			

รูปที่ 5.18 แสดงรีจิสเตอร์การเลือกโหมดการทำงานของไทเมอร์ / เคาน์เตอร์ (TMOD)

ส่วนรายละเอียดของ TMOD แสดงตามตารางข้างล่าง

ชื่อบิต	รายละเอียด
GATE	เท่ากับ 0 เลือกใช้ไทเมอร์ หรือเคาน์เตอร์ได้
C/T	ใช้เลือกการทำงานว่าเป็นไทเมอร์หรือเคาน์เตอร์
	1 = เป็นเคาน์เตอร์นับพัลส์ภายนอกที่ขา T0 , T1
	0 = เป็นไทเมอร์ ใช้สัญญาณอินพุทจากสัญญาณนาฬิกาภายในไมโครคอนโทรลเลอร์
M0	บิตเลือกโหมดใช้งานไทเมอร์ 0 หรือ 1
M1	บิตเลือกโหมดใช้งานไทเมอร์ 0 หรือ 1

ตาราง 5.4 แสดงการควบคุมการใช้งานไทเมอร์ 1

ถ้า M0 กับ M1 = **"00"** โหมดไทเมอร์ / เคาน์เตอร์ 13 บิต

= **"01"** โหมดไทเมอร์ / เคาน์เตอร์ 16 บิต

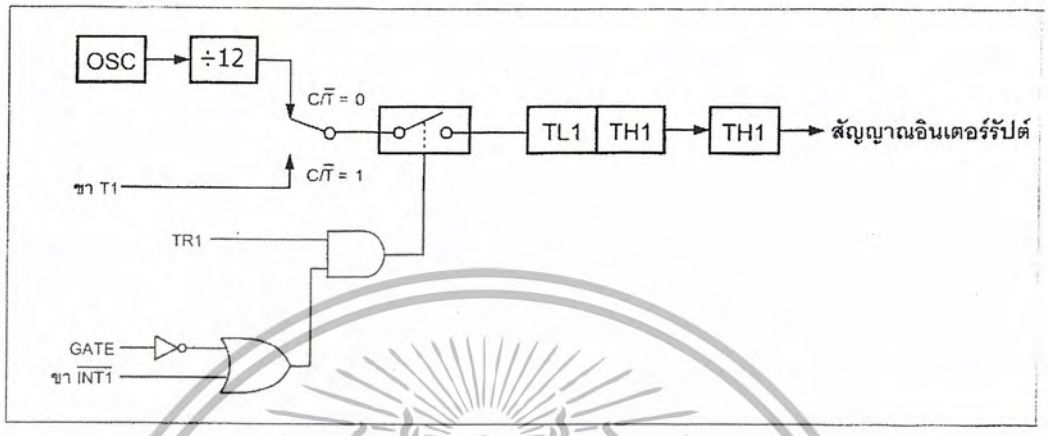
= **"10"** โหมดไทเมอร์ / เคาน์เตอร์ ขนาด 8 บิต แบบตั้งค่าอัตโนมัติ

= **"11"** โหมดไทเมอร์ / เคาน์เตอร์ แบบแยกส่วนโดยแยกออกเป็นไทเมอร์ / เคาน์เตอร์ 8 บิต 2 ตัว รีจิสเตอร์ TL0 ควบคุมการเปิดปิดด้วยบิต TR0 ในรีจิสเตอร์ TCON และรีจิสเตอร์ TH0 เป็นไทเมอร์ / เคาน์เตอร์ 8 บิต อีกตัวหนึ่งควบคุมด้วยบิต TR1 ในรีจิสเตอร์ TCON แต่ถ้าเป็นในกรณีของไทเมอร์ 1 เป็นการสั่งให้ไทเมอร์ / เคาน์เตอร์ 1 หยุดการทำงาน

การเลือกโหมดการทำงานของไทเมอร์ / เคาน์เตอร์ 0 และ 1 สามารถกระทำได้ที่รีจิสเตอร์ TCON และ TMOD โดยที่ TCON ใช้ในการ Enable or Disable Timer / Counter ส่วน TMOD เลือกโหมด และลักษณะของการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

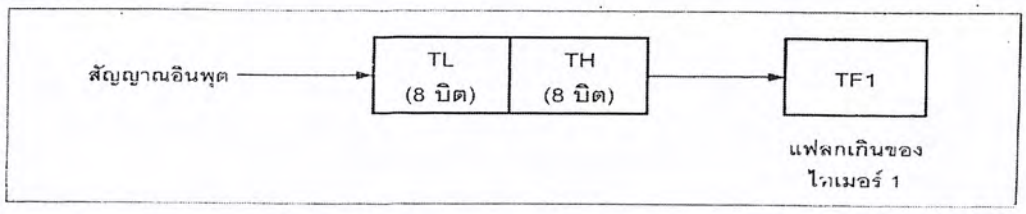
การทำงานในโหมด 0 : ไทเมอร์ / เคาน์เตอร์ 13 บิต



รูปที่ 5.19 การทำงานของไทเมอร์ / เคาน์เตอร์ ในโหมด 0

การทำงานในรูป 5.19 จะใช้ไทเมอร์ / เคาน์เตอร์ 1 โหมดนี้จะเป็นการกำหนดให้ใช้งานรีจิสเตอร์ TL1 เพียง 5 บิต TH1ครบ 8 บิต โดย TL0 จะคอยหาร 32 สัญญาณอินพุทสำหรับการนับจะเลือกจากสัญญาณนาฬิกาภายในหรือภายนอกผ่านทางขา T1 ขึ้นอยู่กับการควบคุมของบิตC/T และ GATE ในรีจิสเตอร์ TMOD บิต TR1 ในรีจิสเตอร์ TCON และสถานะของลอจิกที่ขาอินพุต INT 1 เมื่อ TL1 นับครบ 32 คือจาก 0-31 ก็จะส่งสัญญาณไปยัง TH1 เพื่อทำการเพิ่มค่า ดังนั้นในโหมดนี้ค่าของการนับจะมีขนาด 13 บิตเมื่อทำการนับครบรอบก็จะเซตบิต TF1 ในรีจิสเตอร์TCON ส่วนการทำงานของไทเมอร์ / เคาน์เตอร์ 0 เหมือนกัน เพียงแต่เปลี่ยนรีจิสเตอร์และขาสัญญาณให้เกี่ยวข้องกับไทเมอร์/เคาน์เตอร์ 0

การทำงานในโหมด 1 : ไทเมอร์/เคาน์เตอร์ 16 บิต

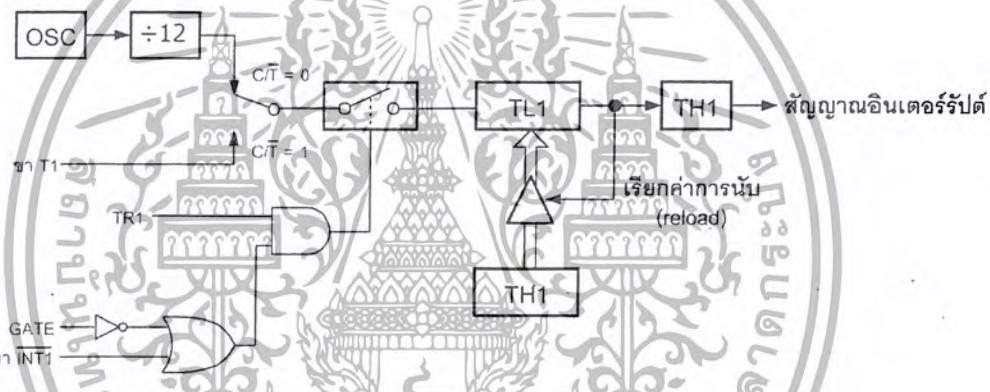


รูปที่ 5.20 การทำงานของไทเมอร์ / เคาน์เตอร์ ในโหมด 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 5.20 ใช้ไทเมอร์ 1 อธิบายการทำงานในโหมดนี้ซึ่งคล้ายกับโหมด 0 แต่จะใช้งาน รีจิสเตอร์TL1 และ TH1 ครบ 8 บิต ดังนั้นในโหมดนี้ค่าของการนับจะมีขนาด 16 บิต คือ 0000H – FFFFH เมื่อทำการนับจนครบรอบ ค่าของการนับเปลี่ยนจาก FFFFH เป็น 0000H ก็จะเซตบิต TF1 ในส่วนรีจิสเตอร์ TCON ส่วนการทำงานของไทเมอร์ 0 เหมือนกันเปลี่ยนแค่รีจิสเตอร์และขาสัญญาณที่เกี่ยวข้องให้เป็นไทเมอร์ 0

การทำงานในโหมด 2 : ไทเมอร์/ เคาน์เตอร์ 8 บิต แบบตั้งค่าอัตโนมัติ



รูปที่ 5.21 การทำงานของไทเมอร์ / เคาน์เตอร์ ในโหมด 2

จากรูปที่ 5.21 จะใช้ไทเมอร์ / เคาน์เตอร์ 1 อธิบายการทำงานในโหมดนี้ โดยแยกรีจิสเตอร์เป็น 2 ตัว ตัวละ 8 บิต โดยรีจิสเตอร์TL1 ทำหน้าที่เป็นตัวนับค่า ส่วนTH1 ใช้ในการเก็บค่าเริ่มต้นของการนับ เมื่อเริ่มต้นการทำงานค่าของรีจิสเตอร์ TH1 จะถูกส่งไปยัง TL1 ทำให้เมื่อเริ่มต้นการทำงานค่าของรีจิสเตอร์TL1 และ TH1 เหมือนกัน เมื่อ TL1 นับถึง FFH และจะเริ่มต้นการนับใหม่ จะเซตบิต TF1 พร้อมๆ กับทำการรับค่าการนับเริ่มต้นจาก TH1 ใหม่อัตโนมัติ หรือเรียกว่า **"รีโหลด"** (Reload) แม้ว่าจะมีการส่งค่าเริ่มต้นไปยัง TL1 แล้วค่าของข้อมูลในรีจิสเตอร์ TH1 ก็ยังเป็นค่าเดิมไม่มีการเปลี่ยนแปลงจนกว่าจะมีการกำหนดค่าใหม่ โดยผ่านกระบวนการทางซอฟต์แวร์ ส่วนการทำงานของไทเมอร์ 0 เหมือนกันเปลี่ยนแค่รีจิสเตอร์และขาสัญญาณที่เกี่ยวข้องให้เป็นไทเมอร์ 0

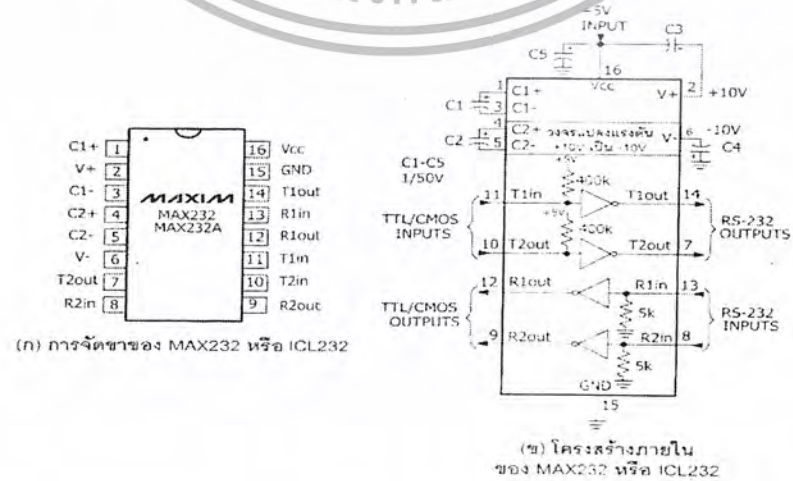
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานในโหมด 3 : ไทเมอร์/เคาน์เตอร์แยกส่วน หรือไทเมอร์ขนาด 8 บิต

เป็นโหมดเดียวที่การทำงานของไทเมอร์ 0 และไทเมอร์ 1 ไม่เหมือนกัน ในส่วนของไทเมอร์ 1 นั้น เมื่อเข้าสู่โหมดนี้จะเป็นการสั่งให้ไทเมอร์ / เคาน์เตอร์หยุดนับ ค่าของการนับก่อนหน้านี้จะถูกเก็บไว้ในรีจิสเตอร์ไทเมอร์ 1 มีลักษณะการทำงานที่เหมือนกับการดิสเอเบิลไทเมอร์ / เคาน์เตอร์ 1 ด้วยการเคลียร์บิต TR1 ในรีจิสเตอร์ TCON ส่วนในการทำงานของไทเมอร์ 0 ในโหมดนี้แสดงดังรูป 5.18 การทำงานในโหมดนี้จะแยกรีจิสเตอร์ ไทเมอร์0 ออกเป็น 2 ตัว ตัวละ 8 บิต คือ รีจิสเตอร์ TL0 และTH0 โดยแยกการทำงานออกจากกันรีจิสเตอร์ TL0 สามารถเลือกการทำงานได้เหมือนกับไทเมอร์ / เคาน์เตอร์ตามปกติ ส่วนรีจิสเตอร์ TH0 สามารถทำงานในโหมดไทเมอร์ได้อย่างเดียว คือ สามารถรับสัญญาณอินพุตจากสัญญาณนาฬิกาภายในเพียงทางเดียวเท่านั้น แต่การแจ้งการนับเกินยังคงเหมือนเดิม หากแต่ TL0 แจ้งผ่านบิต TH0 จะแจ้งผ่านทางบิต TF1 อย่างไรก็ตามสามารถเปลี่ยนแปลงได้ขึ้นอยู่กับผู้ใช้งาน ไม่จำเป็นต้องทำตามค่าเหล่านี้

5.7 การเชื่อมต่อกับพอร์ตอนุกรมของคอมพิวเตอร์

การใช้งานวงจรของพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51 มักนิยมใช้เพื่อการติดต่อแลกเปลี่ยนข้อมูลกับคอมพิวเตอร์ผ่านทางพอร์ตอนุกรม ในมาตรฐาน RS-232 เป็นส่วนใหญ่ แต่เนื่องจากระดับสัญญาณของพอร์ตอนุกรม RS-232 มีระดับตั้งแต่ ± 3 ถึง ± 12V ในขณะที่ระดับสัญญาณของไมโครคอนโทรลเลอร์ MCS-51 อยู่ในระดับ TTL ± 5V จึงไม่สามารถเชื่อมต่อพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51 เข้ากับพอร์ตอนุกรมของคอมพิวเตอร์ได้โดยตรงจึงต้องอาศัยการเชื่อมต่อผ่าน ไอซีพิเศษที่ทำหน้าที่ในการแปลงระดับสัญญาณ



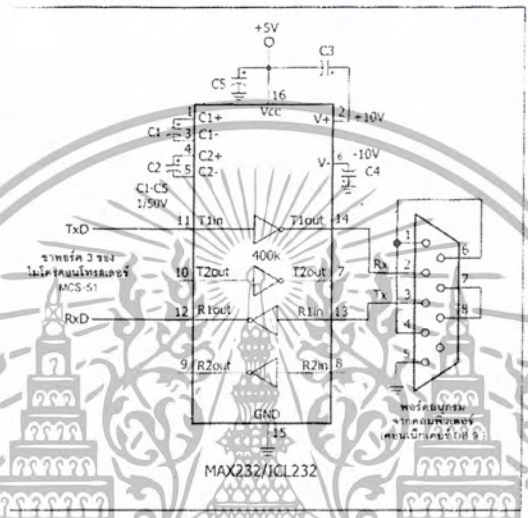
(ก) การจัดขาของ MAX232 หรือ ICL232

(ข) โครงสร้างภายใน ของ MAX232 หรือ ICL232

รูปที่ 5.22 ไอซีแปลงสัญญาณเพื่อเชื่อมต่อกับพอร์ตอนุกรมของคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไอซีที่ทำหน้าที่ในการแปลงระดับสัญญาณนี้ ต้องทำการแปลงข้อมูลภาคส่งของไมโครคอนโทรลเลอร์ MCS-51 จากระดับทีทีแอลไปเป็นระดับของ RS-232 และทำการแปลงข้อมูลรับจากคอมพิวเตอร์จากระดับของ RS-232 เป็นระดับของทีทีแอลเพื่อให้สามารถถ่ายทอดไปยังไมโครคอนโทรลเลอร์ MCS-51 ได้อย่างสมบูรณ์ ไอซี



รูปที่ 5.23 วงจรเชื่อมต่อ MAX 232 เข้ากับพอร์ตอนุกรมของคอมพิวเตอร์ และไมโครคอนโทรลเลอร์ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

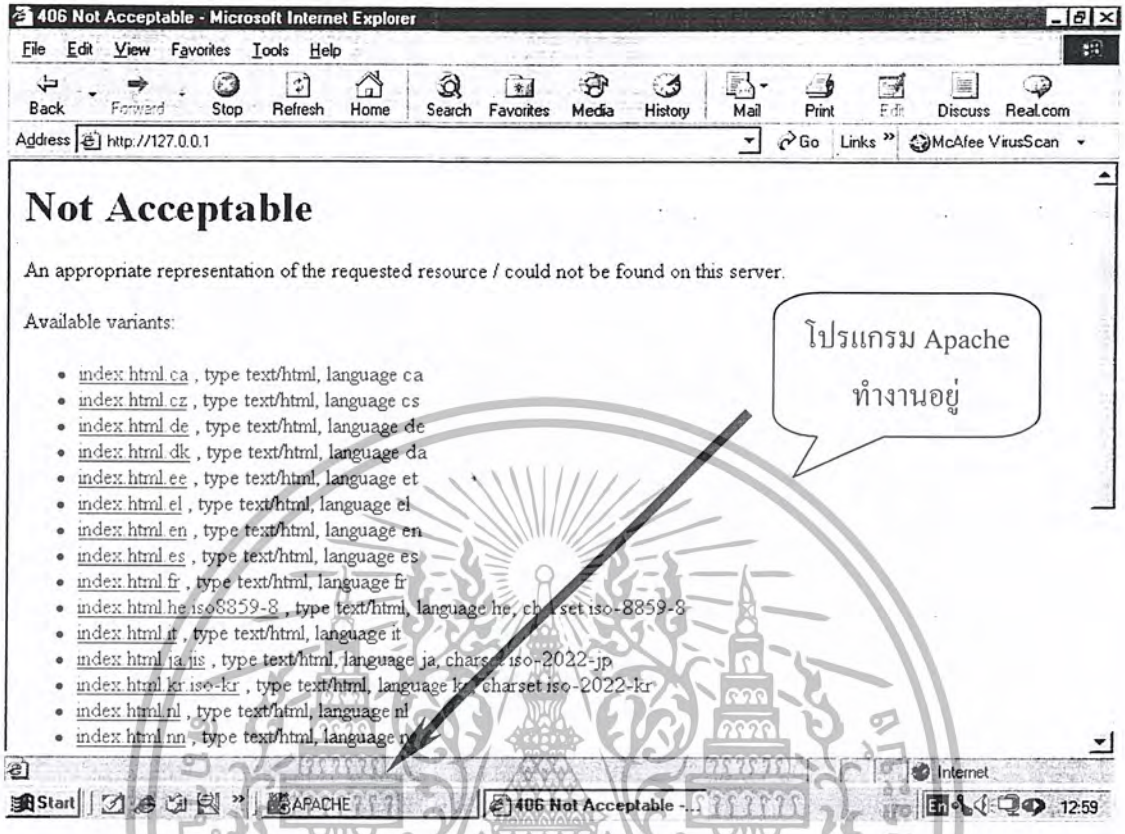
บทที่ 6
การติดตั้ง Apache เพื่อจำลอง WAP Server

การจำลองเครื่องพีซีเป็นเซิร์ฟเวอร์นั้น จะมีโปรแกรมที่สามารถจำลองเป็นเซิร์ฟเวอร์หลายโปรแกรม ได้แก่ Omni HTTPd , PWS (Personal Web Server) , IIS4 (Internet Information Server) , Xitami และ Apache Web Server แต่ในโครงงานนี้จะใช้โปรแกรม Apache Web Server จำลองเป็นเว็บเซิร์ฟเวอร์เนื่องจากประสิทธิภาพในการใช้งานที่มีเสถียรภาพสูง และเป็นที่นิยมมากซึ่งสามารถจะค้นหาดาวน์โหลดได้จาก www.apache.org ซึ่งเป็นเวอร์ชั่น Apache-1.3.29-win32-x86-no-src.exe

สาเหตุที่ต้องจำลองเป็นเว็บเซิร์ฟเวอร์ เนื่องจากต้องใช้ความสามารถของเว็บเซิร์ฟเวอร์ในการประมวลผล Script ที่แทรกอยู่ใน WML ดังนั้นถ้าเราต้องการพัฒนาแอพพลิเคชั่นขึ้นมาไม่ว่าจะแทรกอยู่ใน WAP หรือ WEB ก็ตาม ควรที่จะมีความเข้าใจเกี่ยวกับการทำงานเว็บเซิร์ฟเวอร์ การประมวลผล และการเข้าถึงทรัพยากรของเว็บเซิร์ฟเวอร์

หลังจากที่ติดตั้ง Apache Web Server ลงบนเครื่องPC แล้วจึงทำการเซ็ตให้เครื่องคอมพิวเตอร์ส่วนบุคคลเป็นเว็บเซิร์ฟเวอร์ โดยเข้าไปแก้ไขค่า Configuration ในไดเรกทอรี conf ชื่อไฟล์ httpd.conf โดยจะใช้โปรแกรมประเภท Text Editor เปิดไฟล์ httpd.conf แล้วค้นหา **"Server Name"** เมื่อเจอแล้วให้เอาเครื่องหมาย # ออกก่อนหน้าบรรทัด Server Name (เพราะถ้าไม่เอาเครื่องหมายนี้ออก ข้อความที่ตามมาจะไม่มีผลต่อการทำงาน) หลังจากที่ทำการเซ็ตเสร็จแล้วก็ทำการทดสอบว่า Apache สามารถทำงานได้หรือไม่ โดยการรันโปรแกรม Apache Web Server -> Star Apache in Console ขึ้นมาจะพบคำว่า **"Apache / 1.3.29 (win32) running..."** ในลักษณะของหน้าต่าง DOS ซึ่งยังไม่ต้องปิดหน้าต่างนี้ เพราะการปิดหน้าต่างนี้เท่ากับเป็นการปิดการทำงานของเว็บเซิร์ฟเวอร์ ต่อจากนั้นก็ให้เราก็ให้เปิด Internet Explorer ตรง URL พิมพ์ว่า htttp://127.0.0.1/ ถ้า Apache ทำงาน ได้จริงจะปรากฏภาพดังรูปที่ 6.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.1 เป็นการทดสอบโปรแกรม Apache ว่าสามารถทำงานได้โดย Internet Explorer

6.1 ทำให้ Apache รู้จักกับ WAP

หลังจากที่สามารถจำลองเว็บเซิร์ฟเวอร์ได้แล้ว โปรแกรม Apache Web Server จะรู้จักเพียงเอกสารจำพวก HTML แต่ยังไม่รู้จักไฟล์ในตระกูล WML ที่ใช้เขียน WAP Site เราสามารถทำให้ Apache Web Server รู้จักเอกสารของ WAP ได้ซึ่งมีวิธีอยู่ 2 วิธี คือ

1. แก้ไขข้อมูลในไฟล์ httpd.conf
2. เพิ่มไฟล์ .htaccess ลงไปในไดเรกทอรีที่ใช้ในการเก็บไฟล์ของ WAP

- ถ้าเป็นการจำลองให้เครื่องพีซีเป็นเว็บเซิร์ฟเวอร์จะทำให้มีความสะดวก ในการเข้าไปแก้ไขไฟล์ httpd.conf ได้เลย แต่ในกรณีที่ทำการได้เช่าพื้นที่เว็บไซต์ของผู้อื่นไม่สามารถเข้าไปแก้ไขไฟล์ httpd.conf ได้แม้ว่าผู้ให้บริการจะใช้ Apache เป็นเว็บเซิร์ฟเวอร์ก็ตาม ดังนั้นเมื่อไม่สามารถเข้าไปแก้ไขได้ จึงต้องมาเพิ่มไฟล์ .htaccess ลงไปในเอกสารที่ใช้ในการเก็บเอกสาร WAP แทนซึ่งจะอยู่ในโฟลเดอร์ apache group \ apache\ htdocs ซึ่งสามารถทำได้จาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. แก้ไขข้อมูลในไฟล์ httpd.conf

ใช้โปรแกรม Note Pad เปิดไฟล์ httpd.conf แล้วให้ไปที่บรรทัดที่มีคำว่า Addtype จากนั้นให้เพิ่มข้อความลงไป เพื่อให้เว็บเซิร์ฟเวอร์รู้จักเอกสารต่างๆ ในตระกูล WAP โดยการเพิ่มข้อความดังนี้

```
AddType text/vnd.wap.wml wml
AddType application/vnd.wap.wmlc wmlc
AddType text/vnd.wap.wmlscript wmls
AddType application/vnd.wap.wmlscriptc wmlsc
AddType image/vnd.wap.wbmp  wbmp
```

เสร็จแล้วให้การ Save File httpd.conf และ restart program Apache Web Server หลังจากนั้น Apache ก็จะรู้จักไฟล์ตระกูล WAP

2. การเพิ่มไฟล์ .htaccess ลงไปในไดเรกทอรีที่เก็บเอกสาร WAP

สามารถทำได้โดยการใช้โปรแกรม Text Editor อย่าง Notepad โดยการพิมพ์ข้อความดังนี้

```
DirectoryIndex Index.wml
AddType text/vnd.wap.wml wml
AddType application/vnd.wap.wmlc wmlc
AddType text/vnd.wap.wmlscript wmls
AddType application/vnd.wap.wmlscriptc wmlsc
AddType image/vnd.wap.wbmp  wbmp
```

โค๊ดวิธีเหมือนกับวิธีแรกต่างกันตรงมีโค๊ด DirectoryIndex index.wml ซึ่งไฟล์นี้จะเป็นตัวกำหนดให้ไฟล์ index.wml เป็นไฟล์โฮมเพจของไดเรกทอรีนี้ เช่น ถ้ามีโดเมนเป็น mydomain.com ก็จะหมายถึง http://www.mydomain.com/index.wml ทันที

เนื่องจากในการพัฒนา WAP Application ได้นำภาษาทางฝั่งสคริปต์ซึ่งจัดเป็นภาษาทางฝั่งเซิร์ฟเวอร์เข้ามาช่วยเพิ่มประสิทธิภาพและความสามารถให้กับภาษา HTML ในการสร้างเว็บแอพพลิเคชั่นขึ้นมาซึ่งภาษาที่นำมาใช้ก็คือ PHP (Profressional Home Page)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2 การติดตั้ง PHP ให้ทำงานร่วมกับ Apache Web Server

โปรแกรม PHP สามารถ Download ได้ที่ www.PHP.net ซึ่ง Version ที่ใช้ คือ PHP 4.3.4 - win32.zip หลังจากที่ได้ Download เสร็จแล้วให้ปฏิบัติตามขั้นตอนดังนี้

1. ให้แตกไฟล์ไปยังไดเรกทอรีที่ได้กำหนดไว้
2. ก๊อปปี้ไฟล์ php.ini-dis ไปใส่ไว้ใน c:\ windows\ จากนั้นจึงเรียกใช้งานโปรแกรม Notepad เปิดไฟล์ขึ้นมา แล้วทำการแก้ไขข้อมูลในไฟล์ 3 จุด ซึ่งอยู่ในส่วนของ Paths and Directories ดังนี้

   Doc_root = Folder ที่เก็บเอกสารHTMLใน Apache Web Server ;

   User_dir = Folder ที่เก็บเอกสารHTMLใน Apache Web Server ;

   Extension_dir = Folder ที่ได้ติดตั้งโปรแกรม PHP ;
3. จากนั้นทำการเปิดโปรแกรม Apache ขึ้นมาเพื่อให้ Apacheรู้จักไฟล์ตระกูลของ PHP ในขั้นตอน ต่อมาให้แก้ไขไฟล์ httpd.conf ในโฟลเดอร์ของ apache\conf ซึ่งสามารถทำได้ดังนี้

   3.1 ให้หาคำว่า ScriptAlias ซึ่งจะเจอข้อความดังนี้

   ScriptAlias /cgi-bin/ **" Folder ที่ได้ติดตั้ง apache/ cgi-bin/"**

   แล้วเพิ่มอีก 1 บรรทัด

   ScriptAlias /PHP/ **" Folder ที่ติดตั้งโปรแกรม PHP ไว้"**

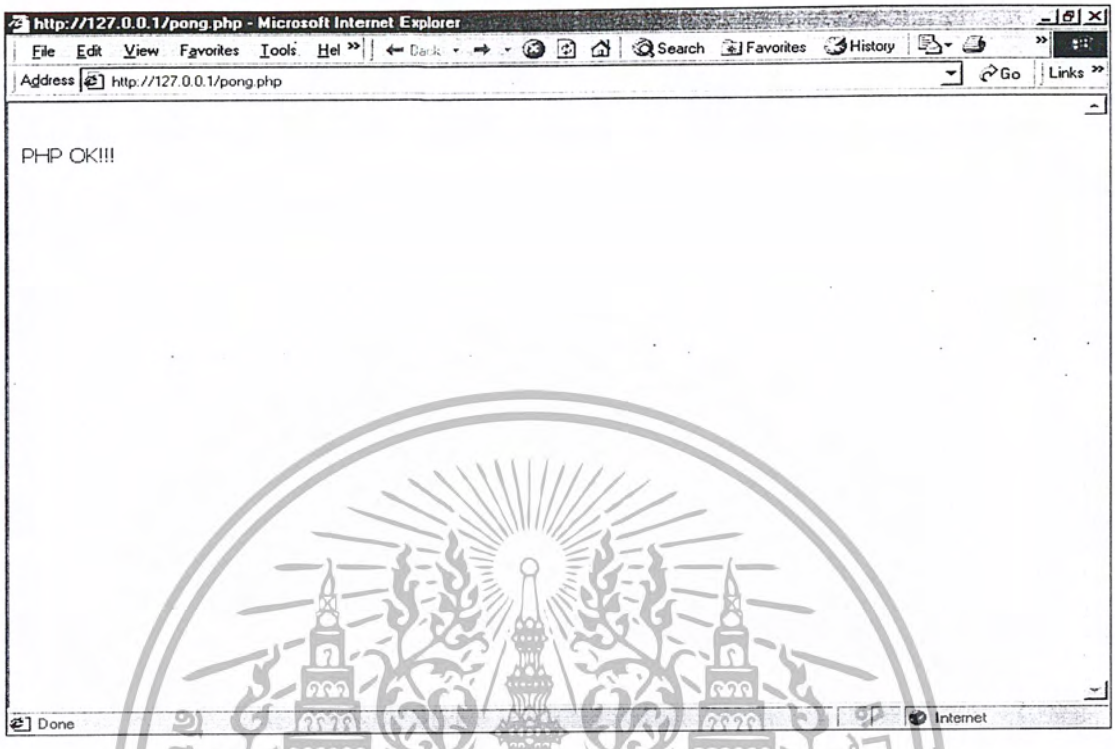
เมื่อทำตามขั้นตอนดังกล่าวเสร็จแล้วก็ทำการทดสอบ PHP ว่าใช้งานได้หรือเปล่า โดยต้องเขียนสคริปต์ของ PHP ขึ้นมาเพื่อทดสอบด้วยใช้โปรแกรม Notepad

ตัวอย่าง เป็นการแสดง code php ที่เขียนขึ้นมา

```
<?
        Echo "PHP OK!!!"
?>
```

แล้วตั้งชื่อไฟล์ที่มีนามสกุล phpนำไปเก็บไว้ใน Folder ของ Apache\ htdocs\ จากนั้นใช้โปรแกรมเบราเซอร์ โดยระบุ URL เป็น http://127.0.0.1/ไฟล์.PHP ในส่วนของโปรแกรมของเบราเซอร์จะปรากฏดังรูปที่ 6.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.2 แสดงผลของสคริปต์ PHP ที่ทำงานร่วมกับ Apache

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7
การออกแบบ และขั้นตอนการทำงาน

บทนี้จะกล่าวถึงโครงงานที่ทำขึ้นมาทั้งหมด ได้แก่ การออกแบบ ขั้นตอนการติดตั้งอุปกรณ์ การทำงานของโปรแกรม และการเซตค่าต่างๆ รวมถึงวัสดุ และอุปกรณ์ที่นำมาใช้ในโครงงานนี้

7.1 ส่วนประกอบของโครงงาน

1. Bluetooth USB Plug จำนวน 3 ชุด
2. คอมพิวเตอร์ PC สำหรับเซิร์ฟเวอร์ 1 เครื่อง และสำหรับไคลเอนต์ 2 เครื่อง
3. โทรศัพท์มือถือ หรือ PDA ที่สามารถใช้บริการ WAP ได้จำนวน 1 เครื่อง
4. อุปกรณ์รับคำสั่งการควบคุมจากไคลเอนต์จำนวน 2 ชุด
5. Software ที่เขียนขึ้น สำหรับเซิร์ฟเวอร์ และไคลเอนต์

7.2 การติดตั้ง และเตรียมโครงงาน

1. อัพโหลดไฟล์ WAP ที่เขียนขึ้นมาสู่พื้นที่ Hosting ในเครื่องคอมพิวเตอร์ที่จำลองเป็น WAP Server (โดยวิธีการคัดลอกไฟล์ของโครงงานเข้าไปสู่โฟล์เดอร์ที่ได้จัดเตรียมไว้ในคอมพิวเตอร์ส่วนบุคคล)
2. ติดตั้งโปรแกรมที่เขียนขึ้นมา สำหรับโปรแกรมเซิร์ฟเวอร์ที่เครื่องเซิร์ฟเวอร์ และโปรแกรมไคลเอนต์ไว้ที่เครื่องไคลเอนต์ทุกเครื่อง
3. ติดตั้ง Driver Bluetooth ที่คอมพิวเตอร์ทุกเครื่อง (โดยใช้แผ่น CD-ROM ที่ติดมากับอุปกรณ์ Bluetooth นั้น)
4. ตั้งค่า Bluetooth และ TCP/IP เพื่อกำหนด IP Address ของแต่ละเครื่อง ซึ่งในการทดลองนี้เลือกใช้
    - สำหรับ Server ใช้ IP Address 192.168.0.5
    - สำหรับ Client 1 ใช้ IP Address 192.168.0.1
    - สำหรับ Client 2 ใช้ IP Address 192.168.0.2

** **หมายเหตุ** IP Address เปลี่ยนได้แต่ต้องอยู่ในWorkgroup เดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

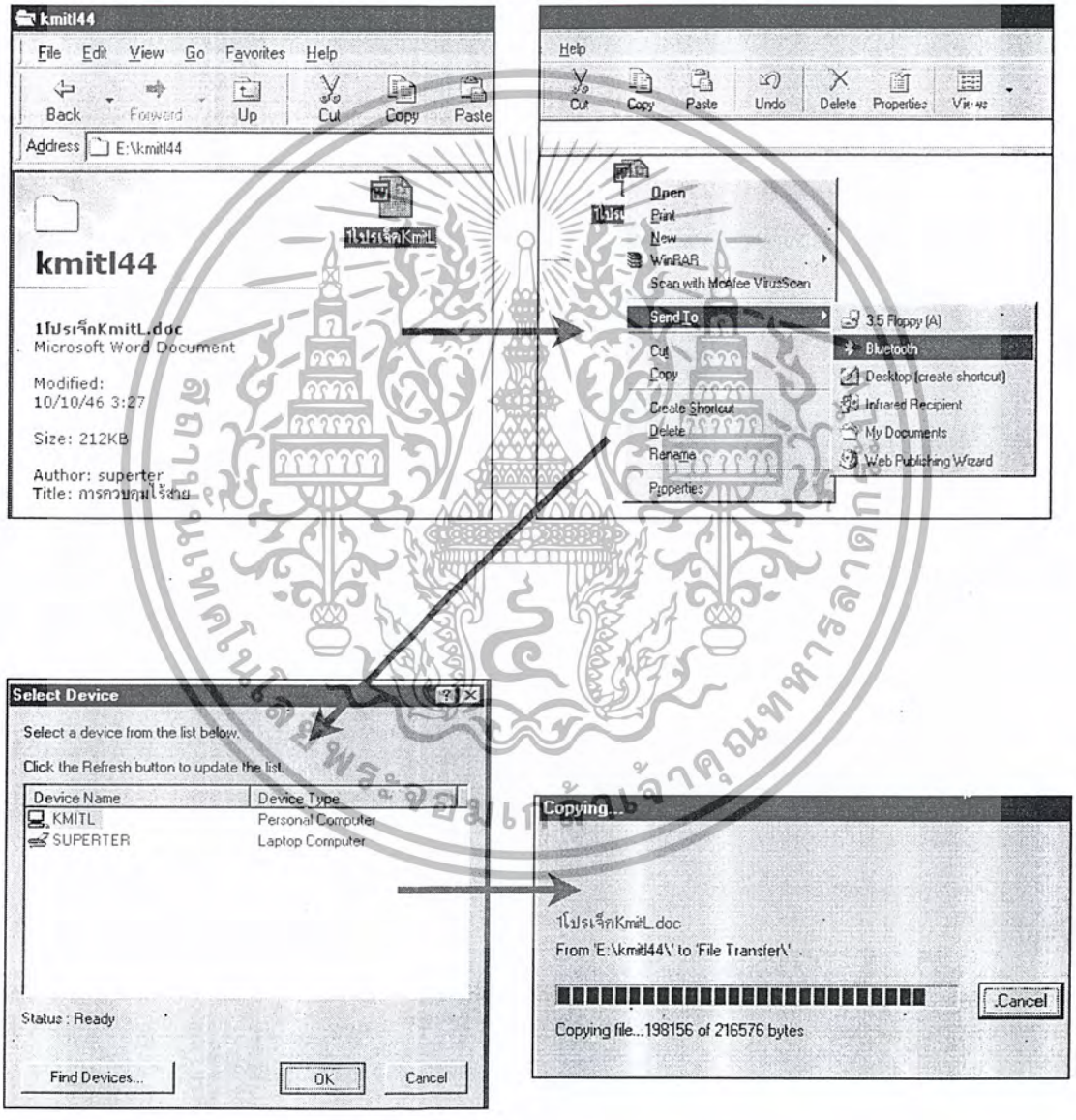
5. ติดต่อไปยังผู้ให้บริการเครือข่ายโทรศัพท์มือถือนั้นๆ จากนั้นผู้ให้บริการระบบเครือข่ายโทรศัพท์ทำการส่งค่าในการตั้งค่า เพื่อเข้าสู่ WAP มาให้โดยอัตโนมัติ

รูปที่ 7.1 โฟลว์ชาร์ตแสดงการติดตั้ง และเตรียมโครงงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำรวจความพร้อม

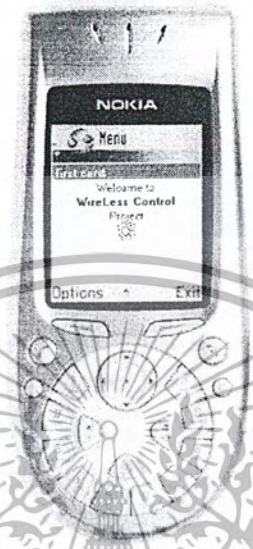
1. เซิร์ฟเวอร์ และ ไคลเอนต์ต้องสามารถติดต่อกันได้ เช่น สามารถส่งไฟล์หากันได้ เป็นต้น โดยใช้โปรแกรมที่มากับอุปกรณ์ Bluetooth เพื่อใช้ในการทดลองการรับ- ส่งไฟล์ข้อมูล ดังแสดงในรูปที่ 7.2



รูปที่ 7.2 แสดงการรับ-ส่งไฟล์ข้อมูลระหว่างกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. WAP Server ที่จำลองไว้ต้องสามารถเข้าถึงได้จริงจากการเชื่อมต่ออินเตอร์เน็ต



รูปที่ 7.3 แสดงผลที่ได้จากการจำลองของ WAP Server

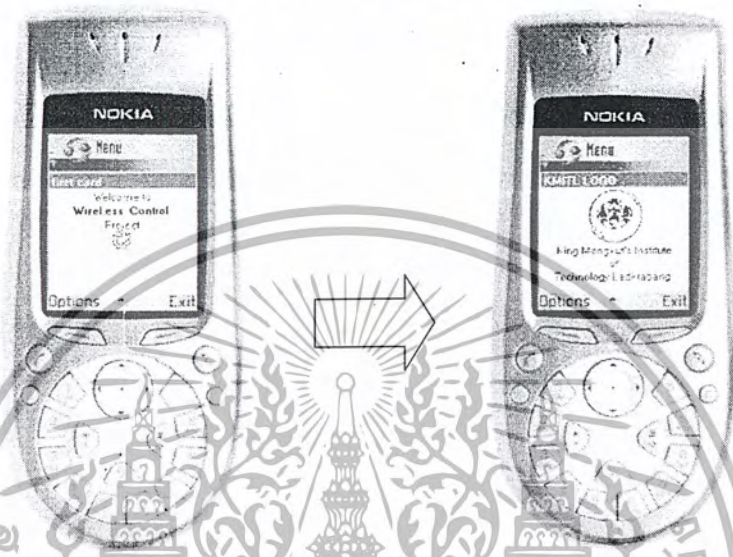
3. โทรศัพท์มือถือต้องสามารถเข้า WAP Site ที่เตรียมไว้ใน WAP Server ได้

รูปที่ 7.4 แสดงผลที่ได้จากการใช้โทรศัพท์เข้ามาที่ WAP Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.3 ขั้นตอนการทำงานของ Wireless Control

1. ให้ Access WAP จากโทรศัพท์มือถือเข้ามาที่เครื่อง WAP Server ที่เตรียมไว้ โดยใช้ IP Address ที่ได้จากการต่อ Internet ในครั้งนั้น ถ้าติดต่อเข้ามาได้แสดงได้ดังรูปที่ 7.5



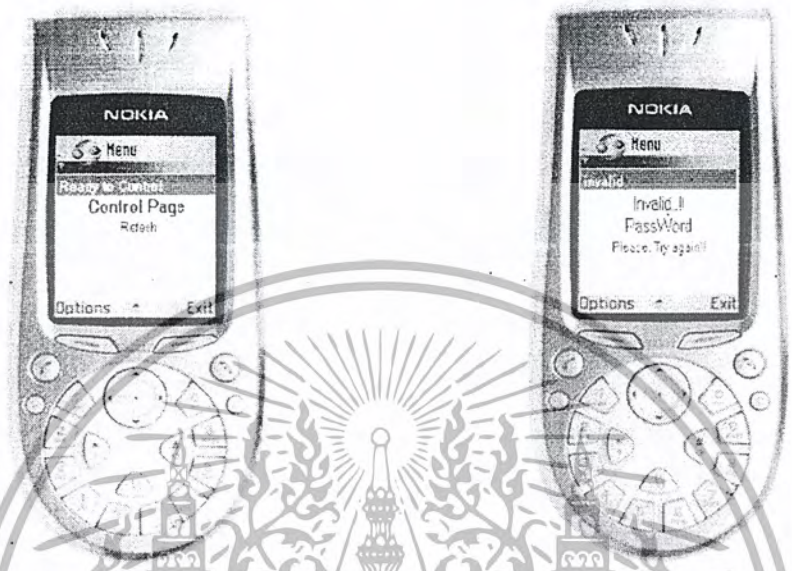
รูปที่ 7.5 แสดงการ Access WAP จากโทรศัพท์มือถือเข้ามาที่เครื่อง WAP Server

2. เมื่อเข้ามาสู่ WAP Project แล้วให้เลือก Login และ กรอก Password เพื่อเข้าสู่ Control Page

รูปที่ 7.6 แสดงหน้าตาการ Login และการป้อนรหัสผ่านบนโทรศัพท์มือถือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 7.6 เมื่อกดที่คำว่า "**Login**" จะแสดงหน้าตาให้กรอกรหัสผ่าน

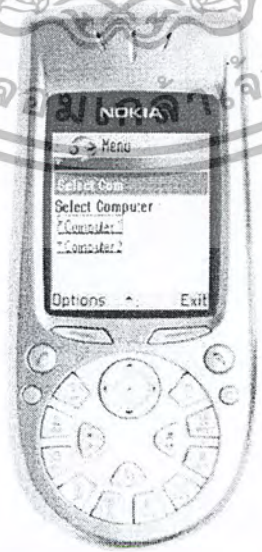


(ก) (ข)

รูปที่ 7.7 (ก) เมื่อกรอก Password ถูกต้อง จะแสดงหน้าตาเข้าสู่ Control Page

รูปที่ 7.7 (ข) เมื่อกรอก Password ผิด จะแสดงหน้าตาร้องเตือน เพื่อให้กรอกใหม่

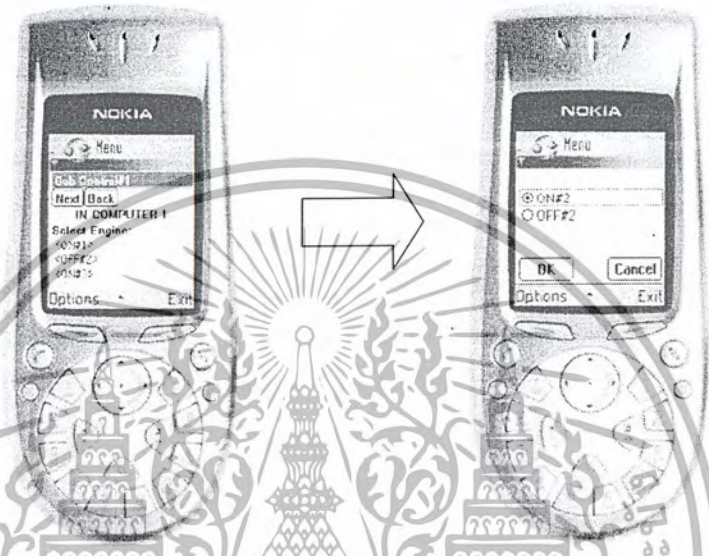
3. จากนั้นให้เลือกเครื่องคอมพิวเตอร์ที่ต้องการควบคุม (Client Computer) ซึ่งได้ทำการเตรียมไว้2 เครื่อง



รูปที่ 7.8 แสดงการเลือกติดต่อกับเครื่องคอมพิวเตอร์ที่ต้องการควบคุม

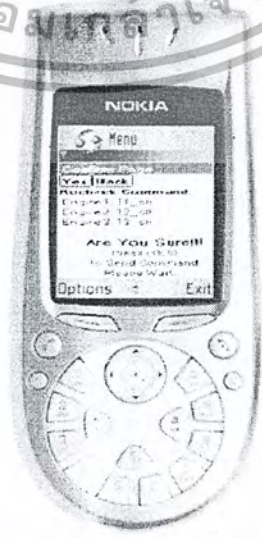
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ต่อจากนั้นให้เลือก Engine ที่ต่อไว้กับ Client Computer ตัวนั้น เพื่อทำการ ON- OFF Engine (ในที่นี้ได้เตรียม Engine ไว้ 3 ชุดต่อ 1 Client Computer) โดยที่ Engine ทั้ง 3 นี้ทำงานเป็นอิสระจากกัน ซึ่งสามารถเลือก ON – OFF ได้ผ่านหน้า Page นี้ แล้วจึงกด Next



รูปที่ 7.9 แสดงการเลือก Engine ในเครื่องคอมพิวเตอร์ที่ต้องการควบคุม และยังสามารถเลือกให้ Engine ตัวใด ๆ ON – OFF ได้

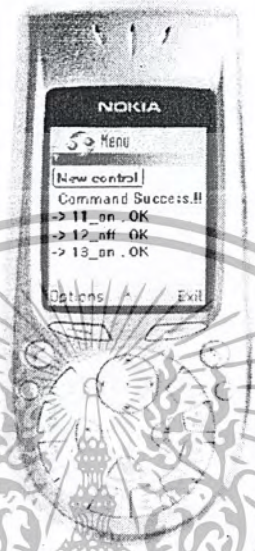
5. WAP Page จะให้ทำการตรวจสอบอีกครั้งเพื่อยืนยันคำสั่ง แล้วกด YES เพื่อส่งคำสั่งไปยัง WAP Server ให้ประมวลผลต่อไป



รูปที่ 7.10 แสดงการยืนยัน Engine ที่ได้เลือกไว้อีกครั้งหนึ่ง

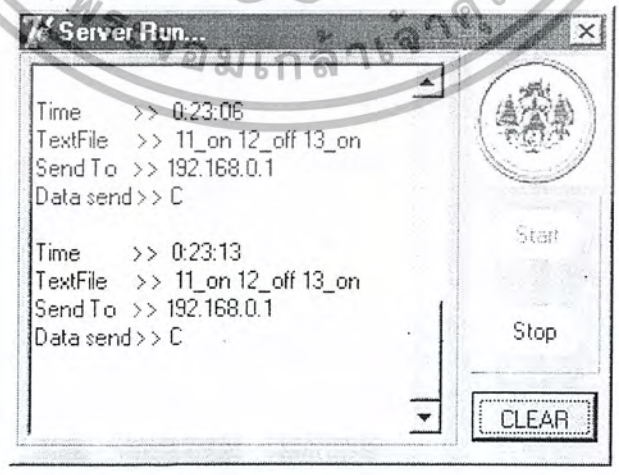
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. เมื่อ WAP Server ได้รับคำสั่งแล้วจึงประมวลผล แล้วส่ง Result Page กลับมายัง WAP Browser จากนั้น WAP Server จะบันทึกผลลัพธ์ที่ได้จากการประมวลผล เพื่อรอการทำงานต่อไป



รูปที่ 7.11 แสดง Result Page ซึ่งยืนยันว่า WAP ได้ปฏิบัติตามคำสั่งนี้เรียบร้อยแล้ว

7. สำหรับโปรแกรมเซิร์ฟเวอร์ที่เขียนขึ้นนั้น หลังจากที่สั่ง Start โปรแกรมแล้ว จะทำการอ่านไฟล์ที่ได้จากการประมวลผลของ WAP Server ทุกๆ 7 วินาที



รูป 7.12 แสดงการทำงานของโปรแกรมเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

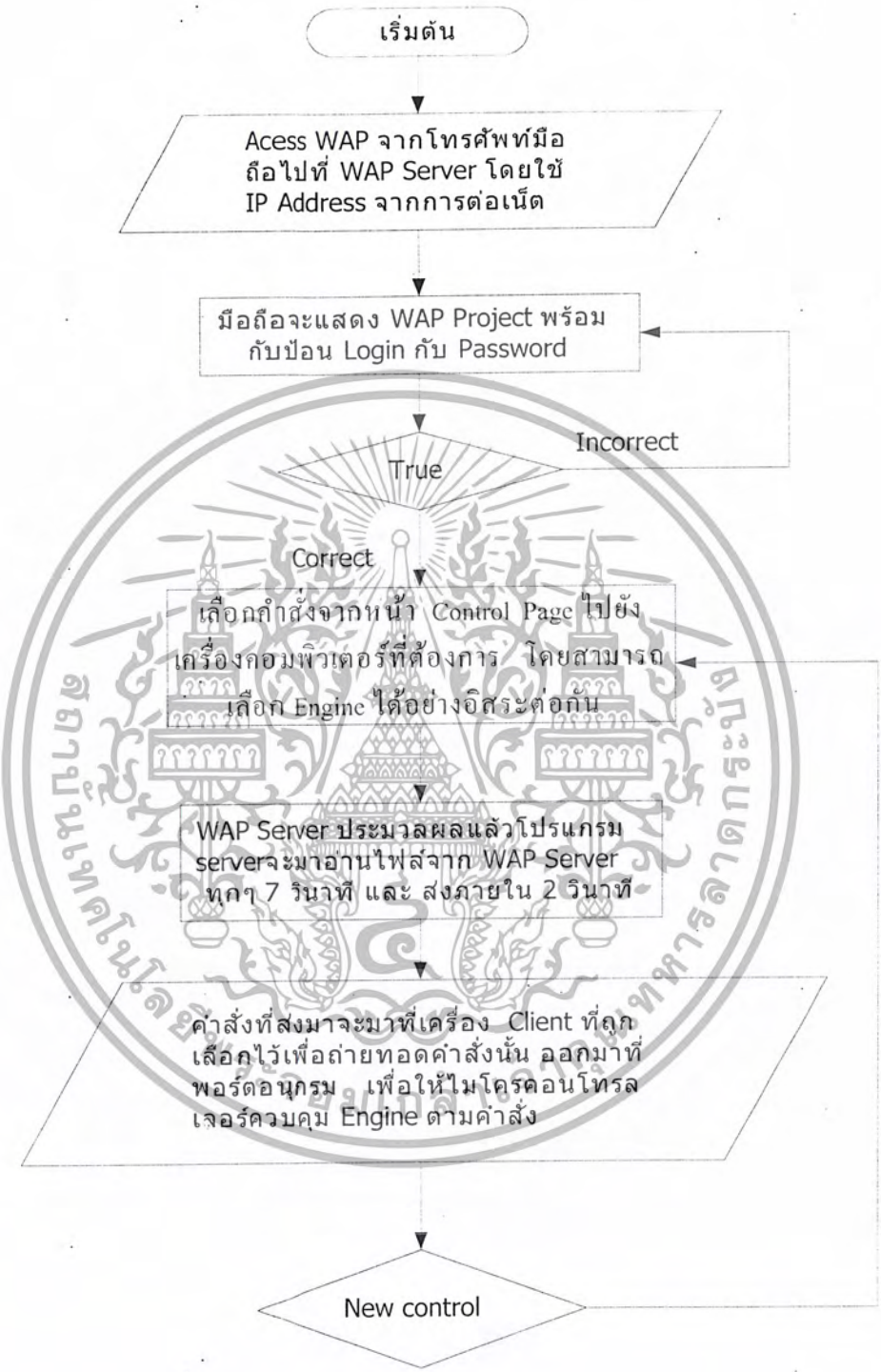
8. ต่อจากนั้น 2 วินาที หลังจากที่โปรแกรม Server อ่านค่าได้ และทำการเข้ารหัสแล้วส่งค่าไปยัง Client Computer ที่เลือกไว้ (จาก Control Page ใน WAP Browser)
9. เมื่อ Client Computer ตัวนั้นๆ ได้รับค่าแล้วจะทำการถอดรหัสทันที เพื่อแปลงเป็นรหัสแอสกี้ดังเดิม แล้วส่งต่อไปยังพอร์ตอนุกรมของคอมพิวเตอร์ เพื่อส่งค่าควบคุมไปให้ไมโครคอนโทรลเลอร์ MCS-51

รูปที่ 7.13 โปรแกรมไคลเอนต์อ่านข้อมูลที่ส่งมา และนำมาถอดรหัส

10. ทันทีที่ไมโครคอนโทรลเลอร์ MCS-51 ได้รับค่าก็จะประมวลผล และควบคุมการ ON - OFF ของ Engine ทั้ง 3 ตัวตามคำสั่ง (ซึ่งในภาพนี้เป็นค่าที่ไมโครคอนโทรลเลอร์รับมาจากโปรแกรมไคลเอนต์ ที่ส่งผ่านทางพอร์ตอนุกรม ก็คือ ค่า **C = ON OFF ON** นั่นเอง)

รูปที่ 7.14 แสดงการถอดรหัสข้อมูลที่ส่งมาจากโปรแกรมของไคลเอนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.15 โฟลว์ชาร์ตแสดงการทำงานของ Wireless Control

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเหตุ

- ส่วนของหน้า WAP Page สามารถวนกลับมาเพื่อเลือก Client Computer ตัวใดก็ได้ โดยไม่ต้องกรอกPassword ใหม่
- โปรแกรมเซิร์ฟเวอร์ ทำงานซ้ำไปเรื่อยๆ ทุกๆ 7 วินาที และเก็บข้อมูลชั่วคราวไว้ที่หน้าเซิร์ฟเวอร์ Display ซึ่งสามารถสั่ง Clear Display ได้ตลอดเวลา
- โปรแกรมไคลเอนต์ จะได้รับคำสั่งจากโปรแกรมเซิร์ฟเวอร์ ทุกๆ 7 วินาที แต่จะล้าหลังกว่าประมาณ 2 วินาที และเก็บข้อมูลชั่วคราวไว้ที่หน้า ไคลเอนต์ Display ซึ่งสามารถสั่ง Clear Display ได้ตลอดเวลา ส่วนข้อมูลถาวรของคำสั่งที่ได้รับถูกบันทึกไว้ในรูปแบบของไฟล์เอกสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 8
ผลการทดลอง และสรุป

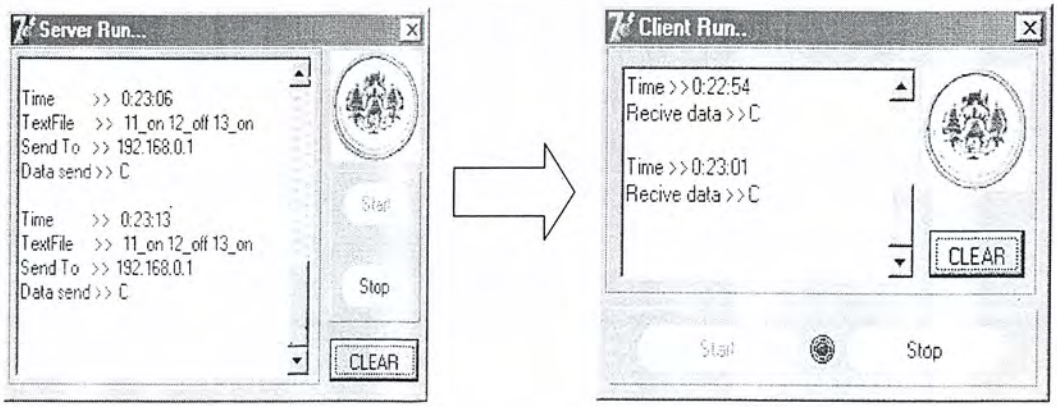
8.1 การทดลองที่ 1 แสดงการติดต่อระหว่างเซิร์ฟเวอร์ กับ ไคลเอนต์

ระยะทาง (เมตร)	Server to Client (วินาที)	ผลการติดต่อ
2	0.5	ติดต่อได้
4	0.5	ติดต่อได้
6	0.5	ติดต่อได้
8	0.5	ติดต่อได้
10	1	ติดต่อได้เป็นบางครั้ง (80%)
12	1	ติดต่อได้เป็นบางครั้ง (30%)
14	-	ไม่สามารถติดต่อได้

ตารางที่ 8.1 แสดงระยะทาง และเวลาที่เซิร์ฟเวอร์ติดต่อ กับ ไคลเอนต์ ผ่าน Bluetooth

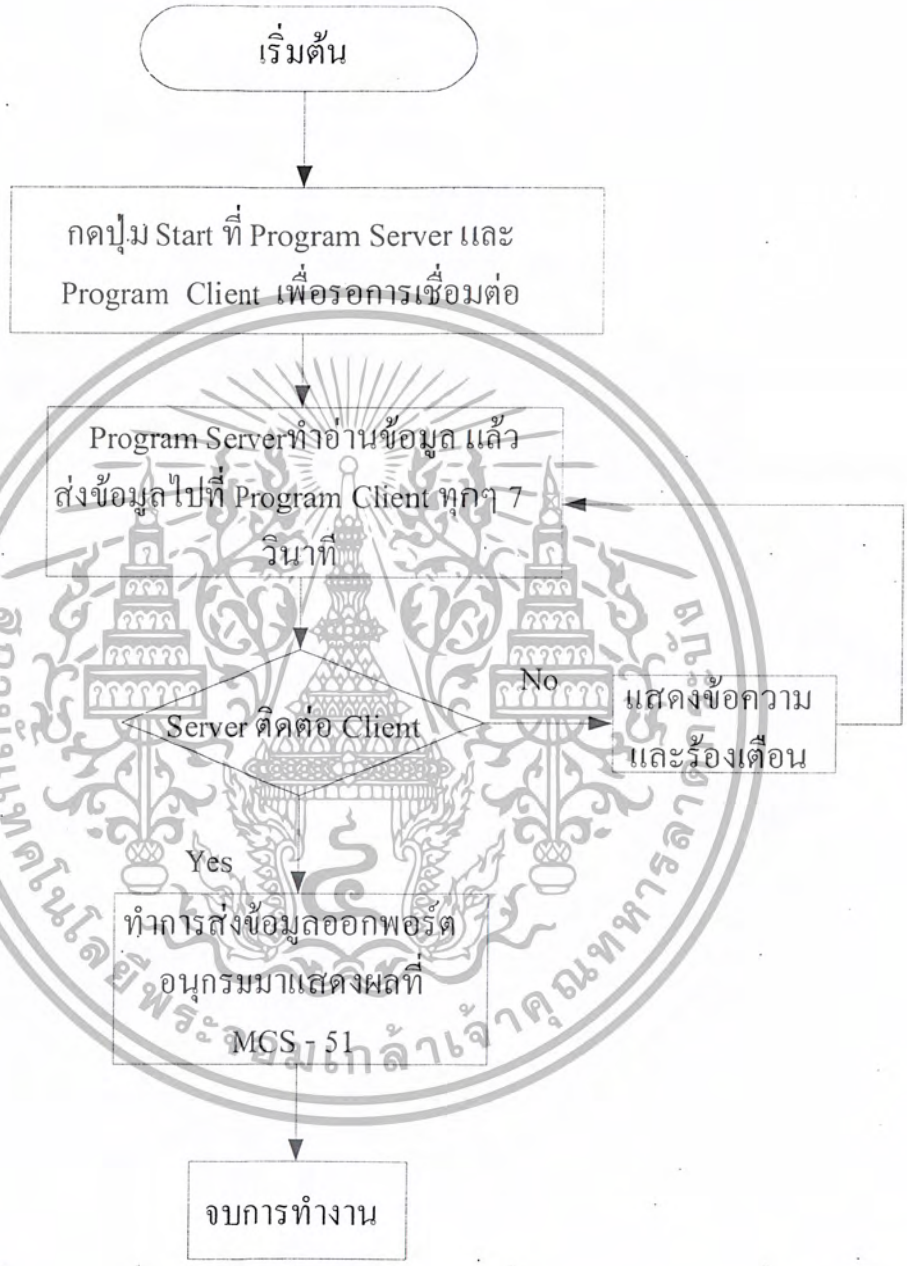
สรุปผลการทดลอง

จากผลการทดลองแสดงให้เห็น ได้ชัดเจนว่าความสามารถของ Bluetooth ด้านการครอบคลุมพื้นที่ถูกจำกัดไว้ประมาณ 10 เมตร ทั้งนี้ขึ้นอยู่กับสภาวะแวดล้อมต่าง ๆ เช่น สภาวะแวดล้อม ประสิทธิภาพ และองค์ประกอบของเครื่องคอมพิวเตอร์ เป็นต้น



รูปที่ 8.1 แสดงการส่งข้อมูลจากโปรแกรมของเซิร์ฟเวอร์ ไปยัง โปรแกรมของไคลเอนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8.2 โฟลว์ชาร์ตแสดงการทำงานของโปรแกรมเซิร์ฟเวอร์ กับ โปรแกรมไคลเอนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

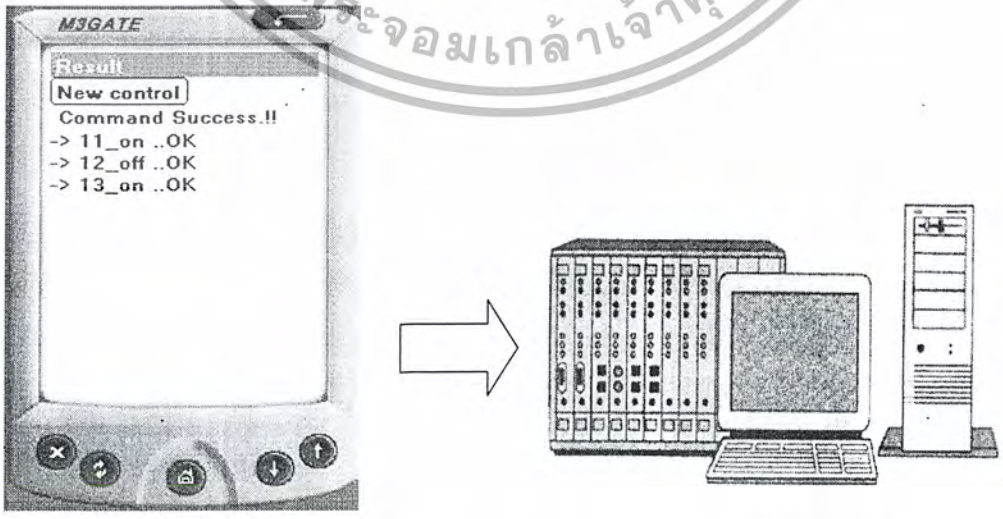
8.2 การทดลองที่ 2 แสดงการส่งข้อมูลจาก WAP Browser ไปยัง WAP Server

ระยะทาง (เมตร)	ระยะเวลา (วินาที)	ผลการติดต่อ
1	10	ได้รับข้อมูลปกติ
10	8	ได้รับข้อมูลปกติ
30	16	ได้รับข้อมูลปกติ
100	9	ได้รับข้อมูลปกติ
500	15	ได้รับข้อมูลปกติ
1,500	10	ได้รับข้อมูลปกติ
เกินกว่า 5,000	13	ได้รับข้อมูลปกติ

ตารางที่ 8.2 แสดงระยะทาง และเวลาการส่งข้อมูลจาก WAP Browser ไปยัง WAP Server

สรุปผลการทดลอง

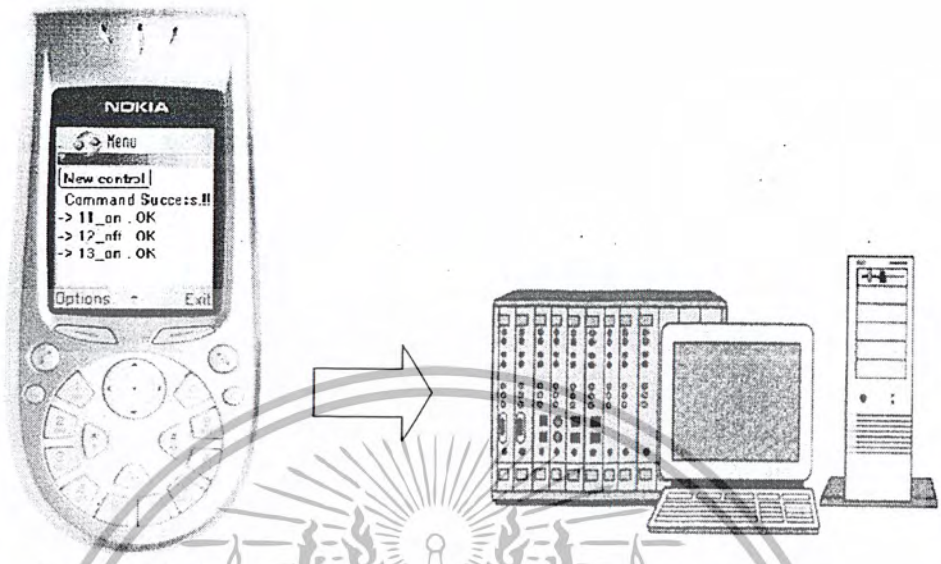
จากผลการทดลองพบว่า ไม่มีระยะทางจำกัดในการส่งข้อมูลจากWAP Browserไปยัง WAP Server ตราบใดที่ผู้ให้บริการโทรศัพท์มือถือยังสามารถบริการได้ แต่ปัญหาที่พบ คือ เวลาที่ไม่แน่นอนในการรับ-ส่งข้อมูล เป็นปัญหาที่แก้ไขได้ยาก และนอกเหนือการควบคุม เนื่องจากเกิดจากระบบInternet และ จำนวนผู้ใช้ปริมาณมหาศาล



WAP Browser (Emulator) WAP Server

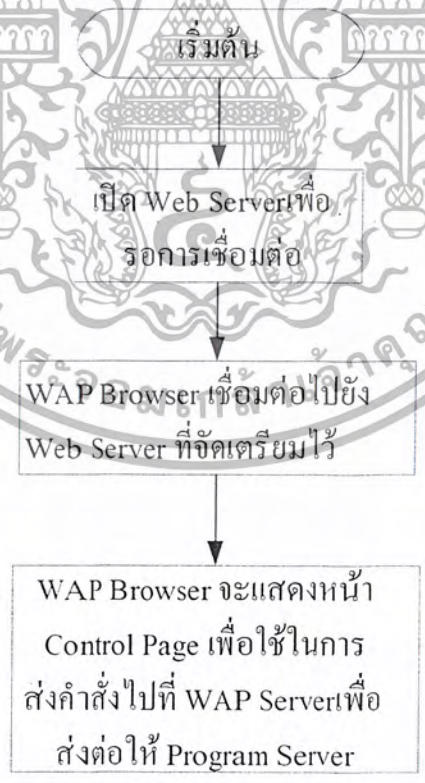
รูปที่ 8.3 แสดงการส่งคำสั่งจาก WAP Browser กับ WAP Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



โทรศัพท์มือถือ WAP Server

รูปที่ 8.4 แสดงการส่งคำสั่งจากโทรศัพท์มือถือ กับ WAP Server



รูปที่ 8.5 โฟลว์ชาร์ตแสดงการทำงานของ WAP Browser กับ WAP Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.3 การทดลองที่ 3 แสดงการส่งข้อมูลจาก WAP Browser ไปยังโปรแกรมไคลเอนต์

ระยะทาง (เมตร)	ระยะเวลา (วินาที)	ผลการทำงาน
1	25	ปฏิบัติตามคำสั่งถูกต้อง
10	13	ปฏิบัติตามคำสั่งถูกต้อง
30	19	ปฏิบัติตามคำสั่งถูกต้อง
100	24	ปฏิบัติตามคำสั่งถูกต้อง
500	17	ปฏิบัติตามคำสั่งถูกต้อง
1,500	17	ปฏิบัติตามคำสั่งถูกต้อง
เกินกว่า 5,000	22	ปฏิบัติตามคำสั่งถูกต้อง

ตารางที่ 8.3 แสดงระยะทาง และการส่งข้อมูลจาก WAP Browser ไปยังโปรแกรมไคลเอนต์

สรุปผลการทดลอง

สอดคล้องกันการทดลองที่ 2 เพราะสามารถส่งคำสั่งจาก WAP Browser ไปยังไคลเอนต์ได้ โดยไม่เป็นอิสระจากระยะทาง อีกทั้งยังสามารถทำงานได้ถูกต้อง 100% แต่ปัญหาเรื่องความล้าช้า ในการส่งข้อมูลของWAP ที่มีในการทดลองที่ 2 ยังคงมีอยู่ และยังมีปัญหาด้านความไม่แน่นอนของอุปกรณ์ Bluetooth รวมถึงปัญหาเรื่องเวลาของโปรแกรมเซิร์ฟเวอร์ที่ทำงานแบบวน ทุกๆ 7 วินาที และส่งข้อมูลหลังจากอ่านแล้ว 2 วินาที สรุปได้เป็นสมการดังนี้

$$T(\text{WAP Browser to Client}) = T1(\text{WAP Delay}) + T2(\text{Bluetooth}) + T3(\text{Program Scan}) + T4(\text{Sender})$$

โดยที่ : T (WAP Browser to Client) = เวลาล้าช้าของ Client ในการตอบสนองคำสั่งของ WAP

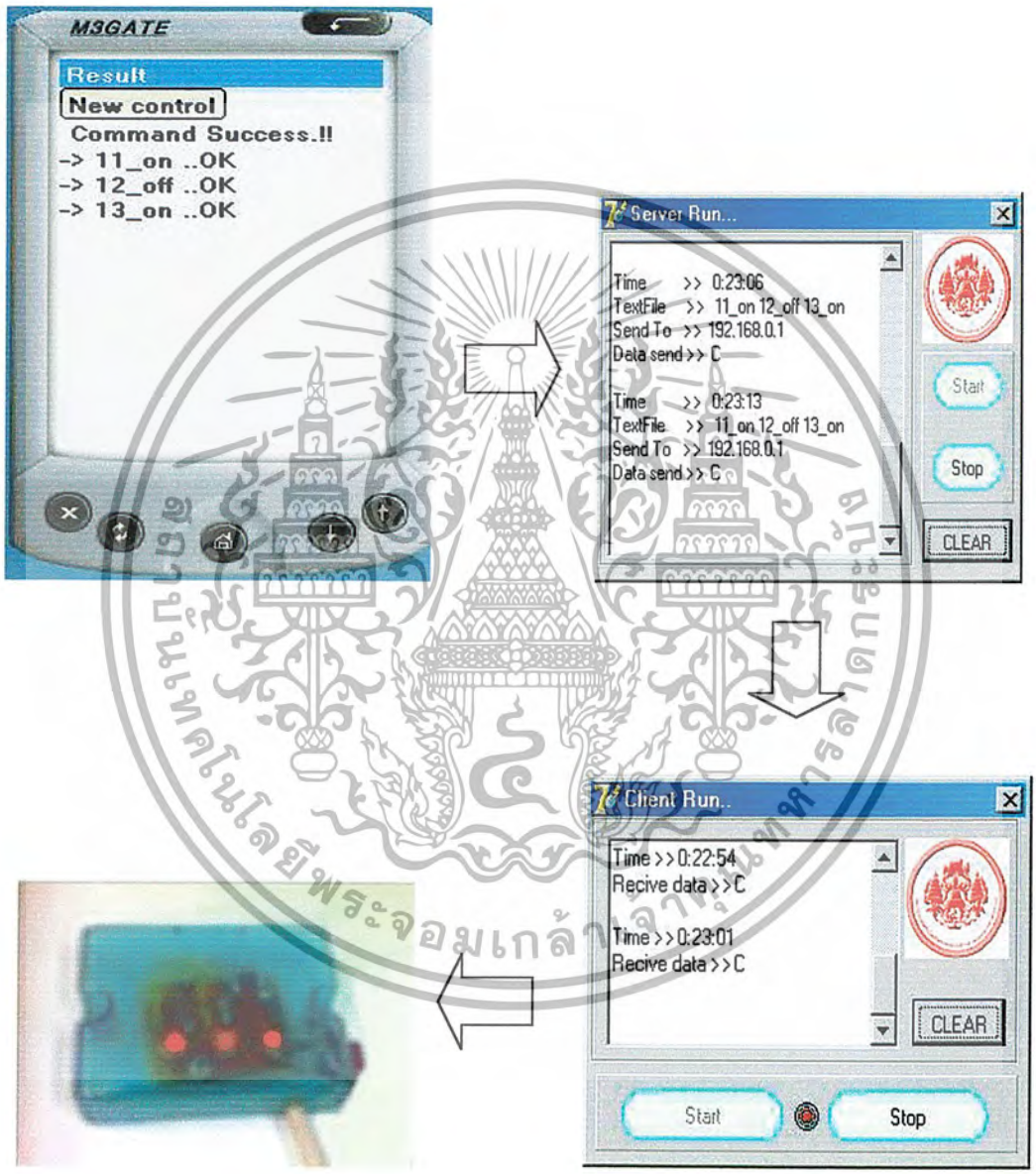
T1 (WAP Delay) = เวลาล้าช้าของ WAP Browser ในการส่งข้อมูลมายัง WAP Server

T2 (Bluetooth) = เวลาล่าช้าในการส่งข้อมูลจากโปรแกรมของเซิร์ฟเวอร์มายังโปรแกรมของไคลเอนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

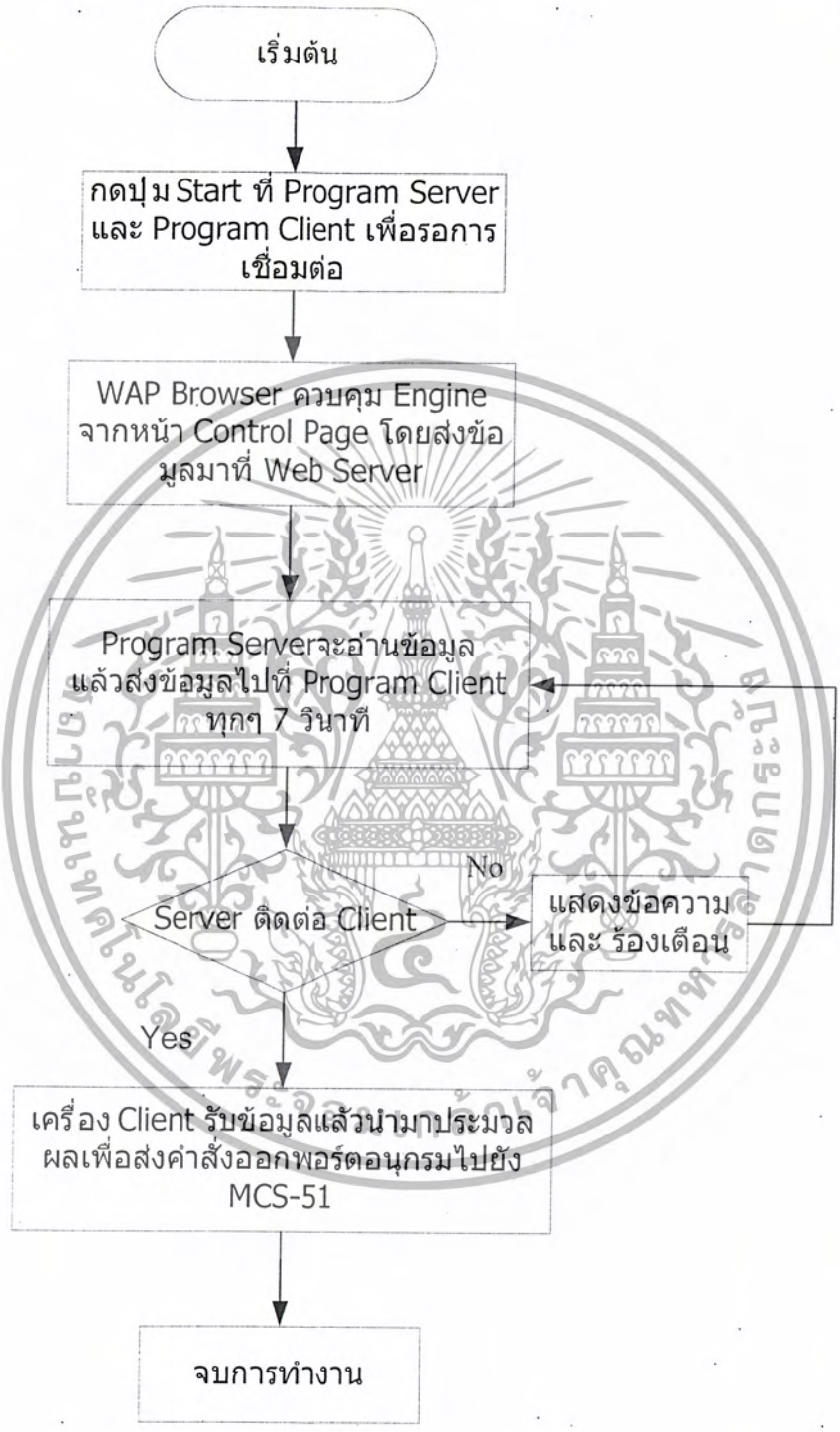
T3 (Program Scan) = เวลาของโปรแกรมเซิร์ฟเวอร์ที่ทำงานแบบวน ทุก ๆ 7 วินาที

T4 (Sender) = เวลาที่โปรแกรมของเซิร์ฟเวอร์ ส่งข้อมูลออกไปยังโปรแกรมของไคลเอนต์หลังจากอ่านแล้ว 2 วินาที



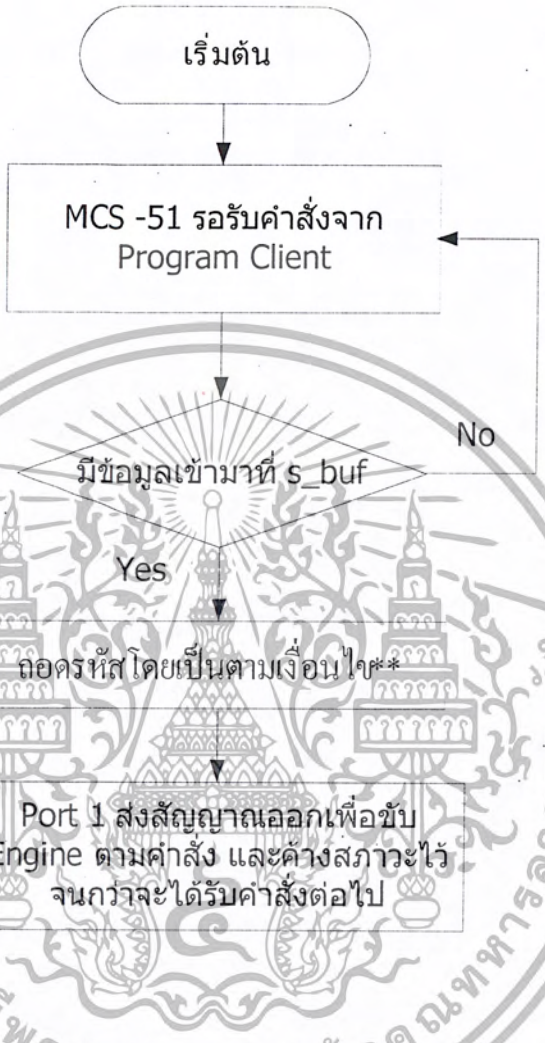
รูปที่ 8.6 แสดงการส่งคำสั่งจาก WAP Browser กับ โปรแกรมไคลเอนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8.7 โฟลว์ชาร์ตแสดงการส่งข้อมูลจาก WAP Browser ไปยังโปรแกรมไคลเอนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8.8 โฟลว์ชาร์ตแสดงการทำงานของ MCS – 51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

** **หมายเหตุ** การเข้ารหัสระหว่างโปรแกรมเซิร์ฟเวอร์ และ โปรแกรมไคลเอนต์ เพื่อให้ MCS –51 ประมวลผล

Engine 1	Engine 2	Engine 3	เข้ารหัสเป็น
ON	ON	ON	A
ON	ON	OFF	B
ON	OFF	ON	C
ON	OFF	OFF	D
OFF	ON	ON	E
OFF	ON	OFF	F
OFF	OFF	ON	G
OFF	OFF	OFF	H

ตารางที่ 8.4 แสดงการถอดรหัสของ MCS-51 ที่รับข้อมูลจากพอร์ตอนุกรมของเครื่องไคลเอนต์มาประมวลผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.4 ปัญหาที่เกิดขึ้น และการแก้ไข

- ความล้าช้าเนื่องจากการส่งค่าในระบบ Internet ของ WAP เริ่มจากผู้ให้บริการโทรศัพท์มือถือ ส่งค่าต่อให้กับ WAP Gateway ไปยังระบบ Internet และส่งไปยังผู้ให้บริการของเครื่องเซิร์ฟเวอร์ ปลายทางแล้วจึงเดินทางมาสู่ WAP Server ที่จัดเตรียมไว้ ใช้เวลาประมาณ 3 – 35 วินาที ขึ้นอยู่กับปัจจัยแวดล้อมหลายประการ เช่น ความพร้อมของผู้ให้บริการโทรศัพท์มือถือ ความหนาแน่นของผู้ใช้งานในเครือข่ายนั้น ณ. เวลานั้นๆ ความสามารถในการประมวลผลของ WAP Server ความสามารถในการประมวลผลของ WAP Browser ในโทรศัพท์มือถือ เป็นต้น
- ความปลอดภัยในการรักษาข้อมูล เนื่องจากส่วนหนึ่งของโครงงานนี้ทำงานอยู่บนระบบอินเตอร์เน็ต ดังนั้นจึงอาจมีปัญหาด้านความปลอดภัยที่อาจถูกบุกรุกจากแฮ็กเกอร์ และผลของไวรัสคอมพิวเตอร์ที่มีอยู่ชุกชุมในระบบอินเตอร์เน็ต
- ความไม่แน่นอนของอุปกรณ์ Bluetooth แบ่งเป็น 2 กรณี

1. **ข้อจำกัดด้านระยะทาง** เนื่องจาก Bluetooth เป็นการสื่อสารโดยใช้คลื่นวิทยุ จึงมีปัจจัยหลายอย่างที่มีผลต่อระยะทางและประสิทธิภาพการส่งข้อมูล เช่น การรบกวนกันเองของแหล่งกำเนิดคลื่นในบริเวณใกล้เคียง หรือได้รับ Noise จากกระบวนการอื่น สภาวะแวดล้อมที่มีอิทธิพลต่อคลื่นวิทยุ ได้แก่ สิ่งกีดขวางในระดับสายตา วัสดุที่เป็นโลหะต่างๆ ต้นไม้ เป็นต้น
2. **การส่งข้อมูลในกลุ่ม PICO NET** ปัญหานี้มีสาเหตุหลักมาจาก Driver ที่ติดมากับอุปกรณ์ Bluetooth ไม่สมบูรณ์เพียงพอ เนื่องจากโครงงานนี้ใน Protocal TCP/IP ซึ่งเป็นProfile ชนิดหนึ่งของ Bluetooth Profile (จากการสำรวจจริงในตลาดเกี่ยวกับ Driver ที่ติดมากับอุปกรณ์ Bluetooth พบว่ามักมีปัญหากับผู้ใช้อยู่เสมอและยังส่งผลกระทบในเชิงลบกับระบบปฏิบัติการ WINDOWS ด้วย) ปัญหาคือ ในกรณีที่เซิร์ฟเวอร์ขาดการติดต่อกับไคลเอนต์ บางตัวไปนานๆ เมื่อต้องการกลับมาติดต่ออีกครั้ง ไคลเอนต์ไม่สามารถตอบสนองได้ทันที ต้องได้รับการกระตุ้น 1 ครั้งก่อนจึงจะตอบสนองตามปกติ ช่วงเวลาที่เซิร์ฟเวอร์ ขาดการติดต่อกับไคลเอนต์ นั้นไม่สามารถระบุได้ชัดเจน จากการทดสอบหลายครั้งกับคอมพิวเตอร์หลายๆเครื่องได้ผลไม่เท่ากันและไม่ใกล้เคียงกัน ประมาณช่วงตั้งแต่ 5 – 50 นาทีขึ้นไป ทั้งนี้อาจขึ้นอยู่กับความเสถียรภาพของระบบปฏิบัติการก็เป็นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.5 แนวทางในการพัฒนาต่อไป

1. ปรับปรุงส่วนของการติดต่อสื่อสารกันระหว่างไคลเอนต์ กับ เซิร์ฟเวอร์ให้ดีขึ้น โดยเฉพาะในเรื่องการเชื่อมต่อ
2. พัฒนาในส่วนของโปรแกรมให้สามารถรองรับอุปกรณ์สื่อสารไร้สายอื่นๆ ได้
3. พัฒนาให้อุปกรณ์ Bluetooth สามารถติดต่อกับ Microcontroller ได้โดยตรง
4. การรักษาความปลอดภัยของไฟล์สำหรับที่ใช้ในการควบคุม
5. ภาษาที่ใช้ในการพัฒนาโครงงานนี้ มีความหลากหลายยากต่อการพัฒนาควรรวมให้เป็นภาษาในตระกูลเดียวกันเพื่อง่ายต่อการแก้ไข และพัฒนาต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมเดลไฟล์ (Delphi)

ในส่วนของโปรแกรมเดลไฟล์จะประกอบไปด้วย 2 ส่วน คือ

โปรแกรมของเซิร์ฟเวอร์

```
unit userver;
interface
uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,Dialogs, Menus,
    ExtCtrls, StdCtrls. DsFancyButton, SckComp. CPort, CPortCtl, jpeg, Buttons;
const
    filename = 'c:\myweb\html\wap\control.txt';
type
    TForm1 = class (TForm)
    Panel1  : TPanel;
    Memo1 : TMemo;
    Timer1  : TTimer;
    Timer2  : TTimer;
    Timer3  : TTimer;
    Panel2   : TPanel;
    btn1 : TDsFancyButton;
    btn2 :  TDsFancyButton;
    BitBtn1 : TBitBtn;
    Image1  : TImage;

      procedure btn1Click(Sender: TObject);
      procedure btn2Click(Sender: TObject);
      procedure FormCreate(Sender: TObject);
      procedure Timer1Timer(Sender: TObject);
      procedure Timer2Timer(Sender: TObject);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        procedure Timer3Timer(Sender: TObject);
        procedure BitBtn1Click(Sender: TObject);
  private
    { Private declarations }
  public
    hostname, data, send51 : string[20];  fi : textfile ;
    client :Tclientsocket;
      procedure onclientread(Sender : TObject; Socket : TCustomWinSocket);
      procedure readfiles();
end;
var
    Form1 : TForm1;
implementation
    {$R *.dfm}

procedure TForm1.btn1Click(Sender: TObject);
begin
    self.Caption := 'Server Run...';
    btn1.Enabled := false;
    btn2.Enabled := true;
    Timer1.Enabled := true;
end;

procedure TForm1.btn2Click(Sender: TObject);
begin
    self.Caption := 'Stop..';
    client.Close ;
    btn1.Enabled := true;
    btn2.Enabled := false;
    self.Caption := 'Server Stop...';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Timer1.Enabled := false;
end;
procedure TForm1.onclientread (Sender : TObject; Socket : TCustomWinSocket);
begin


end;

procedure TForm1.FormCreate(Sender: TObject);
begin
    client := Tclientsocket.Create(self);
    client.Port  := 4000;
    client.OnRead := onclientread;
end;

procedure  TForm1.readfiles;
var  s:shortstring;
begin

try

   AssignFile(fi,filename);
   Reset(fi);
   Read(fi,s);
   //memo1.Lines.Clear ;
   memo1.Lines.Add('Time        >> '+TimeToStr(time));
   memo1.Lines.Add('TextFile    >> '+s);
   hostname :=  copy(s,1,1);


     if  pos('1_on',s) >0  then
         data := 'A'
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
else if pos('1_off',s) >0  then
    data := 'B';
if  pos('2_on',s) >0  then
   data := data+'C'
else if  pos('2_off',s) >0 then
   data := data+'D';
if  pos('3_on',s) >0 then
   data := data+'E'
else if pos('3_off',s) >0 then
   data := data+'F';
finally
CloseFile(fi);

send51 := data ;
 if send51= 'ACE'  then
   send51 := 'A'
else if send51= 'ACF' then
    send51:= 'B'
else if send51= 'ADE' then
   send51 := 'C'
else if send51= 'ADF' then
   send51:= 'D'
else if send51= 'BCE' then
   send51:= 'E'
else if send51= 'BCF' then
   send51:= 'F'
else if send51= 'BDE' then
   send51:= 'G'
else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            send51:= 'H';
    end;
    end;


    procedure TForm1.Timer1Timer(Sender: TObject);
    var  m :Tstrings;
    begin
        m :=Memo1.Lines ;
        readfiles ;
        client.Host := '192.168.0.' +hostname;
        client.Open ;
        m.Append('Send To  >> '+client.Host );
        m.Append('Data send >> '+send51);
        Timer3.Enabled := true;
        m.Append('');
    end;


    procedure TForm1.Timer2Timer(Sender: TObject);
    begin
        client.Close ;
        Timer2.Enabled := false;
    end;


    procedure TForm1.Timer3Timer(Sender: TObject);
    begin
        client.Socket.SendText(data);
        Timer3.Enabled := false;
        Timer2.Enabled := true;
    end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure TForm1.BitBtn1Click(Sender: TObject);
begin
    memo1.Lines.Clear ;
end;

end.
```

****

โปรแกรมของไคลเอนต์

```
unit uclient;
interface
uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs,Menus,
    ExtCtrls, StdCtrls, DsFancyButton,ScktComp,winsock, CPort, CPortCtl, jpeg, Buttons;
const   tername = 'c:\dbclient.txt';
type
    TForm1 = class(TForm)
    Panel1: TPanel;
    Memo1: TMemo;
    ComPort1: TComPort;
    C : TBitBtn;
    Image1: TImage;
    Panel2 : TPanel;
    btn1 : TDsFancyButton;
    btn2 : TDsFancyButton;
        procedure btn1Click(Sender: TObject);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        procedure btn2Click(Sender: TObject);
        procedure CClick(Sender: TObject);
private
    { Private declarations }
public
    fi : textfile;{ Public declarations }
    server:Tserversocket; send51:shortstring;
    data:TCustomWinSocket; tmp:string;
      procedure onAccept(Sender: TObject; Socket: TCustomWinSocket);
      procedure onread(Sender: TObject; Socket: TCustomWinSocket);
end;
var
    Form1: TForm1;
implementation
    uses Math;
    {$R *.dfm}

procedure TForm1.btn1Click(Sender: TObject);
begin
    self.Caption := 'Client Run..';
    server := Tserversocket.Create(self);
    server.Port := 4000;
    server.OnAccept :=  onaccept;
    server.OnClientRead := onread;
    server.Open ;
    btn1.Enabled := false;
    btn2.Enabled := true;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure TForm1.btn2Click(Sender: TObject);
begin
    self.Caption := 'Client Stop..';
    server.Close ;
    btn1.Enabled := true;
    btn2.Enabled := false;
end;

procedure TForm1.onAccept(Sender: TObject; Socket: TCustomWinSocket);
begin
    data := socket;
end;

procedure TForm1.onread(Sender: TObject; Socket: TCustomWinSocket);
var tx:textfile;
begin
    send51 := data.ReceiveText ;
    if send51= 'ACE' then
      send51:= 'A'
    else if send51= 'ACF' then
      send51:= 'B'
    else if send51= 'ADE' then
      send51:= 'C'
    else if send51= 'ADF' then
      send51:= 'D'
    else if send51= 'BCE' then
      send51:= 'E'
    else if send51= 'BCF' then
      send51:= 'F'
    else if send51= 'BDE' then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        send51:= 'G'
      else
        send51:= 'H';
    Memo1.Lines.Append('Time >>' +TimeToStr(time));
    Memo1.Lines.Append('Recive data >>' +send51);
    ComPort1.Open ;
 try
    if FileExists(tername) then
begin
     AssignFile(tx,tername);
     Reset(tx);
 end;
    else
 begin
    AssignFile (tx,tername);
    Rewrite (tx);
end ;
    Append(tx);
    if not ( tmp = send51) then
begin
    Writeln(tx,send51+'      '+ timetostr(now)+'      '+datetostr(now));
    tmp := send51;
end;
    finally
    closefile(tx);
end;
   ComPort1.WriteStr(send51);
   ComPort1.Close ;
   Memo1.Lines.Append('');
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure TForm1.CClick(Sender: TObject);
begin
    memo1.Lines.Clear ;
end;

end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WAP Page

ในส่วนของ WAP Page ประกอบด้วยกัน 5 ส่วน อันได้แก่ คือ

Index.wml

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<!-- created by EasyPad WAPtor (http://www.waptop.net/) -->
<wml>

<card id="MainCard1" ontimer="#MainCard2" title="first card">
<timer value="7"/>
<p align="center"> Welcome to </p>
<p align="center"><b>WireLess Control</b></p>
<p align="center">Project</p>
<p align="left"><img src="sunny.wbmp" alt=" "/></p>
</card>

<card id="MainCard2" ontimer="#MainCard3" title="first card">
<timer value="9"/>
<p align="center"> Welcome to </p>
<p align="center"><b>WireLess Control</b></p>
<p align="center">Project</p>
<p align="center"><img src="sunny.wbmp" alt=" "/></p>
</card>

<card id="MainCard3" ontimer="#logo1" title="first card">
<timer value="7"/>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
<p align="center"> Welcome to </p>
<p align="center"><b>WireLess Control</b></p>
<p align="center">Project</p>
<p align="right"><img src="sunny.wbmp" alt=" "/></p>
</card>

<card id="logo1" ontimer="#logo2" title="KMITL LOGO">
<timer value="40"/>
<p align="center">
<img src="logokmitl.wbmp" alt="KMITL LOGO"/></p>
<p align="center">King Mongkut's Institute<br/> of <br/>Technology Ladkrabang
</p>
</card>

<card id="logo2" ontimer="kmitl2.wml" title="Advisor">
<timer value="25"/>
<p align="center"><img src="taworn.wbmp" alt="MY Advisor"/></p>
<p align="center"><b><big>MY Advisor</big></b></p>
</card>

</wml>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

kmitl2.wml

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<!-- created by EasyPad WAPtor (http://www.waptop.net/) -->
<wml>
<card id="Mainlogin" title="Login Menu">
<do type="accept" label="Begin">
<go href="index.wml"/>
        </do>
<do type="prev" label="Back"><prev/></do>
<p align="center"><b>Select Item</b><br/>
<select name="login">
<option value="login" onpick="#login">Login</option>
<option value="about us" onpick="#about">About Us</option>
<option value="help" onpick="#help">Help</option>
</select>
</p>
</card>

<card id="about" title="Abuot Us">
<do type="accept" label="Back">
<go href="kmitl2.wml"/>
        </do>
<p align="center"><small><b>This project created by</b><br/><br/>
Mr.Songsak Dookan<br/>
Mr.Weerachai Preedapattarapong<br/></small>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
</p>
<p align="center"><b>My advisor</b><br/>
<small>Mr.Taworn benjanarasut</small></p>
</card>

<card id="help" title="Help Page">
<do type="accept" label="Back">
<go href="kmitl2.wml"/>
        </do>
<p>
<b>Introduction</b><br/>
<b>1.</b><small>Select login.</small><br/>
<b>2.</b><small>Enter password.</small><br/>
<b>3.</b><small>Select the computer to control.</small><br/>
<b>4.</b><small>Select the engine turn on-off.</small><br/>
<b>5.</b><small>Next to Recheck Page.</small><br/>
<b>6.</b><small>Wait for respond... </small><br/>
<b>7.</b><small>Return loop to number 3.  </small>
</p>
</card>

<card id="login" title="PassWord" newcontext="true">
<do type="accept" label="Next">
 <go href="pass.wmls#check('$answer')"/>
</do>
<p align="left">
<b> Enter PassWord :</b>
 <input type="password" name="answer"/>
</p>
</card>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
<card id="trypass" ontimer="#Mainlogin" title="invalid" >
<timer value="30"/>
<p align="center"><br/><big>Invalid..!!<br/>PassWord</big></p>
<p align="center">Please. Try again!!</p>
</card>

</wml>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Kmitl3.wml

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<!-- created by EasyPad WAPtor (http://www.waptop.net/) -->
<wml>

<card id="intro" ontimer="#MainControl" title="Ready to Control">
<timer value="10"/>
<p align="center"><big><strong>Control Page</strong></big><br/>
Refesh</p>

</card>
<card id="MainControl" title="Select Com">
<p><b>Select Computer :</b><br/>
<a href="#com1">* Computer 1</a><br/>
<a href="#com2">* Computer 2</a>
</p>
</card>

<card id="com1" title="Sub Control#1">
<do type="accept" label="Next">
<go href="#confirm"/>
</do>
<do type="prev" label="Back"><prev/></do>
<p align="center"><b>IN COMPUTER 1</b><br/></p>
<p><b>Select Engine:</b><br/>
<select name="ctrl1">
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        <option value="11_on" >ON#1</option>
        <option value="11_off" >OFF#1</option>
</select><br/>
<select name="ctrl2">
        <option value="12_on" >ON#2</option>
        <option value="12_off" >OFF#2</option>
</select><br/>
<select name="ctrl3">
        <option value="13_on" >ON#3</option>
        <option value="13_off" >OFF#3</option>
</select>
</p>
</card>

<card id="com2" title="Sub Control#2">
<do type="accept" label="Next">
<go href="#confirm"/>
</do>
<do type="prev" label="Back"><prev/></do>
<p align="center"><b>IN COMPUTER 2</b><br/></p>
<p><b>Select Engine:</b><br/>

<select name="ctrl1">
        <option value="21_on" >ON#1</option>
        <option value="21_off" >OFF#1</option>
</select><br/>
<select name="ctrl2">
        <option value="22_on" >ON#2</option>
        <option value="22_off" >OFF#2</option>
</select><br/>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
<select name="ctrl3">
        <option value="23_on" >ON#3</option>
        <option value="23_off" >OFF#3</option>
</select>
</p>
</card>

<card id="confirm" title="Confirm Page">
<do type="accept" label="Yes">
 <go href="control.php?fctrl1=$(ctrl1)&amp;fctrl2=$(ctrl2)&amp;fctrl3=$(ctrl3)">
 </go>
 </do>
<do type="prev" label="Back"><prev/></do>
<p align="left"><b>Recheck Command:</b><br/>
Engine1: $ctrl1 <br/>
Engine2: $ctrl2 <br/>
Engine3: $ctrl3 <br/>
</p>
<p align="center"><big><b>Are You Sure!!!</b></big><br/>
Press (YES) <br/>To Send Command <br/>
Please Wait.....
</p>
</card>

</wml>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WML Script

```
extern function check(a) {
if(a!="12345")
  WMLBrowser.go("kmitl2.wml#trypass");
else
  WMLBrowser.go("kmitl3.wml");
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Script PHP

```
<?
    header("Content-type: text/vnd.wap.wml");
    echo("<?xml version=\"1.0\"?>\n");
    echo("<!DOCTYPE wml PUBLIC"//WAPFORUM//DTD WML 1.1//EN\"
    \"http://www.wapforum.org/DTD/wml_1.1.xml\">);
?>
<wml>
        <card id="Control" title="Result">
        <do type="accept" label="New control">
        <go href="kmitl3.wml#MainControl"/>
        </do>
        <p align="left"><b>
        <?
    $new = fopen("control.txt","w");
    fputs($new,"$fctrl1 $fctrl2 $fctrl3");
    fclose($new);

    echo"<strong>Command Success.!!</strong><br/><br/>";
    echo"-> $fctrl1 ..OK<br/>";
    echo"-> $fctrl2 ..OK<br/>";
    echo"-> $fctrl3 ..OK";
        ?>
        </b></p>

        </card >
</wml>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมของไมโครคอนโทรลเลอร์

```
;*********************************************************************
; Program          : Serial UART 02
; Description      : Serial UART Exchange Data with PC
;                  : using Bit polled check method
; For              : MCS-51 Microcontroller Training System
; Filename         : lab2502.asm
; Assembler        : RAD51
; Copyright (C) 2000 Innovative Experiment Co.,Ltd.
;*********************************************************************
;---------------------------------------------------------
; Define User Register
;---------------------------------------------------------
MONITOR          BIT     P1.0
;---------------------------------------------------------
; Main Program.
;---------------------------------------------------------
                 ORG     0000H           ; Reset Vector
MAIN:            MOV     TMOD,#020H      ; T1 8Bit Auto, T0 16Bit
                 MOV     TH1,#0FDH       ; 9600 bps Timer1 Default
                 SETB    TR1             ; Start Timer1
                 MOV     SCON,#050H      ; Mode1 RX

START_LOOP:      MOV     P1,#00H
                 SETB    MONITOR
                 CLR     RI

LOOP:            JNB     RI,$            ; Wait byte received
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                CLR     RI
                MOV     A,SBUF          ; Get byte from SBUF

MONI:           CPL     MONITOR

AA:             CJNE    A,#41H,BB
                ACALL   NNN

BB:             CJNE    A,#42H,CC
                ACALL   NNF

CC:             CJNE    A,#43H,DD
                ACALL   NFN

DD:             CJNE    A,#44H,EE
                ACALL   NFF

EE:             CJNE    A,#45H,FF
                ACALL   FNN

FF:             CJNE    A,#46H,GG
                ACALL   FNF

GG:             CJNE    A,#47H,HH
                ACALL   FFN

HH:             CJNE    A,#48H,LOOP
                ACALL   FFF

;----------LABLE:  NNN = P1.1 ->ON ,P1.2 ->ON  ,P1.3 ->ON
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
NNN:    SETB  P1.1
        SETB  P1.2
        SETB  P1.3
        RET

NNF:    SETB  P1.1
        SETB  P1.2
        CLR   P1.3
        RET

NFN:    SETB  P1.1
        CLR   P1.2
        SETB  P1.3
        RET

NFF:    SETB  P1.1
        CLR   P1.2
        CLR   P1.3
        RET

FNN:    CLR   P1.1
        SETB  P1.2
        SETB  P1.3
        RET

FNF:    CLR   P1.1
        SETB  P1.2
        CLR   P1.3
        RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
FFN:    CLR   P1.1
        CLR   P1.2
        SETB  P1.3
        RET

FFF:    CLR   P1.1
        CLR   P1.2
        CLR   P1.3
        RET

        END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. อภิชาติ ภู่พลับ และ สัจจะ จรัสรุ่งรวีวร. "เริ่มต้นเขียนโปรแกรมติดต่อ และควบคุมฮาร์ดแวร์ Visual Basic" , DEV BOOK , 2456
2. อรรถพร บุญยะโภคา , วรพจน์ กอแก้ววัฒนะกุล และ ชัยวัฒน์ ลิ้มพรจิตรวิไล , "เรียนรู้ และปฏิบัติการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกผ่านพอร์ตอนุกรม" , INEX
3. วรพจน์ กอแก้ววัฒนะกุล และ ชัยวัฒน์ ลิ้มพรจิตรวิไล , "เรียนรู้ และปฏิบัติการไมโครคอนโทรลเลอร์ MCS-51 แบบแพลช" , INEX
4. บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน) , "คู่มือดูขาไอซี"
5. สราวุธ อ้อยศรีกุล และ วิทยา ต่อศรีเจริญ. "เปิดมิติ Mobile Inter ด้วย WAP จากพื้นฐานการพัฒนา WAP Site ของจริง" , WITTY GROUP , 2544
6. สมประสงค์ ธิตินิลนิธิ , "เรียนลัด PHP 4" , PROVISION , สิงหาคม 2545
7. วันสุระ ศรีใสดี , "ประยุกต์/อินเตอร์เฟส ไมโครคอนโทรลเลอร์ ภาคปฏิบัติ MCS-51" , บริษัท สำนักพิมพ์ดวงกมล จำกัด , 2542
8. อดิศักดิ์ ชิณะวงศ์ , "MCS-51 ผ่านพอร์ตอนุกรม" , HOBBY ELECTRONICS , ฉบับที่ 127 เดือน มกราคม 2546
9. สัจจะ จรัสรุ่งรวีวร และ จักรพงษ์ สุขประเสริฐ , "คู่มือการสร้างแอพพลิเคชั่นด้วย Delphi 5.0 ฉบับสมบูรณ์" , อินโฟเพรส , 2543
10. สัลยุทธ์ สว่างวรรณ. "เครือข่ายคอมพิวเตอร์" , บริษัท ซีเอ็ดยูเคชั่น จำกัด(มหาชน) , 2542
11. ดร. พิพัฒน์ หิรัณย์วณิชชาการ. "ระบบการสื่อสารข้อมูล และเครือข่ายคอมพิวเตอร์" , บริษัท ซีเอ็ดยูเคชั่น จำกัด(มหาชน) , 2545
12. "MICROCOMPUTER" , ประจำเดือน JULY 2003
13. "MICROCOMPUTER" , ประจำเดือน AUGUST 2003
14. ข้อมูลจาก http://www.bluetooth.com
15. ข้อมูลจาก http://www.nokia.com
16. ข้อมูลจาก http://www.wapforum.org
17. ข้อมูลจาก http://www.symbian.com
18. ข้อมูลจาก http://www.apachesoftware.com
19. ข้อมูลจาก http://www.boardband.motorola.com
20. ข้อมูลจาก http://www.ericsson.com

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

21. ข้อมูลจาก http://www.eitcenter.com
22. ข้อมูลจาก http://www.cellula.co.za
23. ข้อมูลจาก http://www.epiconline.co.uk
24. ข้อมูลจาก http://www.m3gate.com
25. ข้อมูลจาก http://www.apache.org
26. ข้อมูลจาก http://www.php.com
27. ข้อมูลจาก http://www.pyweb.com
28. ข้อมูลจาก http://www.billionton.com
29. ข้อมูลจาก http://www.grouper.ieec.org

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้