สำนักหอสมุดกลาง_พระจอมเกล้าลาดกระบัง

ตัวจัดการอุปกรณ์สำหรับการทำงานในระบบเครือข่ายไร้สายแบบแอดฮอคผ่านบลูทูธ

Device Manager for Bluetooth Wireless Ad-hoc Network

T046189

นางสาวศักดิ์สุดา ฉันทเตยานนท์
นายสิทธิชัย เผือกวัฒนะ
นางสาวสุญาดา อาจไชยชาญ

ส.ท.
ส332ด
2544

เลขหมู่.....
เลขทะเบียน 46189
วัน, เดือน, ปี 20 มี.ค. 2546

.b.....
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2544

b11288619

ตัวจัดการอุปกรณ์สำหรับการทำงานในระบบเครือข่ายไร้สายแบบแอดฮอคผ่านบลูทูธ

Device Manager for Bluetooth Wireless Ad-hoc Network

โดย

นางสาวศักดิ์สุดา ฉันทเตยานนท์

นายสิทธิชัย เผือกวัฒนะ

นางสาวสุญาดา อาจไชยชาญ

อาจารย์ที่ปรึกษา

ผศ. อภิเนตร อูนากูล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

ปริญญานิพนธ์ปีการศึกษา 2544

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ตัวจัดการอุปกรณ์สำหรับการทำงานในระบบเครือข่ายไร้สายแบบแอดฮอคผ่านบลูทูธ

Device Manager for Bluetooth Wireless Ad-hoc Network

ผู้จัดทำ

1. นางสาวศักดิ์สุดา ฉันทเตยานนท์ รหัสประจำตัว 41014416
2. นายสิทธิชัย เผือกวัฒนะ รหัสประจำตัว 41014459
3. นางสาวสุญาดา อาจไชยชาญ รหัสประจำตัว 41014471



อาจารย์ที่ปรึกษา

(ผศ. อภิเนตร อูนากูล)

ตัวจัดการอุปกรณ์สำหรับการทำงานในระบบเครือข่ายไร้สายแบบแอดฮอคผ่านบลูทูธ

นางสาวศักดิ์สุตา ฉันทเตยานนท์ 41014416
นายสิทธิชัย เผือกวัฒนะ 41014459
นางสาวสุญาดา อาจไชยชาญ 41014471
ผศ. อภิเนตร อูนากูล อาจารย์ที่ปรึกษา
ปีการศึกษา 2544

บทคัดย่อ

บลูทูธเป็นเทคโนโลยีไร้สายที่ช่วยให้อุปกรณ์ต่างๆสามารถติดต่อสื่อสารกันได้โดยไม่ต้องคำนึงถึงแหล่งที่มาของผู้ผลิต ซึ่งก่อให้เกิดการทำงานในลักษณะของเครือข่ายแบบแอดฮอคขึ้น เครือข่ายแบบแอดฮอคนี้เป็นเครือข่ายที่มีความยืดหยุ่นสูง เกิดขึ้นในทันทีทันใด ไม่มีการเตรียมตัวมาก่อนและไม่มีการจัดการจากส่วนกลาง ดังนั้นอุปกรณ์ที่ทำงานอยู่ในลักษณะของเครือข่ายแบบนี้นั้น ต้องเป็นอุปกรณ์ที่มีความสามารถในการจัดการการเชื่อมต่อด้วยตนเองได้

ตัวจัดการอุปกรณ์ เป็นตัวที่ช่วยในการจัดการในเรื่องต่างๆของอุปกรณ์เพื่อให้สามารถทำงานได้อย่างมีประสิทธิภาพ เช่น การจัดการด้านประสิทธิภาพการใช้งาน การจัดการด้านความปลอดภัย การจัดการด้านคุณภาพของการบริการ หรือการจัดการด้านการปรับแต่งค่าของอุปกรณ์เพื่อให้สามารถรองรับการทำงานได้ เป็นต้น ในโครงงานนี้เป็นการพัฒนาตัวจัดการอุปกรณ์เพื่อรองรับความสามารถในการค้นหาอุปกรณ์ และการค้นหาการบริการที่มีอยู่ในเครือข่าย โดยตัวจัดการอุปกรณ์นี้จะช่วยให้อุปกรณ์บลูทูธสามารถที่จะค้นหาอุปกรณ์บลูทูธที่มีอยู่ในระยะเครือข่าย และสามารถค้นหาการบริการที่มีอยู่ในอุปกรณ์บลูทูธต่างๆเพื่อเรียกใช้การบริการต่อไปได้

Device Manager for Bluetooth Wireless Ad-hoc Network

Saksuta	Chanthateyanonth	41014416
Sittichai	Puagwatana	41014459
Suyada	Artchaichan	41014471
Asst.Prof.	Apinetr Unakul	Advisor

ABSTRACT

Bluetooth is a wireless communication technology that makes devices communicate to each other without consideration of different manufactory. Bluetooth brings about the working of ad-hoc network which is dynamic, instantaneous and not managed by the center. So devices that work in this environment must have ability to manage connection by themselves.

Device manager helps the management of devices. By providing the facilities such as performance management, security management, quality of service and configuration management. In this project, we develop device manager that provide facilities in device and service discovery by discovering devices that are placed in the area of Bluetooth ad-hoc network and finding their services.

กิตติกรรมประกาศ

การทำปริญญานิพนธ์ฉบับนี้คงไม่อาจเสร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือ และร่วมมือจากหลาย ๆ ฝ่ายด้วยกัน บุคคลแรกที่ต้องกล่าวถึงเพราะเป็นส่วนสำคัญที่ทำให้ปริญญานิพนธ์นี้เสร็จลงได้ก็คือ ผศ. อภิเนตร อูนากูล อาจารย์ที่ปรึกษาปริญญานิพนธ์ ที่ให้ความเอาใจใส่ แนะนำช่วยเหลือเสมอมา ซึ่งต้องขอขอบพระคุณเป็นอย่างสูง ขอขอบคุณอาจารย์ภาควิชาวิศวกรรมคอมพิวเตอร์ทุกๆท่านที่ให้ความรู้และคำปรึกษามาตลอดระยะเวลา 4 ปี ขอขอบคุณบริษัท อิริคสันประเทศไทยจำกัดที่เอื้อเฟื้อชุดพัฒนาบลูทูธมาใช้ในโครงงานนี้ และขอขอบคุณรุ่นพี่และเพื่อนๆทุกคนที่คอยให้คำปรึกษา ช่วยเหลือและให้กำลังใจตลอดมา

สุดท้ายนี้ต้องขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูผู้เขียนมาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่และยังให้กำลังใจ เอาใจใส่เสมอมา ในทุก ๆ ด้านอันหาที่เปรียบมิได้ ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณ และขอกราบขอบพระคุณมา ณ ที่นี้

ศักดิ์สุตา ฉันทเตยานนท์
สิทธิชัย เผือกวัฒนะ
สุญาดา อาจไชยชาญ

สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญภาพ	XI
สารบัญตาราง	XVI
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์ของปริญญานิพนธ์	2
1.3 ขอบเขตของปริญญานิพนธ์	2
1.4 ขั้นตอนการดำเนินงาน	2
บทที่ 2 เทคโนโลยีบลูทูธ	4
2.1 เทคโนโลยีบลูทูธ	4
2.1.1 จุดกำเนิดบลูทูธ	4
2.1.2 บลูทูธซิค (Bluetooth SIG)	4
2.1.3 อนาคตของบลูทูธ	5
2.2 ภาพรวมของบลูทูธ	5
2.2.1 ความหมายของบลูทูธ	5
2.2.2 เทคนิค Frequency Hopping	6
2.2.3 มาสเตอร์ สเลฟและ Frequency Hopping	7
2.2.4 ลักษณะการต่อโครงข่าย	8
2.2.5 ช่องทางการสื่อสารข้อมูล	10
2.2.6 ระบบความปลอดภัย	11
2.3 เทคโนโลยีไร้สายอื่น	12
2.3.1 IrDA	12
2.3.2 HomeRF	12
2.3.3 IEEE 802.11b (Wireless LAN)	13
2.3.4 ข้อได้เปรียบจากการใช้เทคโนโลยีบลูทูธ	13
2.4 บลูทูธโพรโตคอลสแต็ก	14
2.4.1 โพรโตคอลหลักของบลูทูธ	15

สารบัญ (ต่อ)

	หน้าที่
2.4.2 โพรโตคอลแทนที่การใช้สายเคเบิล	16
2.4.3 โพรโตคอลควบคุมการใช้งานในระบบโทรศัพท์	16
2.4.4 โพรโตคอลที่รับเอามาใช้	16
2.5 การนำบลูทูธไปใช้งาน	18
2.5.1 สถานะของบลูทูธในระบบไร้สาย	18
2.5.2 บลูทูธโพรไฟล์	18
2.5.2 ภาพการนำบลูทูธไปใช้งาน	23
2.6 ระบบเครือข่ายแบบแอดฮอค	26
บทที่ 3 บลูทูธโมดูล	30
3.1 เบสแบน (Baseband)	30
3.1.1 แอดเดรสอุปกรณ์บลูทูธ	31
3.1.2 มาสเตอร์ สเลฟ และพิโคเนต	31
3.1.3 การเชื่อมต่อทางกายภาพ	34
3.1.4 โครงสร้างแพ็กเกตบลูทูธ	35
3.1.4.1 ส่วนที่ใช้ในการเข้าถึง	35
3.1.4.2 ส่วนหัวข้อมูล	36
3.1.4.3 ส่วนข้อมูล	37
3.1.5 ชนิดของแพ็กเกต	38
3.1.6 ลอจิคัลแชนเนล	39
3.1.7 สรุปการทำงานของเบสแบน	40
3.2 ส่วนควบคุมการเชื่อมต่อ (LC : Link Controller)	40
3.2.1 สถานะของส่วนควบคุมการเชื่อมต่อ	41
3.2.2 การทำงานของสถานะควบคุมการเชื่อมต่อ	43
3.2.2.1 การค้นหาอุปกรณ์และการอินไควรี่	45
3.2.2.2 การสร้างการเชื่อมต่อและการเพจจิง	46
3.2.3 สรุปการทำงานของส่วนควบคุมการเชื่อมต่อ	47
3.3 ส่วนจัดการการเชื่อมต่อ (LM : Link Manager)	48
3.3.1 หน่วยข้อมูลโพรโตคอล (PDU : Protocol Data Unit)	48
3.3.2 แชนเนลส่วนจัดการการเชื่อมต่อ	48
3.3.3 การเริ่มต้นการเชื่อมต่อ	48
3.3.4 ปิดการเชื่อมต่อ	50

สารบัญ (ต่อ)

	หน้าที่
3.3.5 การสับเปลี่ยนหน้าที่	50
3.3.6 การควบคุมการใช้มัลติสลอตแพ็กเกต	51
3.3.7 ความปลอดภัย	52
3.3.8 โหมดประหยัดพลังงาน	52
3.3.9 การควบคุมพลังงาน	52
3.3.10 คุณภาพของการบริการ	53
3.3.11 ข้อมูลข่าวสาร	53
3.3.12 คุณสมบัติที่รองรับ	53
3.3.13 เวอร์ชัน LMP	53
3.3.14 การร้องขอชื่อ	53
3.3.15 สรุปการทำงานของส่วนจัดการการเชื่อมต่อ	54
3.4 ส่วนควบคุมการเชื่อมต่อโฮสต์ (HCI : the Host Controller Interface)	55
3.4.1 ชนิดของแพ็กเกตใน HCI	56
3.4.2 เลเยอร์ในการรับส่งข้อมูล HCI	56
3.4.2.1 โครงร่าง USB	57
3.4.2.2 เลเยอร์การรับส่งข้อมูลแบบอนุกรม	57
3.4.3 ส่วนควบคุมการไหลของข้อมูล	58
3.4.3.1 คำสั่งในการควบคุมการไหลของข้อมูล	58
3.4.3.2 ส่วนควบคุมการไหลของข้อมูลจากโฮสต์	58
3.4.3.3 ส่วนควบคุมการไหลของข้อมูลจากโมดูล	58
3.4.4 การปรับแต่งค่าในโมดูล	59
3.4.4.1 เวอร์ชัน	59
3.4.4.2 การร้องขอชื่ออุปกรณ์ระยะใกล้และไกล	60
3.4.4.3 คลาสของอุปกรณ์	60
3.4.4.4 คุณสมบัติที่รองรับ	60
3.4.4.5 รหัสประเทศ	61
3.4.4.6 HCI_Read_BD_ADDR	61
3.4.5 การอินไควรี่	61
3.4.5.1 การเริ่มต้นกระบวนการอินไควรี่	61
3.4.5.2 ส่วนควบคุมการตอบกลับการอินไควรี่	62
3.4.5.3 การหยุดกระบวนการอินไควรี่	62

สารบัญ (ต่อ)

	หน้าที่
3.4.5.4 ช่วงเวลาในการอินไควรี่	63
3.4.6 การอินไควรี่แสกน	64
3.4.6.1 GIAC และ LIAC	64
3.4.6.2 ช่วงเวลาในการอินไควรี่แสกน	64
3.4.6.3 การเริ่มทำการอินไควรี่แสกน	64
3.4.7 การเพจจิง	66
3.4.8 สรุปการทำงานของส่วนควบคุมการเชื่อมต่อโฮสต์	66
บทที่ 4 บลูทูธโฮสต์	67
4.1 L2CAP (Logical Link Control and Adaptation Protocol)	67
4.2 RFCOMM	68
4.2.1 พอร์ตอนุกรมและ UART	68
4.2.2 ชนิดของอุปกรณ์ RFCOMM	69
4.2.3 ชนิดของเฟรมใน RFCOMM	70
4.2.4 การเชื่อมต่อและการหยุดการเชื่อมต่อ	70
4.2.5 สรุปการทำงานของ L2CAP	72
4.3 โพรโตคอลค้นหาการบริการ (SDP : Service Discovery Protocol)	73
4.3.1 โมเดลไคลเอนต์ - เซิร์ฟเวอร์ของ SDP	74
4.3.2 ฐานข้อมูล SDP	75
4.3.2.1 แอททริบิวของการบริการ	75
4.3.2.2 ชนิดและขนาดของข้อมูล	77
4.3.2.3 เรคอดของการบริการ	77
4.3.3 การค้นหาเรคอด SDP	79
4.3.4 การค้นหากลุ่ม	79
4.3.5 UUID	80
4.3.5.1 UUID ที่ถูกกำหนดโดยมาตรฐานบลูทูธ	80
4.3.5.2 UUID ที่ถูกกำหนดโดยผู้ผลิต	80
4.3.6 ข้อความใน SDP	81
4.3.6.1 การค้นหาการบริการ	81
4.3.6.2 การค้นหาแอททริบิวของการบริการ	82
4.3.6.3 การค้นหาการบริการและแอททริบิวของการบริการ	83
4.3.6.4 ตัวควบคุมข้อผิดพลาด	84

สารบัญ (ต่อ)

หน้าที่

4.3.7 โพรไฟล์การค้นหาการบริการ 84
4.3.8 โปรแกรมประยุกต์การค้นหาการบริการ 84
4.3.9 สรุปการทำงานของโพรโตคอลค้นหาการบริการ 85
บทที่ 5 การออกแบบและหลักการทำงาน 87
5.1 โครงสร้างหลักของระบบ 87
5.2 ขอบเขตของตัวจัดการอุปกรณ์ 88
5.3 การค้นหาโพรโตคอลสแต็ก 88
5.3.1 OpenBT 89
5.3.2 BlueZ 90
5.3.3 Bluedrekar 92
5.3.4 Cstack 92
5.4 การออกแบบส่วนซอฟต์แวร์ 93
5.4.1 Use Case Diagram 93
5.4.2 Scenario 95
5.4.2.1 Sequence Diagram แสดง Set scan mode 95
5.4.2.2 Sequence Diagram แสดง Set bluetooth address 95
5.4.2.3 Sequence Diagram แสดง Set friendly name 96
5.4.2.4 Sequence Diagram แสดง Find all devices 96
5.4.2.5 Sequence Diagram แสดง Find specific devices 97
5.4.2.6 Sequence Diagram แสดง Find all services 97
5.4.2.7 Sequence Diagram แสดง Find specific services 98
5.4.2.8 Sequence Diagram แสดง Get my information 99
5.4.2.9 Sequence Diagram แสดง Add new device 99
5.4.2.10 Sequence Diagram แสดง Call service 100
5.4.2.11 Sequence Diagram แสดง Chat 100
5.4.2.12 Sequence Diagram แสดง File transfer 101
5.4.2.13 Sequence Diagram แสดง Disconnect 101
5.4.3 Activity Diagram 102
5.4.3.1 Activity Diagram แสดง Set scan mode 102
5.4.3.2 Activity Diagram แสดง Set bluetooth address 102
5.4.3.3 Activity Diagram แสดง Set friendly name 103

สารบัญ (ต่อ)

	หน้าที่
5.4.3.4 Activity Diagram แสดง Find all devices	103
5.4.3.5 Activity Diagram แสดง Find specific devices	104
5.4.3.6 Activity Diagram แสดง Find all services	105
5.4.3.7 Activity Diagram แสดง Find specific services	105
5.4.3.8 Activity Diagram แสดง Get my information	106
5.4.3.9 Activity Diagram แสดง Add new device	107
5.4.3.10 Activity Diagram แสดง Call service	108
5.4.3.11 Activity Diagram แสดง Chat	108
5.4.3.12 Activity Diagram แสดง File transfer	109
5.4.3.13 Activity Diagram แสดง Disconnect	109
5.4.4 Class Diagram	110
บทที่ 6 การทดสอบและผลการทดสอบ	112
6.1 วัตถุประสงค์ในการทดสอบ	112
6.2 ขั้นตอนการทดสอบ	112
6.2.1 การทดสอบความสามารถในการค้นหาอุปกรณ์	112
6.2.2 การทดสอบความสามารถในการค้นหาการบริการ	113
6.2.3 การทดสอบความสามารถในการปรับแต่งคุณสมบัติต่างๆ	113
6.2.4 การทดสอบการเรียกใช้การบริการ	114
6.3 อุปกรณ์ที่ใช้ในการทดสอบ	114
6.4 โปรแกรมที่ใช้ในการทดสอบ	115
6.5 ผลการทดสอบ	115
บทที่ 7 บทวิจารณ์และสรุป	122
7.1 บทวิจารณ์	122
7.1.1 ความสามารถในการค้นหาอุปกรณ์	122
7.1.2 การทดสอบความสามารถในการค้นหาการบริการ	122
7.1.3 การทดสอบความสามารถในการปรับแต่งคุณสมบัติต่างๆ	122
7.1.4 การทดสอบการเรียกใช้การบริการ	123
7.2 แนวทางการพัฒนาต่อ	123
7.2.1 การพัฒนาในส่วนของตัวจัดการอุปกรณ์	124
7.2.2 การพัฒนาในส่วนของการค้นหาการบริการ	126
7.3 สรุปสิ่งที่ได้จากการพัฒนาโครงงาน	132

สารบัญ (ต่อ)

	หน้าที่
ภาคผนวก	133
ภาคผนวก ก. Data Sheet	134
ภาคผนวก ข. Bluetooth Assigned Numbers	156
บรรณานุกรม	173

สารบัญภาพ

	หน้าที่
รูปที่ 2-1 การเชื่อมต่อไร้สายผ่านบลูทูธ	6
รูปที่ 2-2 การส่งข้อมูลแบบ FHSS	7
รูปที่ 2-3 Frequency Hopping ในเครือข่ายบลูทูธ	8
รูปที่ 2-4 โครงข่ายพิโคเนต	8
รูปที่ 2-5 โครงข่ายสแคตเตอร์เนตในกรณีที่อุปกรณ์เป็นได้ทั้งมาสเตอร์และสเลฟ	9
รูปที่ 2-6 โครงข่ายสแคตเตอร์เนตในกรณีที่อุปกรณ์เป็นสเลฟทั้ง 2 โครงข่าย	9
รูปที่ 2-7 ช่องทางการสื่อสารข้อมูล	10
รูปที่ 2-8 การรักษาความปลอดภัยในบลูทูธ	11
รูปที่ 2-9 เทคโนโลยีไร้สายที่น่าสนใจ	12
รูปที่ 2-10 บลูทูธโพรโตคอลสแต็ก	14
รูปที่ 2-11 โครงร่างของโพรโตคอล WAP	17
รูปที่ 2-12 การใช้งานอุปกรณ์บลูทูธผ่านโพรโตคอลสแต็ก	17
รูปที่ 2-13 ภาพการเชื่อมต่ออุปกรณ์ต่างๆผ่านบลูทูธ	18
รูปที่ 2-14 บลูทูธโพรไฟล์	19
รูปที่ 2-15 บลูทูธโพรโตคอลสแต็กสำหรับ SDP	21
รูปที่ 2-16 โมเดลของบลูทูธโพรโตคอลสแต็กแสดงการจำลองการทำงานผ่านพอร์ตอนุกรม	22
รูปที่ 2-17 โมเดลของบลูทูธโพรโตคอลสแต็กแสดงการทำงานของการแลกเปลี่ยนวัตถุ	23
รูปที่ 2-18 โพรโตคอลสแต็กสำหรับการรับส่งไฟล์	24
รูปที่ 2-19 โพรโตคอลสแต็กสำหรับอินเทอร์เน็ตบริดจ์	24
รูปที่ 2-20 โพรโตคอลสแต็กสำหรับการเข้าถึงระบบแลน	25
รูปที่ 2-21 โพรโตคอลสแต็กสำหรับการทำข้อมูลให้ตรงกัน	25
รูปที่ 2-22 โพรโตคอลสแต็กสำหรับโทรศัพท์ 3 ฟังก์ชัน	26
รูปที่ 2-23 โพรโตคอลสแต็กสำหรับเฮดเซทไร้สาย	26
รูปที่ 2-24 แสดงการใช้งานระบบเครือข่ายแบบแอดฮอคผ่านบลูทูธภายในสนามบิน	27
รูปที่ 2-25 การค้นหาอุปกรณ์บลูทูธ	29
รูปที่ 2-26 การค้นหาการบริการ	29
รูปที่ 3-1 ส่วนควบคุมการเชื่อมต่อและเบสแบน	30
รูปที่ 3-2 เวลาการทำงานในแพ็กเกตสลอตเดี่ยว	31
รูปที่ 3-3 รูปแบบต่างๆของสลอต	32
รูปที่ 3-4 พิโคเนตและสแคตเตอร์เนต	33

สารบัญภาพ (ต่อ)

หน้าที่

รูปที่ 3-5 การเชื่อมต่อแบบ ACL 34
รูปที่ 3-6 การเชื่อมต่อแบบ SCO 35
รูปที่ 3-7 โครงสร้างแพ็กเกตของบลูทูธ 35
รูปที่ 3-8 ส่วนที่ใช้ในการเข้าถึง CAC และDAC 36
รูปที่ 3-9 ส่วนที่ใช้ในการเข้าถึง IAC 36
รูปที่ 3-10 ส่วนหัวของข้อมูล 37
รูปที่ 3-11 ส่วนข้อมูล 38
รูปที่ 3-12 ลอจิคัลแชนเนล 39
รูปที่ 3-13 ส่วนควบคุมแชนเนล 39
รูปที่ 3-14 ส่วนแชนเนลของผู้ใช้ 40
รูปที่ 3-15 ไดอะแกรมแสดงสถานะของส่วนควบคุมการเชื่อมต่อ 43
รูปที่ 3-16 ไดอะแกรมแสดงสถานะของกระบวนการอินไควรี่ 44
รูปที่ 3-17 ไดอะแกรมแสดงสถานะของกระบวนการเพจจิง 44
รูปที่ 3-18 ลำดับการส่งข้อความในการอินไควรี่และการอินไควรี่แสกน 45
รูปที่ 3-19 ลำดับการส่งข้อความในการเพจจิงและเพจแสกน 46
รูปที่ 3-20 ลำดับข้อความ LMP ที่ใช้ในการเริ่มต้นการเชื่อมต่อแบบ ACL 49
รูปที่ 3-21 ลำดับข้อความ LMP ที่ใช้ในการเริ่มต้นการเชื่อมต่อแบบ SCO 49
รูปที่ 3-22 ลำดับข้อความ LMP ที่ใช้ในการทำ LMP_detach 50
รูปที่ 3-23 ลำดับข้อความ LMP ที่ใช้ในการสับเปลี่ยนหน้าที่เมื่อสเลฟเป็นตัวเริ่มต้น 50
รูปที่ 3-24 ลำดับข้อความ LMP ที่ใช้ในการสับเปลี่ยนหน้าที่เมื่อมาสเตอร์เป็นตัวเริ่มต้น 51
รูปที่ 3-25 ลำดับข้อความ LMP ที่ใช้ในการควบคุมการใช้มัลติสลอตแพ็กเกต 51
รูปที่ 3-26 ลำดับข้อความ LMP ที่ใช้ในการเปลี่ยนระดับพลังงาน 52
รูปที่ 3-27 สรุปการทำงานของ LMP 54
รูปที่ 3-28 ส่วนเชื่อมต่อ HCI บนบลูทูธ โพร โตคอลสแต็ก 55
รูปที่ 3-29 ชนิดของแพ็กเกตใน HCI 56
รูปที่ 3-30 การเชื่อมต่อสำหรับเลเยอร์การรับส่งข้อมูลแบบ UART 58
รูปที่ 3-31 ลำดับข้อความที่ใช้ในคำสั่งในการควบคุมการไหลของข้อมูล 59
รูปที่ 3-32 ลำดับข้อความที่ใช้ในคำสั่ง HCI_Read_Class_Of_Device 60
รูปที่ 3-33 ลำดับข้อความที่ใช้ในการทำอินไควรี่โดยเสร็จสมบูรณ์ 63
รูปที่ 3-34 เวลาที่ใช้ในการทำการอินไควรี่และการอินไควรี่แสกน 65
รูปที่ 3-35 ลำดับข้อความ HCI ในการอินไควรี่และอินไควรี่แสกน 65

สารบัญภาพ (ต่อ)

หน้าที่

รูปที่ 4-1 สถานะของ L2CAP ในบลูทูธโพรโตคอลสแต็ก 67
รูปที่ 4-2 อุปกรณ์ RFCOMM ชนิด 1 และ ชนิด 2 69
รูปที่ 4-3 การปฏิเสธการเชื่อมต่อ RFCOMM 70
รูปที่ 4-4 แชนเนล RFCOMM เมื่อเวลาหมด 71
รูปที่ 4-5 ลำดับข้อความสำหรับการสร้างการเชื่อมต่อ RFCOMM 72
รูปที่ 4-6 สถานะของ SDP ในบลูทูธโพรโตคอลสแต็ก 73
รูปที่ 4-7 การติดต่อระหว่างไคลเอนต์และเซิร์ฟเวอร์บน SDP 74
รูปที่ 4-8 โครงสร้างแอทริบิวของการบริการ 75
รูปที่ 4-9 เรคอดของการบริการ 78
รูปที่ 4-10 โครงสร้าง SDP สำหรับโทรศัพท์มือถือ 79
รูปที่ 4-11 การค้นหาการบริการ 81
รูปที่ 4-12 การค้นหาแอททริบิวของการบริการ 82
รูปที่ 4-13 การค้นหาการบริการและแอททริบิวของการบริการ 83
รูปที่ 4-14 ตัวควบคุมข้อผิดพลาด 84
รูปที่ 4-15 โครงร่างของโปรแกรมประยุกต์การค้นหาการบริการที่เป็นไปได้ 85
รูปที่ 4-16 ลำดับขั้นตอนในการเชื่อมต่อ SDP 86
รูปที่ 5-1 โครงสร้างหลักของระบบ 87
รูปที่ 5-2 แสดงสถิติจำนวนการเข้าชมเว็บเพจของโครงการและจำนวนผู้ใช้
ที่ดาวน์โหลด OpenBT ไปทดลองใช้งานในช่วงเวลาย้อนหลังไป 31 วัน 90
รูปที่ 5-3 แสดงสถิติจำนวนการเข้าชมเว็บเพจของโครงการและจำนวนผู้ใช้
ที่ดาวน์โหลด OpenBT ไปทดลองใช้งานในช่วงเวลาย้อนหลังไป 7 วัน 90
รูปที่ 5-4 แสดงสถิติจำนวนการเข้าชมเว็บเพจของโครงการและจำนวนผู้ใช้
ที่ดาวน์โหลด BlueZ ไปทดลองใช้งานในช่วงเวลาย้อนหลังไป 31 วัน 91
รูปที่ 5-5 แสดงสถิติจำนวนการเข้าชมเว็บเพจของโครงการและจำนวนผู้ใช้
ที่ดาวน์โหลด BlueZ ไปทดลองใช้งานในช่วงเวลาย้อนหลังไป 7 วัน 91
รูปที่ 5-6 โพรโตคอลสแต็กของ BlueDrekar 92
รูปที่ 5-7 Use Case Diagram ของระบบ 94
รูปที่ 5-8 Sequence Diagram แสดง Set scan mode 95
รูปที่ 5-9 Sequence Diagram แสดง Set bluetooth address 95
รูปที่ 5-10 Sequence Diagram แสดง Set friendly name 96
รูปที่ 5-11 Sequence Diagram แสดง Find all devices 96

สารบัญภาพ (ต่อ)

หน้าที่

รูปที่ 5-12 Sequence Diagram แสดง Find specific devices 97
รูปที่ 5-13 Sequence Diagram แสดง Find all services 98
รูปที่ 5-14 Sequence Diagram แสดง Find specific services 98
รูปที่ 5-15 Sequence Diagram แสดง Get my information 99
รูปที่ 5-16 Sequence Diagram แสดง Add new device 99
รูปที่ 5-17 Sequence Diagram แสดง Call service 100
รูปที่ 5-18 Sequence Diagram แสดง Chat 100
รูปที่ 5-19 Sequence Diagram แสดง File transfer 101
รูปที่ 5-20 Sequence Diagram แสดง Disconnect 101
รูปที่ 5-21 Activity Diagram แสดง Set scan mode 102
รูปที่ 5-22 Activity Diagram แสดง Set bluetooth address 102
รูปที่ 5-23 Activity Diagram แสดง Set friendly name 103
รูปที่ 5-24 Activity Diagram แสดง Find all devices 103
รูปที่ 5-25 Activity Diagram แสดง Find specific devices 104
รูปที่ 5-26 Activity Diagram แสดง Find all services 105
รูปที่ 5-27 Activity Diagram แสดง Find specific services 106
รูปที่ 5-28 Activity Diagram แสดง Get my information 107
รูปที่ 5-29 Activity Diagram แสดง Add new device 107
รูปที่ 5-30 Activity Diagram แสดง Call service 108
รูปที่ 5-31 Activity Diagram แสดง Chat 108
รูปที่ 5-32 Activity Diagram แสดง File transfer 109
รูปที่ 5-33 Activity Diagram แสดง Disconnect 109
รูปที่ 5-34 คลาสไดอะแกรมของระบบ 110
รูปที่ 6-1 ส่วนประกอบของโปรแกรม 115
รูปที่ 6-2 หน้าต่างการกำหนดค่าแอดเดรส 115
รูปที่ 6-3 หน้าต่างการกำหนดโหมดการแสกน 116
รูปที่ 6-4 หน้าต่างแสดงข้อมูลของอุปกรณ์ 116
รูปที่ 6-5 แสดงรายชื่ออุปกรณ์ที่ค้นพบ 117
รูปที่ 6-6 แสดงรายชื่อการบริการในอุปกรณ์ที่ค้นพบ 117
รูปที่ 6-7 แสดงรายชื่ออุปกรณ์ที่ถูกเพิ่มชื่อไปในลิสต์ 118
รูปที่ 6-8 แสดงการค้นหาอุปกรณ์โดยใส่ตัวอักษรที่ทราบ 119

สารบัญภาพ (ต่อ)

หน้าที่

รูปที่ 6-9 แสดงการค้นหาการบริการโดยใส่ตัวอักษรที่ทราบ 119

รูปที่ 6-10 แสดงหน้าต่างการบริการรับส่งข้อความ 120

รูปที่ 6-11 แสดงหน้าต่างการบริการรับส่งข้อความของอุปกรณ์อีกฝั่งหนึ่ง 120

รูปที่ 6-12 หน้าต่างการร้องขอที่ไปปรากฏฝั่งเซิร์ฟเวอร์ 120

รูปที่ 6-13 หน้าต่างแสดงข้อความเมื่อเซิร์ฟเวอร์ไม่อนุญาตให้ใช้การบริการรับส่งไฟล์ 121

รูปที่ 6-14 หน้าต่างกำหนดค่าการรับส่งไฟล์เมื่อเซิร์ฟเวอร์อนุญาตให้ใช้การบริการรับส่งไฟล์ 121

รูปที่ 7-1 การเชื่อมต่อในรูปแบบต่างๆ 124

รูปที่ 7-1 SLP เอเจนต์ และทรานแซคชันสำหรับการค้นหาการบริการและการลงทะเบียน 127

รูปที่ 7-2 การค้นหาการบริการที่ไม่มี DA 128

รูปที่ 7-3 องค์ประกอบของเครือข่ายจินี่ 129

รูปที่ 7-4 การบริการทำการค้นหาตารางในการค้นหาการบริการ 129

รูปที่ 7-5 ตารางในการค้นหาการบริการตอบรับด้วยออบเจคเซอร์วิส 129

รูปที่ 7-6 การบริการส่งออบเจคของการบริการไปลงทะเบียนที่ลุกอัพรีจิสทรา 130

รูปที่ 7-7 ไคลเอนต์ได้รับออบเจคของการบริการ 130

รูปที่ 7-8 ไคลเอนต์ติดต่อกับการบริการผ่านออบเจคของการบริการ 130

รูปที่ 7-9 โมเดลของ Salutation Manager 131

สารบัญตาราง

	หน้าที่
ตารางที่ 2-1 ความถี่ที่ใช้ในบลูทูธ	6
ตารางที่ 2-2 ความเร็วในการรับส่งข้อมูลในลักษณะต่างๆ	10
ตารางที่ 2-3 เปรียบเทียบเทคโนโลยีไร้สายที่ใช้ในระยะสั้น	13
ตารางที่ 3-1 ชนิดของแพ็กเกตที่ใช้ในบลูทูธ	38
ตารางที่ 3-2 การกำหนดเงื่อนไขการสแกน	66
ตารางที่ 4-1 เรคอดของการบริการหูฟังไร้สายบลูทูธ	78
ตารางที่ 4-2 เรคอดการบริการในการค้นหาแบบกลุ่มสำหรับโทรศัพท์มือถือ	80
ตารางที่ 7-1 เปรียบเทียบโพรโตคอลค้นหาการบริการต่างๆ	132

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มา

ปัจจุบันเป็นยุคของข้อมูลข่าวสาร การติดต่อสื่อสาร รวมทั้งการเจรจาทางธุรกิจก่อให้เกิดการแลกเปลี่ยนข้อมูลข่าวสาร ข้อมูลข่าวสารเป็นสิ่งที่ต้องการทั้งในภาคธุรกิจ และภาคเอกชน ยิ่งในภาคของธุรกิจนั้น การได้ข้อมูลที่ถูกต้อง รวดเร็ว และแม่นยำถือได้ว่าเป็นปัจจัยสำคัญที่มีผลต่อความสำเร็จของธุรกิจนั้นๆ ดังนั้น อุปกรณ์พกพาต่างๆ เช่น ปาล์ม โทรศัพท์มือถือ และโน้ตบุ๊ก จึงเกิดขึ้นมาเพื่อช่วยในการติดต่อสื่อสาร รวมทั้งช่วยในการบริหารข้อมูลข่าวสารที่มีอยู่เป็นจำนวนมากให้เป็นระเบียบเรียบร้อย

การเก็บข้อมูลของอุปกรณ์พกพาต่างๆนั้น มีความสามารถที่จะเก็บข้อมูลไว้ได้ในอุปกรณ์หลายชนิด เพื่อประโยชน์ในการแชร์ข้อมูลและเป็นการสำรองข้อมูลไปในตัวด้วย จากลักษณะเช่นนี้ จะเห็นได้ว่าการทำข้อมูลให้ตรงกันเป็นเรื่องสำคัญ เพื่อให้ข้อมูลที่มีอยู่ในอุปกรณ์ต่างๆตรงกัน ปัจจุบันการทำข้อมูลให้ตรงกันเป็นเรื่องที่ยุ่งยาก เสียเวลา และต้องใช้สายเคเบิลที่เหมาะสมเพื่อให้เกิดการโอนถ่ายข้อมูล ผลจากความยุ่งยากเหล่านี้อาจทำให้ข้อมูลที่มีอยู่สูญหาย หรือได้รับข้อมูลที่ผิดพลาดได้ ซึ่งค่าใช้จ่ายในการบำรุงรักษาหรือค่าใช้จ่ายในการซ่อมแซมส่วนนี้เป็นค่าใช้จ่ายที่ค่อนข้างสูง ปัญหาอีกอย่างหนึ่งซึ่งเกิดจากการที่จำนวนของผู้ใช้คอมพิวเตอร์และอุปกรณ์สื่อสารต่างๆมีการเพิ่มขึ้นอย่างต่อเนื่อง คือ จะมีวิธีการเชื่อมต่ออุปกรณ์ต่างๆเหล่านี้อย่างไร

เพื่อที่จะแก้ปัญหาข้อจำกัดต่างๆเหล่านี้ บริษัทต่างๆที่อยู่ในอุตสาหกรรมคอมพิวเตอร์และโทรคมนาคมจึงได้ร่วมมือกันสร้างเทคโนโลยีใหม่ขึ้นมา นั่นคือที่มาของ เทคโนโลยีบลูทูธ ซึ่งถูกพัฒนามาจากกลุ่มซิค (SIG : Bluetooth Special Interest Group) บลูทูธถูกพัฒนาขึ้นมาเพื่อแก้ปัญหาความยุ่งยากในการเชื่อมต่ออุปกรณ์ต่างๆเข้าด้วยกันและความไม่ลงรอยกันของข้อมูลที่แชร์ร่วมกันระหว่างอุปกรณ์พกพาต่างๆ โดยทำให้การเชื่อมต่ออุปกรณ์ต่างๆสะดวกและง่ายขึ้นโดยที่ไม่ต้องพึ่งพาสายเคเบิล ข้อมูลในอุปกรณ์ต่างๆตรงกันและสามารถใช้งานร่วมกันได้

คุณสมบัติเด่นอย่างหนึ่งที่มีอยู่ในบลูทูธ คือ สามารถทำงานเป็นเครือข่ายแบบแอดฮอค (Ad-hoc Network) ซึ่งเป็นคุณสมบัติสำคัญที่จะช่วยให้ชีวิตประจำวันของคนเราง่ายขึ้นมาก ระบบเครือข่ายแบบแอดฮอคผ่านบลูทูธนี้จะช่วยลดความยุ่งยากในการเชื่อมต่ออุปกรณ์ต่างๆเข้าด้วยกัน รวมทั้งช่วยให้การทำงานร่วมกันกับอุปกรณ์อื่นเป็นไปได้อย่างสะดวก เพราะเป็นระบบเครือข่ายที่ยืดหยุ่นในการติดตั้ง ไม่ต้องใช้ระบบศูนย์กลางในการทำงานร่วมกันระหว่างอุปกรณ์ต่างๆ อุปกรณ์แต่ละชิ้นจะสามารถรู้จักและค้นหาการบริการที่มีอยู่ในเครือข่ายขณะนั้น การเรียกใช้บริการต่างๆที่มีอยู่ในเครือข่ายเป็นเรื่องที่ง่าย สะดวกและรวดเร็ว ผู้ใช้ไม่

มีความจำเป็นต้องมาปรับแต่งค่าต่างๆให้ยุ่งยาก โดยที่ตัวอุปกรณ์บลูทูธเองมีความสามารถในการรับทราบเมื่ออุปกรณ์ที่เรียกใช้งานอยู่ออกจากเครือข่ายก็จะทำการยกเลิกการบริการที่ใช้งานอยู่นั้นออกไป

1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. เพื่อพัฒนาตัวจัดการอุปกรณ์ในส่วนของการค้นหาอุปกรณ์และการบริการที่มีอยู่ในเครือข่าย
2. เพื่อแสดงความสามารถในการสร้างเครือข่ายแบบแอดฮอคผ่านบลูทูธ
3. เพื่อสร้างแอพพลิเคชันบนโพรโตคอลสแต็กที่มีอยู่ให้สามารถนำไปใช้งานได้จริง
4. เพื่อศึกษาโครงสร้างการทำงานโพรโตคอลสแต็กของบลูทูธ

1.3 ขอบเขตของปริญญานิพนธ์

ปริญญานิพนธ์นี้ได้จัดทำขึ้นเพื่อพัฒนาตัวจัดการอุปกรณ์ในส่วนของการค้นหาอุปกรณ์และการบริการที่มีอยู่ในเครือข่ายเพื่อแสดงความสามารถของบลูทูธในการทำงานเป็นระบบเครือข่ายแบบแอดฮอค ซึ่งเป็นเครือข่ายที่ไม่ต้องพึ่งพาการจัดการจากส่วนกลาง โดยอุปกรณ์ที่อยู่ในเครือข่าย จะมีความสามารถในการค้นหาอุปกรณ์ต่างๆที่อยู่ในเครือข่ายเดียวกัน และสามารถค้นหาได้ว่ามีการบริการอะไรให้ใช้งานได้บ้าง พร้อมทั้งสิ้นสุดการเชื่อมต่อกับอุปกรณ์ในกรณีที่อุปกรณ์เหล่านั้นออกจากระยะของเครือข่าย

การที่จะทำให้อุปกรณ์ต่างๆสามารถทำงานได้ตามที่กล่าวมานั้นจำเป็นต้องอาศัยการทำงาน 2 ส่วนคือ ส่วนของตัวจัดการอุปกรณ์และส่วนโพรโตคอลสแต็ก ซึ่งทั้ง 2 ส่วนต้องทำงานร่วมกัน ดังนั้นการทำปริญญานิพนธ์จึงต้องมีการศึกษาและพัฒนาทั้ง 2 ส่วนนี้ โดยในส่วนของโพรโตคอลสแต็กจะเป็นส่วนที่กำหนดโครงสร้างพื้นฐานการทำงานต่างๆ และในส่วนของตัวจัดการอุปกรณ์จะเป็นตัวที่ติดต่อสื่อสารกับโพรโตคอลสแต็กเพื่อแสดงความสามารถในการค้นหาอุปกรณ์และการบริการที่มีอยู่ในบลูทูธขึ้นมาแสดงให้เห็นได้

1.4 ขั้นตอนการดำเนินงาน

การดำเนินงานเริ่มต้นจากการศึกษาทฤษฎีพื้นฐานเทคโนโลยีบลูทูธ โดยการศึกษาการทำงานของโพรโตคอลสแต็กในแต่ละเลเยอร์ที่จำเป็นในการทำปริญญานิพนธ์นี้ แล้วจึงทำการค้นหาโพรโตคอล สแต็กที่เหมาะสม นำมาศึกษาโครงสร้าง รายละเอียดและการทำงาน จากนั้นจึงพัฒนาในส่วนของตัวจัดการอุปกรณ์ซึ่งจะเป็นส่วนที่ติดต่อสื่อสารกับโพรโตคอลสแต็กเพื่อแสดงความสามารถในการค้นหาอุปกรณ์และการบริการที่มีอยู่ในบลูทูธขึ้นมาแสดงให้เห็นได้ ซึ่งมีขั้นตอนการดำเนินงานดังต่อไปนี้

1. กำหนดขอบเขตและคุณสมบัติพื้นฐานของตัวจัดการอุปกรณ์ที่จะสร้างขึ้น
2. ทำการศึกษาโพรโตคอลสแต็กในส่วนต่างๆ ถึงลักษณะการทำงานโดยรวมของเลเยอร์ทุกเลเยอร์ เพื่อสามารถระบุเลเยอร์ที่เกี่ยวข้องกับตัวจัดการอุปกรณ์ให้ได้

3. ทำการศึกษาเลเยอร์ที่เกี่ยวข้องกับของตัวจัดการอุปกรณ์โดยละเอียด และต้องพยายามระบุให้ได้ว่าการทำงานตรงส่วนไหนหรือคำสั่งไหนจำเป็นที่จะต้องใช้กับของตัวจัดการอุปกรณ์ที่จะสร้างขึ้น ซึ่งอาจจะทำเป็นเอกสารขึ้นมาเพื่อไว้อ้างอิงเมื่อถึงขั้นตอนการนำไปปฏิบัติจริง
4. ทำการค้นหาโพรโตคอลสแต็กที่เหมาะสมจากแหล่งข้อมูลต่างๆ ซึ่งโพรโตคอลสแต็กที่เหมาะสมนั้นต้องรองรับเลเยอร์ที่เกี่ยวข้องกับของตัวจัดการอุปกรณ์และมีความสมบูรณ์ในระดับหนึ่ง
5. ทำการศึกษาโครงสร้างโพรโตคอลสแต็กที่หามาได้ และทำการทดสอบดูว่าในแต่ละเลเยอร์นั้นมีความสมบูรณ์พอที่จะใช้มากน้อยแค่ไหน และถ้าตรงส่วนไหนไม่สมบูรณ์พอที่จะใช้ ต้องทำการพัฒนาตรงส่วนนั้นให้สามารถใช้งานตามที่ต้องการได้
6. ทำการออกแบบโครงสร้างโดยรวมของของตัวจัดการอุปกรณ์ ทั้งส่วนของตัวจัดการอุปกรณ์เองและส่วนที่ต้องทำการติดต่อกับโพรโตคอลสแต็ก
7. ทำการออกแบบฟังก์ชันและลงลึกไปถึงอัลกอริทึม แล้วทำการเขียนโปรแกรม
8. นำตัวจัดการอุปกรณ์ที่เขียนขึ้นมารวมกับโพรโตคอลสแต็กแล้วทำการทดสอบ ถ้าพบว่ายังมีข้อผิดพลาดอยู่ก็นำไปแก้ไข
9. สร้างโปรแกรมประยุกต์ (การบริการ) ขึ้นมาเพื่อทดสอบการทำงานของตัวจัดการอุปกรณ์

เมื่อได้ทำการทดลองและแก้ไขข้อผิดพลาดต่างๆเรียบร้อยแล้ว จึงนำมาสรุปผลการดำเนินงานของปริญญานิพนธ์นี้ รวมถึงหาแนวทางในการปรับปรุงและพัฒนาในอนาคต

บทที่ 2
เทคโนโลยีบลูทูธ

2.1 เทคโนโลยีบลูทูธ

บลูทูธเป็นเทคโนโลยีไร้สายที่ช่วยให้อุปกรณ์พกพาต่างๆ ไม่ว่าจะเป็น โทรศัพท์มือถือ, ปาล์ม หรือโน้ตบุก รวมทั้งอุปกรณ์ที่เป็นลักษณะของเครือข่ายส่วนบุคคล (Personal Area Network : PAN) เช่น เดสก์ทอปคอมพิวเตอร์ พรินเตอร์ หรือกล้องถ่ายรูปดิจิตอล สามารถติดต่อสื่อสารกันได้ โดยที่สามารถที่จะเรียกใช้งานระหว่างอุปกรณ์ต่างๆได้ โดยไม่ต้องคำนึงแหล่งที่มาของผู้ผลิตและความเข้ากันได้ของอุปกรณ์ต่างๆ

2.1.1 จุดกำเนิดบลูทูธ

การค้นคว้าศึกษาเทคโนโลยีบลูทูธเริ่มขึ้นในปี 1994 เมื่ออีริคสันค้นพบวิธีการเชื่อมต่อโดยใช้คลื่นความถี่วิทยุเชื่อมต่ออุปกรณ์ต่างๆที่ใช้งานร่วมกันกับโทรศัพท์มือถือ ประเด็นในการศึกษาครั้งนั้น คือ ราคา, ความสามารถในการใช้งาน และขนาด เพื่อที่จะนำมาใช้แทนที่การเชื่อมต่ออุปกรณ์ต่างๆกับโทรศัพท์มือถือที่ต้องใช้สายเคเบิล เริ่มจากการกำหนดคลื่นความถี่วิทยุที่จะนำมาใช้งาน เหตุที่เลือกใช้คลื่นวิทยุ เพราะในการใช้งานคลื่นวิทยุไม่ต้องคำนึงถึงทิศทาง และไม่จำเป็นต้องเชื่อมต่อแบบเป็นเส้นตรง ส่วนความสามารถอื่นๆที่ต้องรองรับได้ เช่น ความสามารถในการรับส่งข้อมูลทั้งที่เป็นข้อมูลธรรมดาและข้อมูลที่เป็นเสียง

ผลจากการศึกษาค้นคว้าในครั้งนั้นก่อให้เกิดข้อกำหนดโครงสร้างของบลูทูธ เวอร์ชัน 1.0 (Specification for Bluetooth Version 1.0) ขึ้นมาในเดือนกรกฎาคม ปี 1999 ซึ่งชื่อบลูทูธนี้ได้มาจากกษัตริย์ฮาราล บลาทาน (Harald Blatand) ซึ่งเป็นกษัตริย์ไวกิ้งที่รวบรวมเดนมาร์กและนอรเวย์เข้าด้วยกันในช่วงศตวรรษที่ 10 ชื่อนี้จึงถูกนำมาใช้ในเทคโนโลยีไร้สายใหม่นี้ เพื่อแสดงให้เห็นว่าเป็นเทคโนโลยีที่ทำให้อุปกรณ์ต่างๆทางด้านสื่อสารและอุปกรณ์ทางคอมพิวเตอร์สามารถติดต่อสื่อสารกันได้ โดยไม่ต้องคำนึงถึงแหล่งที่ผลิตอุปกรณ์ขึ้นมา

2.1.2 บลูทูธซิค (Bluetooth SIG)

กลุ่มบลูทูธซิค (Bluetooth Special Interest Group, SIG) เป็นกลุ่มของบริษัทต่างๆที่รวมตัวกันขึ้นมาเพื่อร่วมกันประชาสัมพันธ์และกำหนดรายละเอียดโครงสร้างของบลูทูธ บลูทูธซิคก่อตั้งขึ้นในเดือนกุมภาพันธ์ ปี 1998 โดยมีแกนหลัก คือ

- Ericsson Mobile Communications AB.
- Intel Corp.
- IBM Corp.

- Toshiba Corp.
- Nokia Mobile Phones.

ในเดือนพฤษภาคม ปี 1998 แกนหลักของกลุ่มซิคได้ประกาศเชิญชวนบริษัทต่างๆที่สนใจเข้าร่วมมาเป็นสมาชิกเพื่อร่วมกันพัฒนากำหนดรายละเอียดโครงสร้างของบลูทูธ จนกระทั่งในเดือนกรกฎาคม ปี 1999 จึงได้ทำการเผยแพร่รายละเอียดโครงสร้างของบลูทูธ เวอร์ชัน 1.0 ในเว็ปไซต์ http://www.bluetooth.com และในเดือนธันวาคม ปี 1999 ได้มีแกนหลักสำคัญที่เข้าร่วมมาเป็นกลุ่มซิค ดังนี้ คือ

- Microsoft.
- Lucent.
- 3Com.
- Motorola.

ปัจจุบันกลุ่มซิคมีจำนวนสมาชิกที่เข้าร่วมไม่ต่ำกว่า 2000 ราย และยังคงร่วมกันทำงานเพื่อกำหนดรายละเอียดโครงสร้างของบลูทูธในเวอร์ชันต่อไป

2.1.3 อนาคตของบลูทูธ

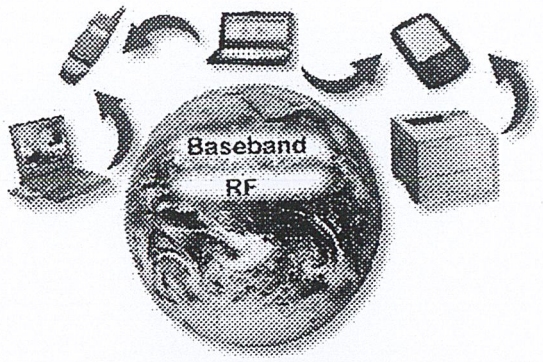
กลุ่มบลูทูธซิคได้มีความตั้งใจที่จะพัฒนารายละเอียดโครงสร้างของบลูทูธต่อไปใน 3 ประเด็น คือ ในเรื่องการแก้ไขและอธิบายรายละเอียดโครงสร้างของบลูทูธเวอร์ชัน 1.0 ให้กระจ่างมากยิ่งขึ้น, การพัฒนาในส่วนของโพรไฟล์ใหม่ๆ และการพัฒนาประสิทธิภาพของเบสแบนด์และคลื่นวิทยุในรายละเอียดโครงสร้างของบลูทูธเวอร์ชัน 2.0 ต่อไป โดยในเวอร์ชัน 2.0นี้จะเพิ่มประสิทธิภาพทางด้านการรับส่งข้อมูลที่รวดเร็วขึ้น (2-10 เมกะบิตต่อวินาที) เพื่อรองรับการรับส่งข้อมูลที่เป็นลักษณะของมัลติมีเดีย

2.2 ภาพรวมของบลูทูธ

ในหัวข้อนี้จะกล่าวถึงรายละเอียดเบื้องต้นในส่วนต่างๆของบลูทูธ ซึ่งจะช่วยให้มีความเข้าใจในเทคโนโลยีบลูทูธมากยิ่งขึ้น

2.2.1 ความหมายของบลูทูธ

บลูทูธเป็นมาตรฐานในการเชื่อมต่อแบบไร้สาย โดยใช้คลื่นความถี่วิทยุ ในย่านของ ISM Band (Industrial–Scientific-Medical) ซึ่งย่านความถี่นี้เป็นย่านความถี่ที่ไม้ต้องจดทะเบียนใช้งาน เป็นย่านความถี่อิสระที่เราสามารถนำมาใช้งานได้ โดยย่านความถี่นี้จะอยู่ในช่วง 2.4 กิกะเฮิร์ตซ ใช้หลักการของ Frequency Hopping Channel ในการส่งข้อมูล ซึ่งอุปกรณ์ที่ใช้เทคโนโลยีบลูทูธนั้น โดยทั่วไปจะทำงานอยู่ในช่วง 10 เมตร และอาจถึง 100 เมตรได้ ขึ้นอยู่กับกำลังส่ง ช่องทางการสื่อสารรองรับทั้งการขนส่งข้อมูลแบบ Synchronous (ข้อมูลที่เป็นเสียง) และ Asynchronous (ข้อมูลทั่วไป) ซึ่งช่องทางการสื่อสารนั้น มีแบนด์วิธทั้งหมด 1 เมกะบิตต่อวินาที



รูปที่ 2-1 การเชื่อมต่อไร้สายผ่านบลูทูธ

ในช่วงความถี่ 2.4 กิกะเฮิร์ตซ นี้ บลูทูธจะแบ่งช่องสัญญาณออกเป็นหลายๆช่อง แต่ละช่องมีขนาด 1 MHz โดยใช้หลักการของ GFSK (Guassian Frequency Shift Keying) ทำให้ได้ขนาดแบนด์วิธเท่ากับ 1 เมกะบิต ต่อวินาที

Country	Frequency Range	RF Frequency	Channels
US & Europe	2400 - 2483.5 MHz	f = 2402 + k MHz	k = 0,...,78
Japan	2471 - 2497 MHz	f = 2473 + k MHz	k = 0,...,22
Spain	2445 - 2475 MHz	f = 2449 + k MHz	k = 0,...,22
France	2446.5 - 2483.5 MHz	f = 2454 + k MHz	k = 0,...,22

ตารางที่ 2-1 ความถี่ที่ใช้ในบลูทูธ

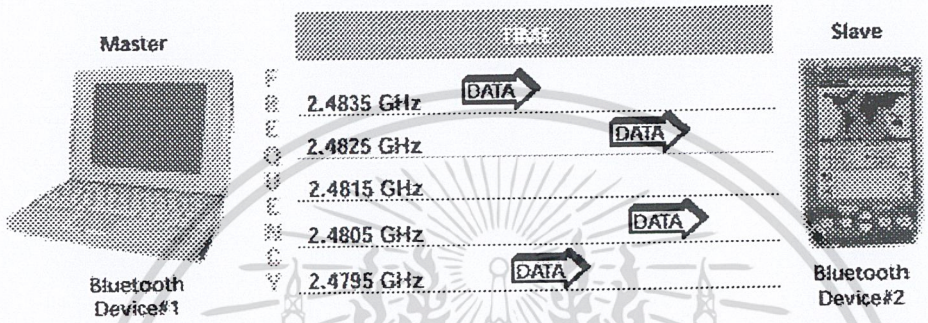
2.2.2 เทคนิค Frequency Hopping

บลูทูธใช้หลักการ Frequency Hop Spread Spectrum (FHSS) ในการส่งทั้งข้อมูลธรรมดาและข้อมูลที่เป็นเสียง หลักการนี้ได้นำมาใช้ในกองทัพเป็นเวลาหลายปี ด้วยเหตุผลที่ว่ามันช่วยเพิ่มความปลอดภัยในการส่งข้อมูล หลักการนี้จะใช้แพ็กเกตสวิทช์ชิ่งในการส่งข้อมูลจากตัวส่งในโมดูลบลูทูธไปยังตัวรับอีกตัวหนึ่ง ซึ่งต่างจากเซอร์กิตสวิทช์ชิ่งที่ใช้กำหนดช่องสัญญาณที่ใช้ในการสื่อสารที่แน่นอน ขั้นตอนเริ่มจากการแบ่งข้อมูลออกเป็นแพ็กเกตย่อยๆและส่งออกไปในช่วงความถี่ต่างๆกัน โดย 1 แพ็กเกตนี้จะมีคาบเท่ากับ 625 ไมโครวินาที ในการส่ง ตัวส่งจะส่งโดยการสวิทช์หรือฮ็อบภายในคลื่นความถี่ 2.4 กิกะเฮิร์ตซ (มีทั้งหมด 79 ฮ็อบ) ที่ความเร็ว 1600 ฮ็อบต่อวินาที (1 / 625 us) ถึงแม้ว่าเทคนิคนี้จะทำให้ประสิทธิภาพในการใช้แบนด์วิธต่ำกว่าวิธีอื่น แต่ข้อดีที่เหนือกว่า คือ ความมั่นคงและปลอดภัยกว่าวิธีอื่น

จากที่กล่าวมาข้างต้น ด้วยหลักการ FHSS ข้อมูลจะถูกแบ่งออกเป็นแพ็กเกตย่อยๆและส่งออกไปในความถี่ที่แตกต่างกันถึง 79 ช่อง แพ็กเกตหนึ่งเมื่อถูกส่งที่ความถี่หนึ่ง แพ็กเกตต่อมาก็จะถูกส่งที่ความถี่อื่น เป็นเช่นนี้เรื่อยๆที่อัตราความเร็วเท่ากับ 1600 ฮ็อบต่อวินาที รูปแบบของการส่งนี้ถูกกำหนดโดยตัว

อัลกอริทึม ซึ่งมาสเตอร์จะเป็นตัวบอกสเลฟเองว่ารูปแบบใดที่จะใช้ในการส่งข้อมูล สเลฟก็จะรู้ว่าจะรับข้อมูลได้ที่ช่องสัญญาณใด

ผลลัพธ์จากการใช้ FHSS คือ เพิ่มความปลอดภัยให้กับการส่งข้อมูล และลดปัญหาการรบกวนจากอุปกรณ์อื่นที่ใช้คลื่นความถี่เดียวกัน เช่น อุปกรณ์พวกไมโครเวฟ หรือระบบแลนไร้สาย เป็นต้น เทคนิคนี้จะช่วยลดปัญหาการรบกวนจากอุปกรณ์บลูทูธตัวอื่นที่ไม่ได้เข้าจังหวะกับความถี่ให้เหมาะสม ข้อมูลที่ส่งผ่านด้วยเทคนิคนี้เปรียบเหมือนกับสัญญาณรบกวนสำหรับอุปกรณ์อื่นที่ไม่ได้เข้าจังหวะกัน หรืออุปกรณ์ที่ไม่ได้รับอนุญาต



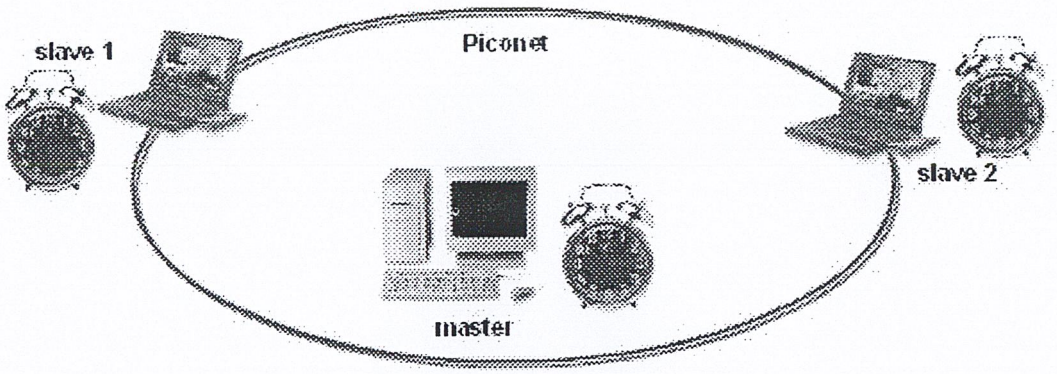
รูปที่ 2-2 การส่งข้อมูลแบบ FHSS

2.2.3 มาสเตอร์ สเลฟและ Frequency Hopping

อุปกรณ์ในบลูทูธจะทำงานใน 2 ลักษณะ คือ เป็นมาสเตอร์หรือสเลฟ มาสเตอร์จะเป็นตัวที่กำหนดค่าลำดับความถี่ที่ใช้ฮ็อบ (Frequency hop)ส่วนสเลฟจะเข้าจังหวะเวลาและความถี่กับลำดับความถี่ที่ใช้ฮ็อบของตัวมาสเตอร์

อุปกรณ์ที่ใช้เทคโนโลยีบลูทูธทุกตัวจะมีแอดเดรสของตัวอุปกรณ์ (Bluetooth Device Address) และบลูทูธคล็อก (Bluetooth Clock) ที่แตกต่างกัน โดยที่ส่วนของเบสแบนด์ (Baseband) ในรายละเอียดโครงสร้างของบลูทูธจะบอกถึงอัลกอริทึมที่อธิบายการคำนวณลำดับความถี่ที่ใช้ฮ็อบ จากแอดเดรสอุปกรณ์บลูทูธและคล็อกของตัวมาสเตอร์ เมื่อสเลฟทำการติดต่อไปยังตัวมาสเตอร์ มันจะได้รับแอดเดรสอุปกรณ์บลูทูธและ คล็อกของตัวมาสเตอร์ หลังจากนั้นมันจะนำค่าที่ได้นี้มาคำนวณเพื่อหาลำดับความถี่ที่ใช้ฮ็อบ เนื่องจากการที่ สเลฟทุกตัวใช้แอดเดรสของอุปกรณ์และคล็อกของตัวมาสเตอร์ ดังนั้นสเลฟทุกตัวจะเทียบกับลำดับความถี่ที่ใช้ ฮ็อบของมาสเตอร์ด้วย

มาสเตอร์จะเป็นตัวควบคุมการแบ่งใช้งานแบนด์วิธให้กับสเลฟแต่ละตัว โดยตัดสินใจว่าจะติดต่อกับสเลฟแต่ละตัวเมื่อไหร่และบ่อยแค่ไหน จำนวนเวลาของสล็อตที่แต่ละอุปกรณ์จะได้รับขึ้นอยู่กับความต้องการในการส่งข้อมูล ระบบที่แบ่งช่วงเวลาของสล็อตระหว่างอุปกรณ์หลายๆตัวนี้ เรียกว่า Time Division Multiplexing (TDM)

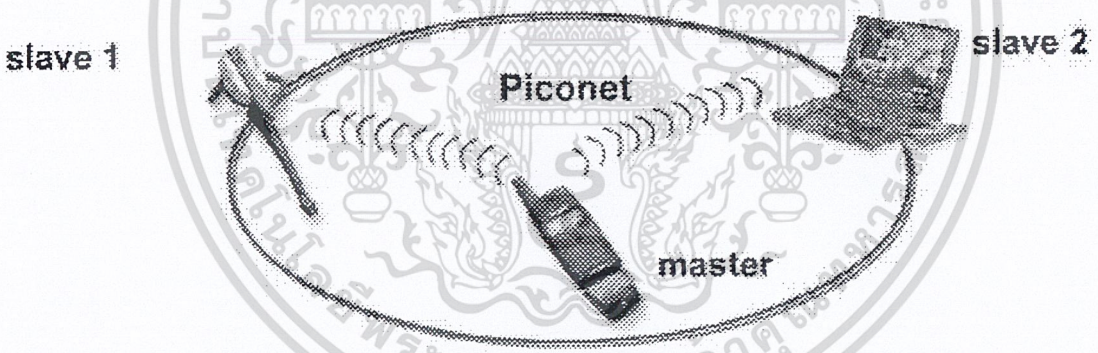


รูปที่ 2-3 Frequency Hopping ในเครือข่ายบลูทูธ

2.2.4 ลักษณะการต่อโครงข่าย

การต่อโครงข่ายเพื่อใช้งานบลูทูธมีอยู่ด้วยกัน 2 ลักษณะ คือ

1. พิโคเน็ต (Piconet) เป็นกลุ่มของสเลฟที่ทำงานร่วมกันภายใต้การควบคุมของมาสเตอร์ตัวเดียวกัน อุปกรณ์ทุกตัวภายในพิโคเน็ตจะตามลำดับความถี่ที่ใช้ฮ็อบของมาสเตอร์ การเชื่อมต่อในพิโคเน็ตจะมีได้ 2 แบบ คือ แบบจุดต่อจุด และแบบจุดต่อหลายจุด โดยที่สเลฟแต่ละตัวจะสามารถเชื่อมต่อกับมาสเตอร์เท่านั้น ไม่สามารถเชื่อมต่อโดยตรงระหว่างสเลฟกับสเลฟเองได้



รูปที่ 2-4 โครงข่ายพิโคเน็ต

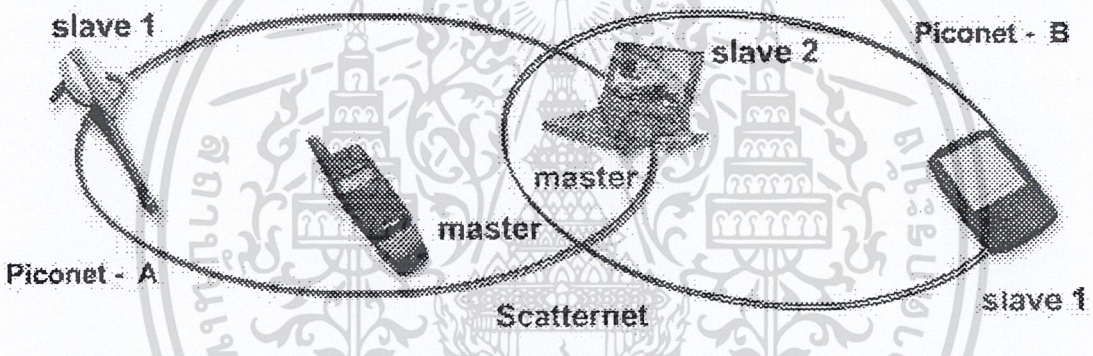
ช่องการสื่อสารข้อมูลจะถูกใช้ร่วมกันระหว่างอุปกรณ์ต่างๆเหล่านั้นภายในพิโคเน็ต และในขณะใดขณะหนึ่งสามารถที่จะมีอุปกรณ์ที่เป็น Slave Active ได้เพียง 7 ตัวเท่านั้น ทั้งนี้เพราะว่าบลูทูธได้กำหนดให้ใช้ข้อมูลที่มีชื่อว่า Active Member Address (AM_ADDR) ซึ่งประกอบไปด้วยข้อมูลจำนวน 3 บิต มีมาสเตอร์ 1 ตัว และนอกนั้นเป็นสเลฟ 7 ตัว โดยสเลฟทั้ง 7 ตัวนี้จะถูกอ้างอิงโดยตัวเลขไม่ซ้ำกัน ภายใน 3 บิต AM_ADDR สำหรับกรณีที่ตัวเลข 3 บิต นั้นเป็น 0 หมด คือ จองเอาไว้สำหรับใช้ในกรณีประกาศภายในพิโคเน็ต

ถ้าเกิดมีสเลฟมากกว่า 7 ตัวในพิโคเน็ต ก็สามารถมีได้ และยังสามารถทำให้เข้าจังหวะกันกับมาสเตอร์ได้ แต่มันจะไม่มี AM_ADDR เรียก สถานะอุปกรณ์ที่อยู่ในสถานะนี้ว่า สถานะ Parked ซึ่งอุปกรณ์ที่

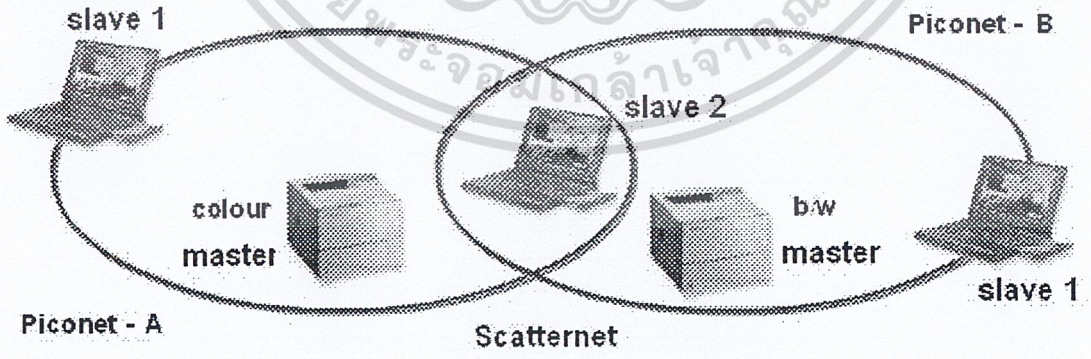
อยู่ในสถานะ Parked นี้จะมีแอดเดรสอีกตัวหนึ่ง ซึ่งมีชื่อเรียกว่า Parked Member Address (PM_ADDR) จะมีขนาด 8 บิต ซึ่งแสดงว่าสามารถมีอุปกรณ์ที่อยู่ในสถานะนี้ได้ถึง 256 ตัวเลยทีเดียว ซึ่งอุปกรณ์ที่อยู่ในสถานะนี้ ยังคงสามารถทำให้เข้าจังหวะกันกับมาสเตอร์ได้ แต่จะไม่สามารถติดต่อสื่อสารแบบแลกเปลี่ยนข้อมูลได้ภายในพิโคเน็ตได้ อุปกรณ์ที่อยู่ในสถานะ Parked นี้ก็สามารถกลับกลายเป็น สถานะ Active เพื่อทำการติดต่อสื่อสารกันในพิโคเน็ตได้ทันที (ถ้ามีที่ว่าง AM_ADDR เหลืออยู่)

เมื่ออุปกรณ์ตัวหนึ่งอยู่ในพิโคเน็ตมากกว่า 1 แห่ง จะต้องใช้ลักษณะของการแชร์เวลา โดยแบ่งสล็อตให้กับแต่ละพิโคเน็ต อุปกรณ์ตัวหนึ่งอาจเป็นสเลฟในพิโคเน็ตหนึ่งแต่เป็นมาสเตอร์ในอีกพิโคเน็ตหนึ่งก็ได้ แต่ไม่มีทางที่จะเป็นมาสเตอร์ทั้ง 2 พิโคเน็ตพร้อมๆกันได้ เนื่องจากเหตุผลที่อุปกรณ์ทุกตัวในพิโคเน็ตต้องเข้าจังหวะกับลำดับการฮ็อบของมาสเตอร์ ดังนั้น โดยนิยามแล้ว อุปกรณ์ทุกตัวที่อยู่ภายใต้มาสเตอร์ตัวเดียวกันจะต้องอยู่ในพิโคเน็ตเดียวกันด้วย

2. สแคตเตอร์เน็ต (Scatternet) จะเป็นลักษณะการเชื่อมโยงในกรณีที่มีพิโคเน็ตมากกว่า 1 พิโคเน็ตมาอยู่ในบริเวณเดียวกัน



รูปที่ 2-5 โครงข่ายสแคตเตอร์เน็ตในกรณีที่อุปกรณ์เป็นได้ทั้งมาสเตอร์และสเลฟ



รูปที่ 2-6 โครงข่ายสแคตเตอร์เน็ตในกรณีที่อุปกรณ์เป็นสเลฟทั้ง 2 โครงข่าย

ลักษณะการต่อเชื่อมโยงแบบนี้เรียกว่าสแคตเตอร์เน็ต ซึ่งสเลฟในพิโคเน็ตใดๆสามารถติดต่อไปยัง อุปกรณ์บลูทูธอื่นที่อยู่คนละพิโคเน็ตได้ ซึ่งอาจจะเป็นมาสเตอร์หรือสเลฟก็ได้ โดยใช้หลักการของ Time Division Multiplexing (TDM)

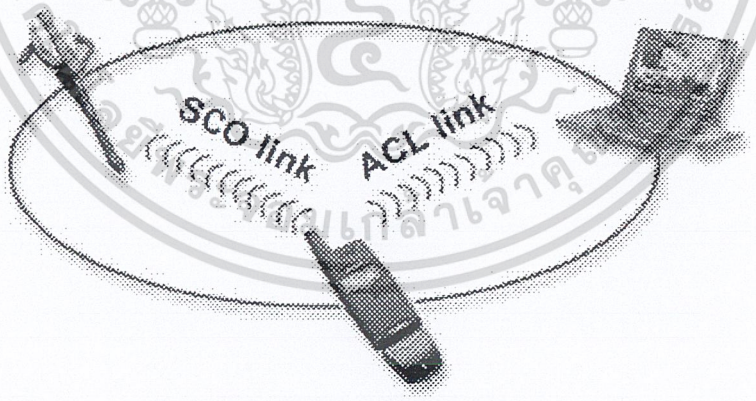
ภายในสแคตเตอร์เน็ตนี้พิโคเน็ตทั้งหลายที่อยู่ในสแคตเตอร์เน็ต จะไม่ทำการเข้าจังหวะข้อมูลกัน หมายถึง มันจะไม่รบกวนกัน เพราะว่าด้วยหลักการของ Frequency Hopping Channel นั่นเอง แต่อุปกรณ์ที่อยู่ภายในพิโคเน็ต สามารถติดต่อสื่อสารกับอุปกรณ์อื่นๆที่อยู่คนละพิโคเน็ตได้โดยอาศัยหลักการ TDM

2.2.5 ช่องทางการสื่อสารข้อมูล

ช่องทางการสื่อสารของบลูทูธนั้นมีแบนด์วิธทั้งหมด 1 เมกะบิตต่อวินาที แบ่งออกเป็น 2 ประเภท คือ

1. SCO (Synchronous Connection Oriented) สำหรับการสื่อสารข้อมูลในรูปแบบของเสียง รองรับการเชื่อมต่อแบบสมมาตร เซอร์กิตสวิทช์ และการเชื่อมต่อแบบจุดต่อจุด เมื่อมีการรับส่งข้อมูลจะสงวนช่องทางนั้นไว้ ความเร็วในการส่งข้อมูลอยู่ที่ 64 กิโลบิตต่อวินาที สามารถเชื่อมต่อได้ 3 ช่องสัญญาณพร้อมกัน

2. ACL (Asynchronous Connectionless) สำหรับการสื่อสารข้อมูลทั่วไป รองรับการเชื่อมต่อทั้งแบบสมมาตรและไม่สมมาตร แพ็กเกตสวิทช์ และการเชื่อมต่อแบบจุดต่อหลายจุด ในการเชื่อมต่อแบบสมมาตรนั้นมีความเร็วในการรับส่งข้อมูลอยู่ที่ 433.9 กิโลบิตต่อวินาที ส่วนการเชื่อมต่อแบบไม่สมมาตรนั้นมีความเร็วอยู่ที่ 723.2 กิโลบิตต่อวินาที และ 57.6 กิโลบิตต่อวินาที ในอีกทิศทางหนึ่ง



รูปที่ 2-7 ช่องทางการสื่อสารข้อมูล

ลักษณะการใช้งาน	Upstreme Data Rate สูงสุด	Downstreme Data Rate สูงสุด
3 Simultaneous Voice Channels	64 kb/sec X 3 channels	64 kb/sec X 3 channels

ลักษณะการใช้งาน	Upstreme Data Rate สูงสุด	Downstreme Data Rate สูงสุด
Symmetric Data	433.9 kb/sec	433.9 kb/sec
Asymmetric Data	723.2 kb/sec or 57.6 kb/sec	57.6 kb/sec or 723.2 kb/sec

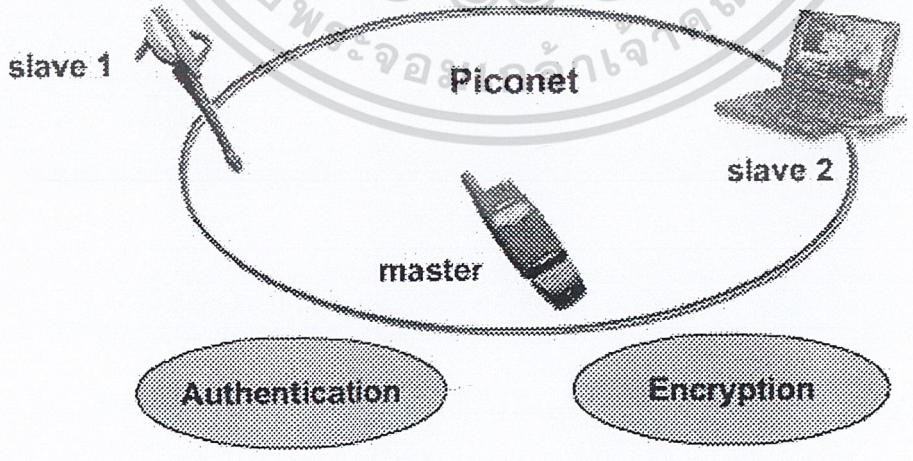
ตารางที่ 2-2 ความเร็วในการรับส่งข้อมูลในลักษณะต่างๆ

2.2.6 ระบบความปลอดภัย

เนื่องจากบลูทูธเป็นเทคโนโลยีที่รับส่งข้อมูลแบบไร้สาย ดังนั้นความปลอดภัยของข้อมูลเป็นสิ่งสำคัญที่ต้องคำนึงถึงเพื่อป้องกันการดักจับข้อมูลและการเปลี่ยนแปลงแก้ไขข้อมูล การพิสูจน์สิทธ์และการเข้ารหัสเป็นสิ่งที่เพิ่มเข้ามาในตัวบลูทูธ การพิสูจน์สิทธ์ใช้จำกัดสิทธ์ในการเข้าถึงข้อมูลเพื่อป้องกันการแก้ไขข้อมูลในส่วนที่สำคัญ ส่วนการเข้ารหัสข้อมูลใช้เพื่อป้องกันการดักจับข้อมูล ทั้งสองเทคนิคนี้ใช้รวมเข้ากับการทำ Frequency hopping ในการส่งข้อมูล ระบบความปลอดภัยในบลูทูธแบ่งออกเป็น 3 ระดับ คือ

- โหมด 1: ไม่มีการรักษาความปลอดภัยใดๆ
- โหมด 2: การรักษาความปลอดภัยขึ้นอยู่กับระดับของการบริการ
- โหมด 3: การรักษาความปลอดภัยขึ้นอยู่กับระดับของการเชื่อมต่อ

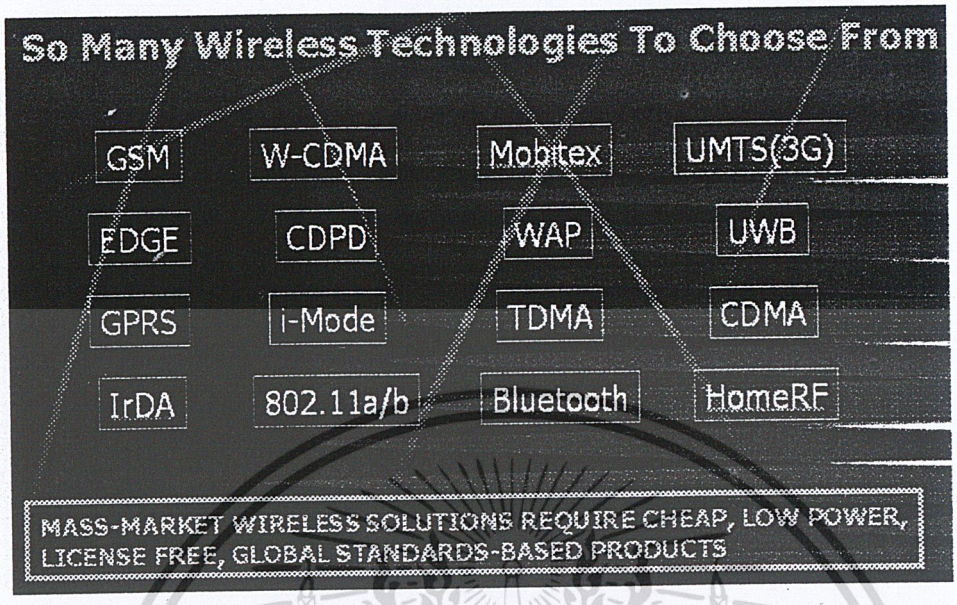
ข้อแตกต่างระหว่างโหมด 2 และโหมด 3 คือ โหมด 2 อุปกรณ์บลูทูธจะเริ่มต้นกระบวนการรักษาความปลอดภัยที่เลเยอร์ระดับสูง ในขณะที่โหมด 3 จะทำที่เลเยอร์ระดับต่ำ



รูปที่ 2-8 การรักษาความปลอดภัยในบลูทูธ

2.3 เทคโนโลยีไร้สายอื่น

เทคโนโลยีไร้สายที่มีอยู่ในปัจจุบันมีอยู่เป็นจำนวนมาก ดังรูปที่ 2-9



รูปที่ 2-9 เทคโนโลยีไร้สายที่น่าสนใจ

ในที่นี้จะกล่าวถึงเทคโนโลยีไร้สายที่ใช้งานในระยะใกล้ที่เหมือนกับบลูทูธ ซึ่งมีอยู่ด้วยกัน 3 ตัว คือ IrDA, HomeRF และIEEE802.11b (Wireless LAN)

2.3.1 IrDA

IrDA (INFRARED DATA ASSOCIATION) เป็นมาตรฐานการเชื่อมต่อไร้สายที่ใช้รังสีอินฟราเรดในการส่งถ่ายข้อมูล อัตราการส่งผ่านข้อมูล 4 เมกะบิตต่อวินาที และระยะในการรับส่งอยู่ที่ 5 เมตร มีการนำมาใช้ในอุปกรณ์ต่างๆเป็นจำนวนมาก ลักษณะการใช้งานต้องเชื่อมต่อเป็นแบบเส้นตรงและรองรับการสื่อสารเฉพาะแบบจุดต่อจุดเท่านั้น ซึ่งกลายมาเป็นข้อจำกัดในการใช้งาน ปัญหาอีกอย่างของ IrDA คือ ไม่มีมาตรฐานกลางที่มารองรับ ทำให้อุปกรณ์ต่างชนิดกันไม่สามารถติดต่อสื่อสารระหว่างกันได้

2.3.2 HomeRF

HRFWG (HomeRF Working Group) เริ่มก่อตั้งในปีเดือนมีนาคม ปี 1998 เพื่อสร้างมาตรฐานเปิดสำหรับกลุ่มอุตสาหกรรมในการสื่อสารไร้สายระหว่างคอมพิวเตอร์กับอุปกรณ์อิเล็กทรอนิกส์ทั่วๆไป รองรับการทำงานในลักษณะเครือข่ายไร้สายภายในบ้าน ใช้คลื่นความถี่วิทยุ 2.4 กิกะเฮิร์ตซ ในการรับส่งข้อมูลที่อัตราความเร็ว 1.6 เมกะบิตต่อวินาที โดยใช้เทคนิค SWAP (Shared Wireless Access Protocol) ซึ่งจะทำให้รองรับอุปกรณ์ได้ถึง 127 ตัวในเครือข่ายย่อย ข้อจำกัด คือ มาตรฐานนี้ไม่ได้เป็นของฟรี ต้องเสียค่าธรรมเนียมสำหรับการเป็นสมาชิก และเทคนิค SWAP ที่ใช้ใน HomeRF นี้จะใช้ลำดับความถี่ในการฮ็อบ

เท่ากับ 50 ฮ็อบต่อวินาที ซึ่งน้อยมากเมื่อเทียบกับ บลูทูธ ซึ่งลำดับความถี่ในการฮ็อบจะมีผลต่อความปลอดภัยในการส่งข้อมูล

2.3.3 IEEE 802.11b (Wireless LAN)

IEEE 802.11b เกิดจากกลุ่มการทำงาน 802.11 ของมาตรฐาน IEEE ในสหรัฐอเมริกา ซึ่งมีหน้าที่หลักในการกำหนดและแก้ไขปรับปรุงรายละเอียดมาตรฐานระบบเครือข่ายไร้สายแบบท้องถิ่น (WLANs) มาตรฐาน IEEE 802.11b นี้ทำงานบนคลื่นความถี่วิทยุ 2.4 กิกะเฮิร์ตซ ใช้เทคนิค Direct-Sequence Spread Spectrumความเร็วในการส่งข้อมูลอยู่ที่ 11 เมกะบิตต่อวินาที ภายในระยะ 150 เมตร กินพลังงานมาก ค่าใช้จ่ายสูง ทำให้ระบบนี้เหมาะที่จะทำงานในลักษณะขององค์กรธุรกิจ เช่น ออฟฟิศ หรือสำนักงาน เป็นต้น

	Bluetooth	Wireless LAN	Home RF	Infrared
Frequency:	2.4 GHz	2.4 GHz	2.4 GHz	Ir Band
Data rate:	1Mbps	11Mbps	1.6 Mbps	4 Mbps
Range:	32 feet (10m)	Up to 500 feet (150m)	Up to 150 feet (45m)	Up to 15 feet (5m)
Standard:	Bluetooth	802.11b	SWAP (Shared Wireless Access Protocol)	IrDA
Communication Technique	FHSS	DSSS	FHSS	(NA)
Usage Model	Cable Replacement	High-Speed Network Access	Low-Speed, Low-Cost Voice & Data Home Networking	Point-to-Point Data Transmission

ตารางที่ 2-3 เปรียบเทียบเทคโนโลยีไร้สายที่ใช้ในระยะสั้น

2.3.4 ข้อได้เปรียบจากการใช้เทคโนโลยีบลูทูธ

- ความสะดวกสบายในการใช้งานและข้อมูลที่ถูกต้องตรงกัน : บลูทูธจะช่วยให้ข้อมูลที่มีอยู่ในอุปกรณ์ต่างๆภายในพิโคเน็ตเดียวกันตรงกันโดยไม่ต้องพึ่งพาสายเคเบิล สิ่งที่อุปกรณ์ต้องทำเพียงแค่อยู่ในระยะที่ใกล้พอที่ข้อมูลจะสามารถส่งผ่านได้ภายในรัศมีของบลูทูธ
- ความเข้ากันได้ของอุปกรณ์ต่างๆ : บลูทูธเป็นมาตรฐานที่กำหนดพื้นฐานการติดต่อสื่อสารที่ใช้ในตัวโมดูลบลูทูธ ดังนั้นอุปกรณ์บลูทูธทุกตัวจะสามารถที่จะติดต่อสื่อสารกับอุปกรณ์บลูทูธอื่นๆได้โดยไม่ต้องคำนึงถึงแหล่งที่ผลิต
- ต้นทุนต่ำสำหรับผู้ใช้งาน : บลูทูธช่วยลดต้นทุนสำหรับผู้ใช้งานให้ต่ำลงในเรื่องของค่าสายเคเบิล รวมไปถึงค่าบำรุงรักษาในการเปลี่ยนหรือซ่อมแซมสายเคเบิล และยังเพิ่มโอกาสในการแลกเปลี่ยนข้อมูลข่าวสารในกรณีที่ลืมสายเคเบิลหรือสายเคเบิลหาย

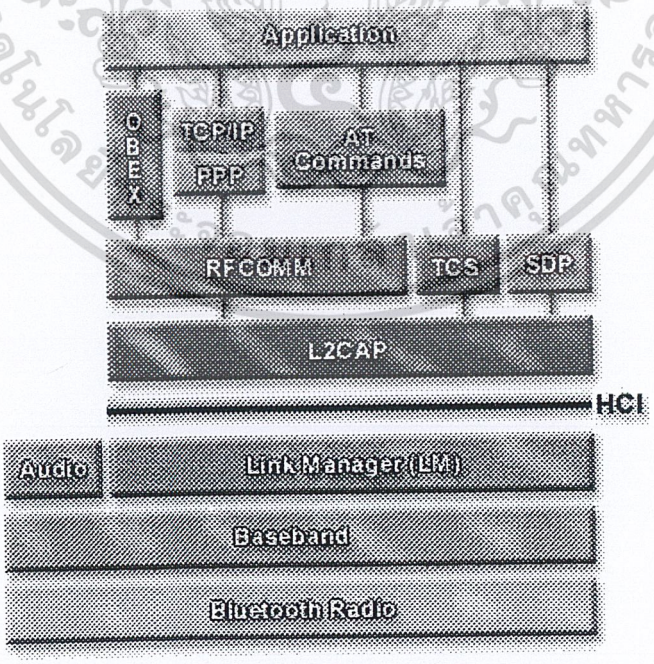
- การรับส่งข้อมูลที่ปลอดภัยจากการรบกวน : บลทูธถูกออกแบบมาเพื่อให้มันคงทนอยู่ได้จากการรบกวนของอุปกรณ์อื่นๆที่ทำงานอยู่ภายใต้คลื่นความถี่วิทยุ 2.4 กิกะเฮิร์ตซ เช่น ระบบแลนไร้สาย ไมโครเวฟ เป็นต้น ด้วยการใช้เทคนิค Frequency hopping จะช่วยลดการชนกันของแพ็กเกตในการรับส่งข้อมูลและช่วยลดความเสียหายที่เกิดจากการรบกวน

2.4 บลูทูธโพรโตคอลสแต็ก

จุดประสงค์หลักของบลูทูธคือการทำให้อุปกรณ์ต่างๆสามารถติดต่อสื่อสารกันได้ โดยไม่ต้องคำนึงแหล่งที่มาของผู้ผลิต เพื่อที่จะบรรลุวัตถุประสงค์นี้ บลูทูธไม่ได้เลือกที่จะกำหนดระบบคลื่นวิทยุขึ้นมาใหม่ แต่เลือกที่จะกำหนดซอฟต์แวร์สแต็กขึ้นมาแทน เพื่อให้โปรแกรมประยุกต์ต่างๆสามารถค้นหาอุปกรณ์บลูทูธตัวอื่นๆที่อยู่ในระยะเครือข่าย ค้นหาการบริการที่มีอยู่และทำการเรียกใช้งานการบริการเหล่านั้นได้

การที่จะเข้าใจการทำงานภายในตัวบลูทูธนั้น ต้องเข้าใจโครงสร้างของมันก่อน ซึ่งโครงสร้างภายในบลูทูธดังที่กล่าวมาแล้วว่าเป็นลักษณะของโพรโตคอลสแต็ก ซึ่งเป็นลักษณะของชั้นต่างๆซ้อนทับกันอยู่ สามารถแบ่งออกเป็น 4 ส่วนหลักๆคือ

- โพรโตคอลหลักของบลูทูธ
- โพรโตคอลแทนที่การใช้สายเคเบิล
- โพรโตคอลควบคุมการใช้งานในระบบโทรศัพท์
- โพรโตคอลที่รับเอามาใช้



รูปที่ 2-10 บลูทูธโพรโตคอลสแต็ก

2.4.1 โพรโตคอลหลักของบลูทูธ

- **Baseband** : เลเยอร์ส่วนนี้เป็นการเชื่อมต่อทางกายภาพของคลื่นความถี่วิทยุระหว่างอุปกรณ์บลูทูธในพิโคเน็ต เนื่องจากระบบคลื่นวิทยุที่บลูทูธใช้เป็นการกระโดดของคลื่นความถี่ซึ่งแพกเกตต่างๆจะถูกส่งโดยการกำหนดของไทม์สลอตบนความถี่ที่กำหนด ดังนั้นเลเยอร์นี้จะใช้กระบวนการอินไดวรี่และเพจจิงเพื่อควบคุมการเข้าจังหวะการรับส่งบนความถี่ที่กระโดดและคลอกของอุปกรณ์บลูทูธแต่ละชนิด

ช่องทางการสื่อสารของบลูทูธแบ่งเป็น 2 แบบ คือ SCO (Synchronous Connection Oriented) และ ACL (Asynchronous Connectionless) สำหรับแพกเกต ACL จะใช้ในการส่งข้อมูลทั่วไปเท่านั้น แต่สำหรับแพกเกต SCO สามารถประกอบไปด้วยข้อมูลที่เป็นเสียงอย่างเดียว หรือรวมกันระหว่างข้อมูลธรรมดาและข้อมูลที่เป็นเสียงก็ได้ ข้อมูลทุกอย่างไม่ว่าจะเป็นข้อมูลเสียงหรือข้อมูลธรรมดาสามารถที่จะส่งไปในหลายระดับของ FEC หรือ การแก้ข้อผิดพลาด CRC และสามารถเข้ารหัสข้อมูลด้วยได้

- **Audio** : การส่งผ่านสัญญาณเสียงสามารถทำได้ระหว่างอุปกรณ์บลูทูธหลายๆตัวโดยใช้โมเดลในการใช้งานที่แตกต่างกันได้ ข้อมูลสัญญาณเสียงไม่ได้ถูกส่งผ่านไปยังเลเยอร์ L2CAP แต่จะส่งผ่านโดยตรงระหว่างอุปกรณ์บลูทูธที่ทำการเชื่อมต่อ

- **HCI** (Host Controller Interface) : กำหนดรูปแบบวิธีการการเชื่อมต่อเพื่อเข้าถึงตัวอุปกรณ์บลูทูธ รวมทั้งรูปแบบคำสั่งที่ใช้ควบคุม Baseband และ Link Manager ในกรณีที่โพรโตคอลสแต็กบลูทูธทั้งหมดทำงานอยู่บนไมโครโพรเซสเซอร์เดียวกัน อาจไม่มีความจำเป็นต้องใช้เลเยอร์นี้

- **LMP** (Link Manager Protocol) : มีหน้าที่ในการเริ่มต้นการเชื่อมต่อระหว่างอุปกรณ์บลูทูธ การควบคุมและการติดต่อสื่อสารของแพ็กเกตที่ใช้ในการรับส่งข้อมูล ควบคุมโหมดของการใช้พลังงานและสถานะของอุปกรณ์บลูทูธในพิโคเน็ต รวมไปถึงเรื่องของความปลอดภัย เช่น การพิสูจน์สิทธิ์และการเข้ารหัสโดยการสร้าง, แลกเปลี่ยนและควบคุมการเชื่อมต่อของคีย์เข้ารหัสที่ใช้ในการพิสูจน์สิทธิ์และการเข้ารหัสข้อมูล

นอกจากนี้ยังดูแลในเรื่องของโหมดการใช้พลังงาน, ช่วงเวลาการใช้คลื่นวิทยุของอุปกรณ์บลูทูธและสถานะการเชื่อมต่อของอุปกรณ์บลูทูธในพิโคเน็ตด้วย

- **L2CAP** (Logical Link Control and Adaptation Protocol) : ให้การบริการข้อมูลที่เป็นการเชื่อมต่อแบบ Connection-oriented และ Connectionless ให้กับเลเยอร์ที่อยู่ชั้นบน โดยมีหน้าที่หลักอยู่ 4 อย่าง คือ

  - Multiplexing รองรับโพรโตคอลที่ใช้การ Multiplexing เช่น SDP, RFCOMM
  - การแบ่งและการรวมเข้าด้วยกันของแพ็กเกต
  - ควบคุมความถูกต้องในการให้บริการ
  - การรวมกันเป็นกลุ่ม โดยจะอนุญาตให้มีการสร้างกลุ่มขึ้นมาในพิโคเน็ต

- **SDP** (Service Discovery Protocol) : กำหนดวิธีการที่อุปกรณ์บลูทูธจะทำการค้นหารูปแบบการบริการที่มีอยู่ โดยที่ตัวที่ร้องขอไม่จำเป็นต้องรู้มาก่อนว่ามีการบริการอะไรให้ใช้บ้าง รวมทั้งค้นหาการบริการรูปแบบใหม่เมื่อมีอุปกรณ์บลูทูธเข้ามาในเครือข่าย และเมื่ออุปกรณ์นั้นออกจากเครือข่าย

2.4.2 โพรโตคอลแทนที่การใช้สายเคเบิล

- **RFCOMM** : เป็นโพรโตคอลที่จำลองการทำงานของพอร์ตอนุกรม RS-232 การควบคุมและการส่งสัญญาณผ่านบลูทูธ Baseband ซึ่งอิงมาตรฐาน ETSI 07.10

2.4.3 โพรโตคอลควบคุมการใช้งานในระบบโทรศัพท์

- **TCS BIN** (Telephony Control – Binary) : กำหนดการควบคุมการรับส่งสัญญาณเพื่อเริ่มต้นการติดต่อสื่อสารข้อมูลธรรมดาและข้อมูลเสียงระหว่างอุปกรณ์บลูทูธ ทั้งการเชื่อมต่อแบบจุดต่อจุดและการเชื่อมต่อแบบจุดต่อหลายจุดด้วย TCS BINนี้ใช้มาตรฐาน ITU-T Recommendation Q.931
- **Telephony Control** – AT commands : รองรับรูปแบบคำสั่งของ AT commands ในการส่งสัญญาณควบคุมสำหรับการใช้งานในระบบโทรศัพท์ ใช้ RFCOMM ในการส่งข้อมูล

2.4.4 โพรโตคอลที่รับเอามาใช้

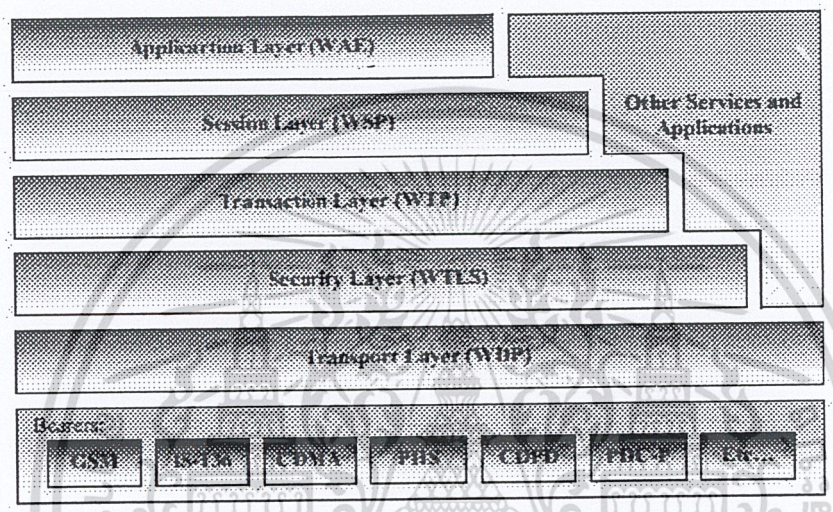
- **PPP** (IETF Point-t-Point Protocol) : นำมาใช้ทำงานอยู่บน RFCOMM ในการเชื่อมต่อแบบจุดต่อจุด PPP เป็นโพรโตคอลลักษณะของแพ็กเกตที่ใช้การเชื่อมต่อแบบอนุกรมโดยการแปลงข้อมูลที่อยู่ในรูปของกระแสข้อมูลแพ็กเกตให้เป็นกระแสข้อมูลแบบอนุกรม
- **TCP/UDP/IP** : เป็นโพรโตคอลมาตรฐานที่ถูกกำหนดขึ้นโดย IETF (Internet Engineering Task Force) และใช้ในการติดต่อสื่อสารในอินเทอร์เน็ต ปัจจุบันเป็นโพรโตคอลที่มีการใช้งานกันอย่างแพร่หลายมากทั่วโลก สแต็กของ TCP/IP ปรากฏอยู่บนอุปกรณ์หลากหลายชนิด รวมไปถึงพรินเตอร์, คอมพิวเตอร์พกพา หรืออุปกรณ์พกพาต่างๆด้วย การใช้งานโพรโตคอลนี้ขึ้นกับระบบปฏิบัติการที่ใช้งานอยู่ถึงแม้ว่าโดยทั่วไปแล้วจะเขียนโปรแกรมผ่านทางส่วนเชื่อมต่อแบบซ็อกเกต การนำโพรโตคอลนี้มาใช้ในบลูทูธเพื่อให้อุปกรณ์บลูทูธสามารถที่จะติดต่อกับอุปกรณ์บลูทูธตัวอื่นๆเพื่อที่จะติดต่อไปใช้อินเทอร์เน็ตได้
- **OBEX Protocol** : เป็นโพรโตคอลที่ใช้ควบคุมการเชื่อมต่อถูกพัฒนาโดย IrDA (Infrared Data Association) เพื่อที่จะแลกเปลี่ยนออบเจ็กต์ซึ่งกันและกัน โพรโตคอล OBEX มีการเตรียมฟังก์ชันพื้นฐานที่จำเป็นในการใช้งานเหมือนเช่นโพรโตคอล HTTP, ใช้โมเดลของไคลเอ็นต์-เซิร์ฟเวอร์และเป็นอิสระจากวิธีที่ใช้ในการรับ-ส่งข้อมูลและ API ที่ใช้ในการส่งข้อมูล ภายในโพรโตคอลจะมีแกรมมาของ OBEX เองที่ใช้ในการติดต่อสื่อสารระหว่างอุปกรณ์ โพรโตคอล OBEX ยังจัดเตรียมโมเดลที่ใช้ในการแทนออบเจ็กต์และการทำงานต่างๆไว้ นอกจากนี้ยังกำหนดรูปแบบของการแสดงลำดับโฟวเดอร์ ซึ่งใช้ในการเรียกดูข้อมูลของโฟวเดอร์ในอุปกรณ์ระยะไกล โดยการส่งผ่านข้อมูลผ่านทาง RFCOMM

ในการใช้งานโพรโตคอล OBEX นี้ เราอาจนำมาใช้ร่วมกับ vCard และ vCalendar ด้วย ซึ่งทั้งสองตัวนี้เป็นมาตรฐานเปิดที่ถูกพัฒนาขึ้นมา ปัจจุบันอยู่ในการควบคุมของ Internet Mail Consortium รายละเอียดของมาตรฐานทั้งสองนี้จะกำหนดรูปแบบของนามบัตรอิเล็กทรอนิกส์และรายละเอียดการนัดหมายต่างๆที่อยู่ในปฏิทินส่วนบุคคล มาตรฐาน vCard และ vCalendar นี้ไม่ได้กำหนดวิธีในการรับส่งข้อมูล เพียงแต่กำหนดรูแบบข้อมูลที่จะส่ง ในการพัฒนา vCard และ vCalendar นั้น

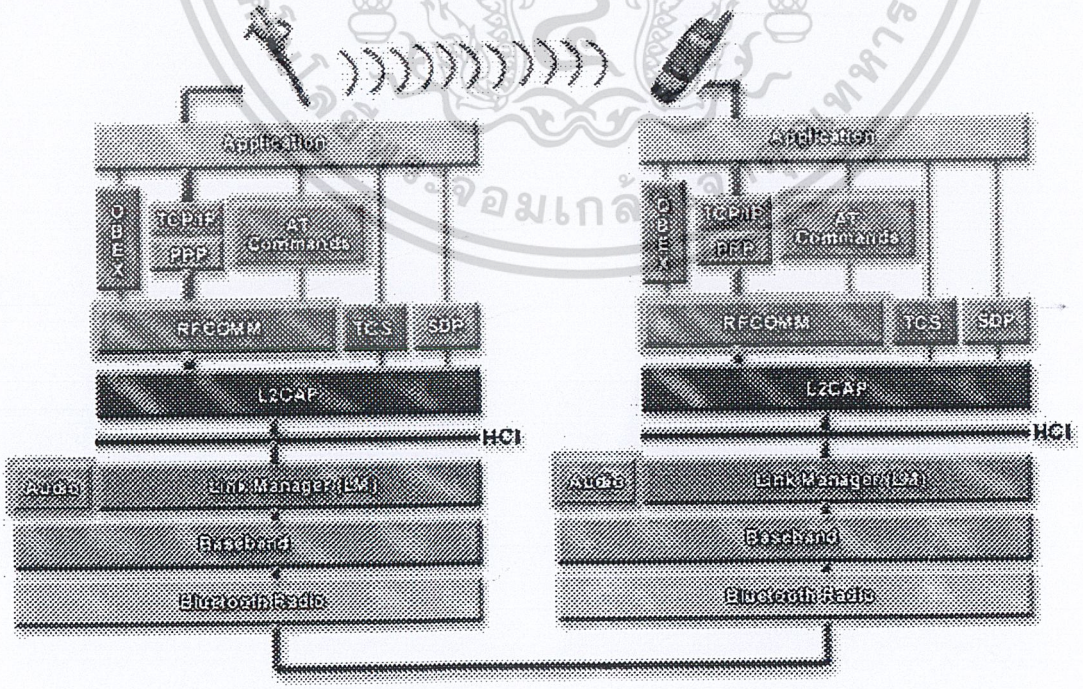
สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

สำหรับรูปแบบของข้อมูลอื่นซึ่งจะนำมาใช้ส่งผ่านโพรโตคอล OBEX บลูทูธนั้นมี vMessage และ vNote เป็นมาตรฐานเปิดเช่นเดียวกันใช้ในการแลกเปลี่ยนข้อความและโน้ต รายละเอียดกำหนดไว้ใน IrMC ซึ่งจะกำหนดรูปแบบของข้อมูลที่จำเป็นเมื่ออุปกรณ์ต่างๆทำการเข้าจังหวะกันเพื่อทำข้อมูลต่างๆให้ตรงกัน

- **WAP** (Wireless Application Protocol) : เป็นโพรโตคอลไร้สายที่ถูกสร้างขึ้นมาเพื่อให้ใช้ได้กับเทคโนโลยีเครือข่ายไร้สายตัวอื่นๆ โดยมีวัตถุประสงค์เพื่อนำเอาข้อมูลต่างๆบนอินเทอร์เน็ตและการบริการของระบบโทรศัพท์ส่งไปยังโทรศัพท์มือถือต่างๆและเทอร์มินอลไร้สายตัวอื่นๆ



รูปที่ 2-11 โครงร่างของโพรโตคอล WAP



รูปที่ 2-12 การใช้งานอุปกรณ์บลูทูธผ่านโพรโตคอลสแต็ก

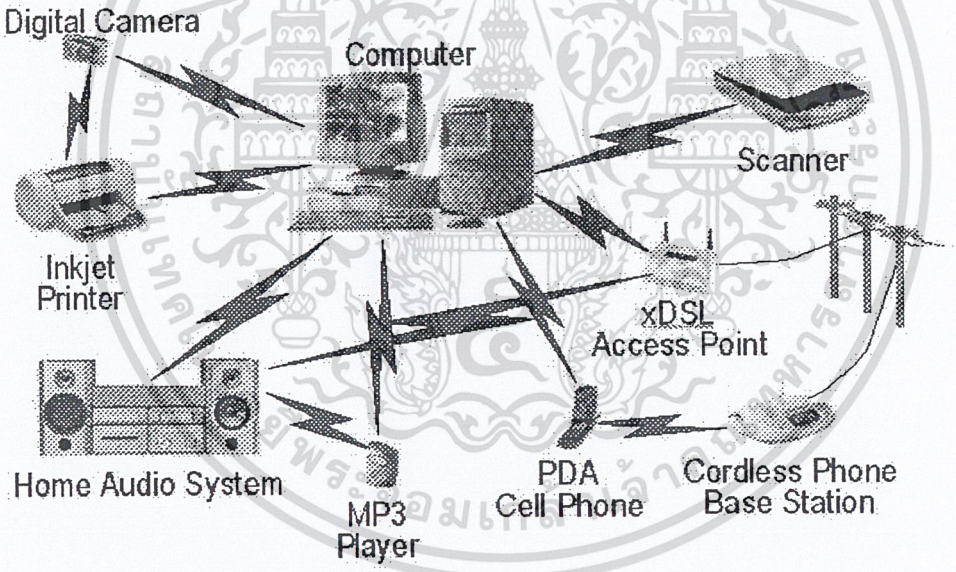
46189

2.5 การนำบลูทูธไปใช้งาน

เนื่องจากเทคโนโลยีไร้สายที่มีอยู่ในปัจจุบันมีอยู่เป็นจำนวนมาก แต่ละชนิดก็มีความเหมาะสมกับลักษณะการใช้งานที่แตกต่างกันออกไป ในหัวข้อนี้จะกล่าวถึงบทบาทของบลูทูธที่มีในระบบไร้สาย เมื่อทราบบทบาทของบลูทูธแล้ว หลังจากนั้นจะมาดูถึงลักษณะงานต่างๆที่เหมาะจะนำบลูทูธไปใช้งาน และดูถึงบลูทูธโพรไฟล์ที่กำหนดรายละเอียดการใช้งานบลูทูธในลักษณะงานด้านต่างๆ

2.5.1 สถานะของบลูทูธในระบบไร้สาย

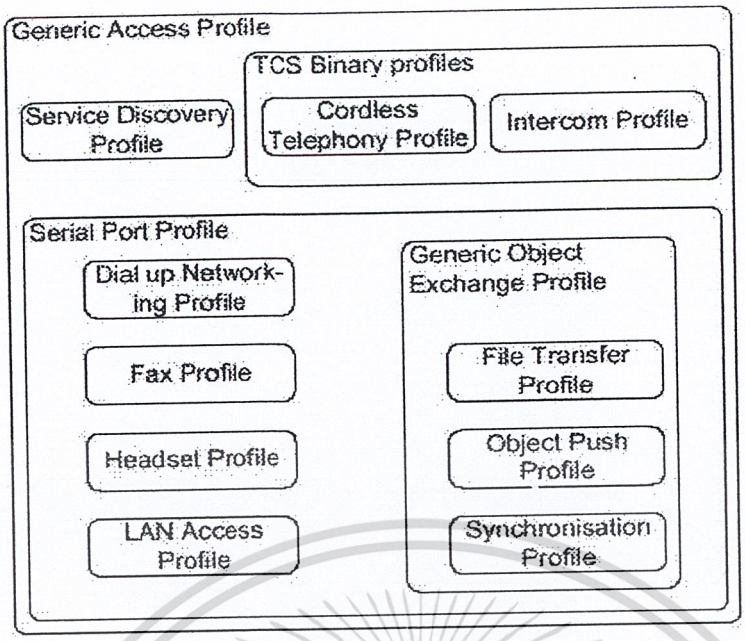
เทคโนโลยีไร้สายที่มีอยู่ในปัจจุบันมีอยู่มากมายหลายชนิด แต่ละเทคโนโลยีถูกสร้างขึ้นมาเพื่อรองรับการใช้งานที่แตกต่างกันไป สำหรับบลูทูธเองเหมาะที่จะนำไปใช้ในลักษณะของเครือข่ายส่วนบุคคล (PAN : Personal Area Network) เนื่องจากบลูทูธมีรัศมีการทำงานอยู่ที่ 10 เมตร (อาจขยายได้ถ้าเพิ่มกำลังส่ง) ดังนั้นอุปกรณ์ที่ใช้ก็จะเป็นลักษณะของอุปกรณ์ที่อยู่รอบๆตัว โดยเฉพาะในกลุ่มของอุปกรณ์พกพาต่างๆ เช่น โทรศัพท์มือถือ, เครื่องคอมพิวเตอร์แบบพกพา, โน้ตบุ๊ก เป็นต้น ซึ่งบลูทูธจะมาช่วยให้อุปกรณ์ต่างๆเหล่านี้สามารถติดต่อสื่อสาร, รับส่งข้อมูลระหว่างกันได้โดยที่ไม่ต้องคำนึงถึงแหล่งผลิต



รูปที่ 2-13 ภาพการเชื่อมต่ออุปกรณ์ต่างๆผ่านบลูทูธ

2.5.2 บลูทูธโพรไฟล์

ในการนำบลูทูธไปใช้งานในรูปแบบต่างๆนั้น ได้มีการกำหนดรายละเอียดการปรับแต่งค่า, โหมดการทำงาน หรือโพรโตคอลต่างๆที่ต้องนำมาใช้เพิ่มกับตัวบลูทูธเพื่อประโยชน์ในการใช้งานด้านนั้นๆได้อย่างมีประสิทธิภาพ ซึ่งรายละเอียดที่กล่าวมานี้จะอยู่ในส่วนของบลูทูธโพรไฟล์ โพรไฟล์หลักๆที่มีอยู่ในบลูทูธเวอร์ชัน 1.0 แบ่งออกเป็น 4 ส่วนหลักๆ คือ



รูปที่ 2-14 บลูทูธโพรไฟล์

โพรไฟล์ทั่วไปในการใช้งาน (GAP : Generic Access Profile)

เป็นโพรไฟล์พื้นฐานที่สำคัญที่สุดในบลูทูธโพรไฟล์ โพรไฟล์อื่นๆที่เหลือจะถูกสร้างขึ้นบน GAP และใช้ประโยชน์จากโพรไฟล์นี้ วัตถุประสงค์ของ GAP เพื่อที่จะรับประกันว่าอุปกรณ์บลูทูธทุกตัวสามารถที่จะติดต่อสื่อสารและสร้างการเชื่อมต่อขึ้นมาได้ ดังนั้น GAP จึงกำหนด

- คุณสมบัติต่างๆที่จำเป็นต้องมีในการอุปกรณ์บลูทูธทุกตัว
- กระบวนการพื้นฐานในการค้นหาอุปกรณ์
- การจัดการการเชื่อมต่อในการติดต่ออุปกรณ์บลูทูธ
- กระบวนการที่เกี่ยวข้องกับความปลอดภัยในแต่ละระดับ
- รูปแบบพื้นฐานที่จำเป็นในการเรียกใช้งานอุปกรณ์ต่างๆในระดับของการติดต่อกับผู้ใช้

- การใช้ถ้อยคำ : GAP กำหนดการใช้ถ้อยคำที่จะถูกใช้ในการติดต่อกับผู้ใช้ โดยการกำหนดการใช้ถ้อยคำพื้นฐานเหล่านี้
  - คำที่เกี่ยวกับการเชื่อมต่อ เช่น แชนเนล, ลิงค์ เป็นต้น
  - คำที่เกี่ยวกับอุปกรณ์ เช่น อุปกรณ์ที่เชื่อใจได้ เป็นต้น
  - คำที่เกี่ยวกับกระบวนการ เช่น การค้นหาการบริการ, การค้นหาชื่อ เป็นต้น
  - คำที่เกี่ยวข้องกับความปลอดภัย เช่น การเชื่อใจ, การพิสูจน์สิทธิ์ เป็นต้น
- โหมดการทำงาน : GAP อธิบายโหมดการทำงานของอุปกรณ์บลูทูธ กำหนดว่าสิ่งใดที่จำเป็นต้องมีและสิ่งใดที่เป็นตัวเสริมการทำงาน การทำงานที่ถูกอธิบาย คือ
  - ความสามารถในการค้นหา : โดยการใช้กระบวนการอินไควรี่สแกน ทำให้อุปกรณ์อื่นๆสามารถที่จะค้นหาอุปกรณ์บลูทูธเมื่อเข้ามาภายในระยะเครือข่ายได้

- ความสามารถในการติดต่อ : โดยการใช้กระบวนการเพจสแกน ทำให้อุปกรณ์อื่นๆสามารถติดต่อกับอุปกรณ์บลูทูธได้เมื่อเข้ามาภายในระยะเครือข่าย
- ความสามารถในการเปรียบเทียบ : โดยการใช้กระบวนการเปรียบเทียบในการจัดการการเชื่อมต่อ ซึ่งถูกใช้ในการสร้างคีย์เชื่อมต่อสำหรับใช้ในการเข้ารหัส
- ความปลอดภัย : กำหนดวิธีการเข้ารหัสและเมื่อใดที่จะทำการเข้ารหัส

สำหรับความสามารถในการค้นหานั้น มีโหมดการทำงานอยู่ 3 โหมด คือ ไม่สามารถค้นพบอุปกรณ์ ค้นหาอุปกรณ์แบบจำกัด และการค้นหาอุปกรณ์แบบทั่วไป โดยโหมดในการค้นหานั้นจะขึ้นอยู่กับกระบวนการอินไควรี่และการสแกนดังนี้

- ไม่สามารถค้นพบอุปกรณ์ ในโหมดนี้จะไม่ทำกระบวนการอินไควรี่สแกน ดังนั้นจะไม่สามารถถูกค้นพบโดยอุปกรณ์ตัวอื่นได้
- ค้นหาอุปกรณ์แบบจำกัด ในโหมดนี้อุปกรณ์จะทำกระบวนการอินไควรี่สแกน โดยอุปกรณ์จะถูกค้นพบเมื่ออุปกรณ์ตัวอื่นค้นหาแบบการค้นหาแบบจำกัดเท่านั้น (LIAC : Limited Inquiry Access Code)
- การค้นหาอุปกรณ์แบบทั่วไป ในโหมดนี้อุปกรณ์จะทำกระบวนการอินไควรี่สแกน โดยอุปกรณ์จะถูกค้นพบเมื่ออุปกรณ์ตัวอื่นค้นหาแบบการค้นหาแบบทั่วไป (GIAC : General Inquiry Access Code)

สำหรับความสามารถในการติดต่อ มีโหมดการทำงานอยู่ 2 โหมด คือเชื่อมต่อและไม่เชื่อมต่อ ในโหมดของการเชื่อมต่อ อุปกรณ์จะทำกระบวนการเพจสแกนเป็นระยะๆเพื่อยอมให้อุปกรณ์บลูทูธอื่นๆมาติดต่อด้วยได้ ส่วนอุปกรณ์ที่ทำงานในโหมดไม่เชื่อมต่อนั้นจะไม่ทำกระบวนการเพจสแกน ดังนั้นอุปกรณ์ในโหมดนี้จะมีการเชื่อมต่อก็ต่อเมื่อตนเองเป็นคนเชื่อมต่อไปยังอุปกรณ์ตัวอื่นเท่านั้น แต่อุปกรณ์ตัวอื่นไม่สามารถเริ่มต้นติดต่อมายังตนเองได้

สำหรับความสามารถในการเปรียบเทียบ มีโหมดการทำงานอยู่ 2 โหมด คือ เปรียบเทียบและไม่เปรียบเทียบ

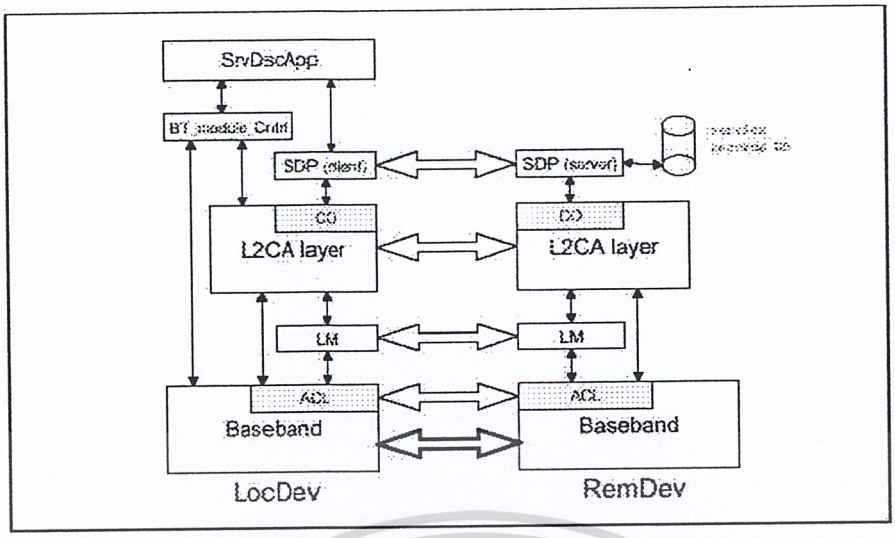
ในเรื่องของความปลอดภัย มีการทำงานอยู่ 3 โหมดด้วยกัน คือ

- โหมด 1 : ไม่มีการรักษาความปลอดภัยใดๆ
- โหมด 2 : การรักษาความปลอดภัยขึ้นอยู่กับระดับของการบริการ
- โหมด 3 : การรักษาความปลอดภัยขึ้นอยู่กับระดับของการเชื่อมต่อ

การพิสูจน์ตนเป็นความสามารถเสริมสำหรับอุปกรณ์ที่ทำงานในโหมด 1 แต่เป็นสิ่งจำเป็นสำหรับอุปกรณ์ที่ทำงานในโหมดที่ 2 และ 3

โพรไฟล์การค้นหาการบริการ (SDAP : Service Discovery Application Profile)

อธิบายถึงกระบวนการค้นหาการบริการที่มีอยู่ในอุปกรณ์นั้นๆ SDAP จะเกี่ยวข้องกับแอพพลิเคชัน โดยแอพพลิเคชันเหล่านี้จะคอยติดต่อสื่อสาร คือ จะคอยส่งและรับ service inquiries จากอุปกรณ์บลูทูธตัวอื่นๆ ดังนั้น SDAP จะอธิบายถึงการเชื่อมต่อระหว่างแอพพลิเคชันกับโพรโตคอลบลูทูธเพื่อดึงเอาประโยชน์สูงสุดในการใช้งานออกมา



รูปที่ 2-15 บลูทูธโพรโตคอลสแต็กสำหรับ SDP

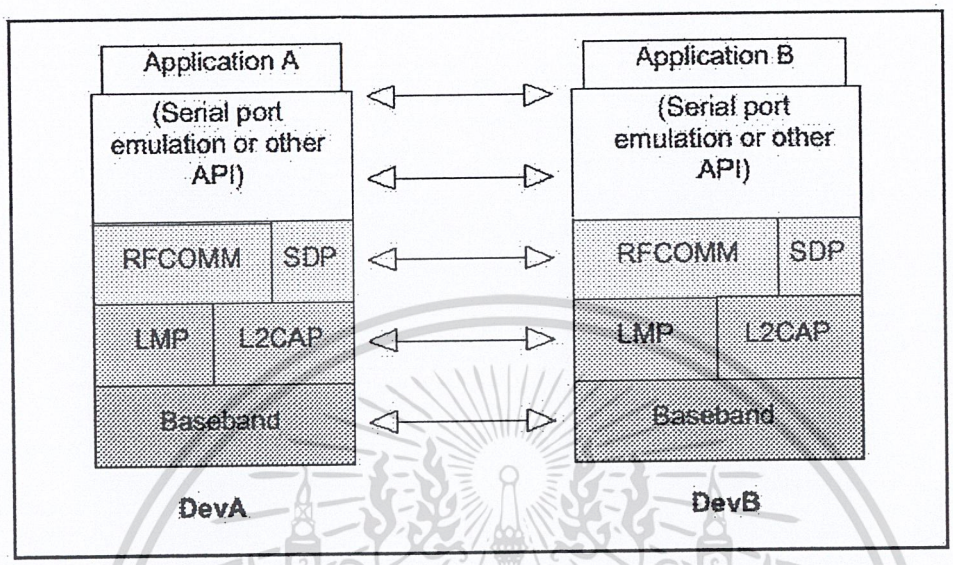
โพรไฟล์จำลองการทำงานผ่านพอร์ตอนุกรม (Serial Port Profile)

เป็นโพรไฟล์ที่จำลองพอร์ตอนุกรม RS-232สำหรับใช้งานในอุปกรณ์บลูทูธ ดังนั้นโปรแกรมประยุกต์ต่างๆไม่มีความจำเป็นต้องแก้ไขเพื่อให้สามารถนำมาใช้กับบลูทูธได้ โปรแกรมเหล่านั้นสามารถทำงานโดยเปรียบเสมือนว่าบลูทูธติดต่อสื่อสารผ่านพอร์ตอนุกรมได้เลย ในโพรไฟล์นี้ไม่เพียงแต่ใช้สำหรับโปรแกรมประยุกต์เท่านั้น แต่ยังทำให้โปรแกรมประยุกต์เหล่านั้นสามารถทำงานร่วมกันได้ด้วย

โพรไฟล์จำลองการทำงานผ่านพอร์ตอนุกรมนี้อิงมาตรฐานของ GSM 07.10 ซึ่งจะยอมให้มีการมัลติเพลกการเชื่อมต่อหลายๆอันบนสายอนุกรมเพียงเส้นเดียว รองรับการทำงานร่วมกับอุปกรณ์ 2 ประเภท คือ อุปกรณ์ที่เป็นปลายทางการเชื่อมต่อ เช่น โน้ตบุก และอุปกรณ์ที่อยู่ระหว่างกลาง ซึ่งเป็นส่วนที่สร้างการเชื่อมต่อ เช่น โมเด็ม อุปกรณ์ชนิด 1 จะจำลองการทำงานของพอร์ตอนุกรมทำให้บลูทูธสามารถรองรับการทำงานกับโปรแกรมประยุกต์ที่ติดต่อผ่านพอร์ตอนุกรมได้ ส่วนอุปกรณ์ชนิด 2 นั้นจะเป็นส่วนหนึ่งที่จะสร้างการเชื่อมต่อกับบลูทูธโดยการแทนที่สายเคเบิลที่เชื่อมอุปกรณ์บลูทูธไปยังโมเด็มหรืออุปกรณ์ที่ทำหน้าที่เป็นตัวกลางอื่นๆ ในโพรไฟล์จำลองการทำงานผ่านพอร์ตอนุกรมนี้ มีโพรไฟล์อื่นๆที่ถูกสร้างขึ้นบนโพรไฟล์นี้ คือ

- การเชื่อมต่อไปยังเครือข่าย (DUN : Dial up Networking)
- แฟกซ์ (Fax)
- หูฟังไร้สาย (Headset)
- การใช้งาน LAN (LAN Access)
- การแลกเปลี่ยนออบเจ็กต์พื้นฐาน (Generic object exchange)
- การรับ-ส่งไฟล์ (File transfer)
- การส่งออบเจ็กต์ (Object push)
- การทำข้อมูลให้ตรงกัน (Synchronization)

โพรไฟล์จำลองการทำงานผ่านพอร์ตอนุกรมใช้ RFCOMM ในการจำลองการทำงานผ่านพอร์ตอนุกรม อุปกรณ์ที่เริ่มต้นติดต่อ RFCOMM เรียกว่าตัวเริ่มต้น (Initiator) และอุปกรณ์อีกตัวเรียกว่าเป็นตัวตอบสนอง (Responder)



รูปที่ 2-16 โมเดลของบลูทูธโพรโตคอลสแต็กแสดงการจำลองการทำงานผ่านพอร์ตอนุกรม

เนื่องจากขั้นตอนในการเริ่มต้นจำลองพอร์ตอนุกรมและสร้างการเชื่อมต่อเป็นสิ่งสำคัญ สำหรับฟังก์ชันในการจัดการกับอุปกรณ์บางชนิดจำเป็นที่จะต้องจำลองพอร์ตอนุกรมและสร้างการเชื่อมต่อก่อน

ขั้นตอนแรกในการจำลองพอร์ตอนุกรมคือการค้นหาแอดเดรสของอุปกรณ์บลูทูธที่ต้องการติดต่อด้วย มีหลายวิธีที่จะทำได้ ดังนี้

- เริ่มกระบวนการอินไควเพื่อทำการค้นหาอุปกรณ์ที่อยู่ในระยะเครือข่าย หลังจากนั้นผู้ใช้เป็นผู้เลือกอุปกรณ์ที่ต้องการติดต่อด้วย
- ผู้ใช้ทำการใส่แอดเดรสของอุปกรณ์โดยตรง
- อุปกรณ์เป็นตัวที่ทำการติดต่อเอง โดยที่ผู้ใช้ไม่จำเป็นต้องมาปรับแต่งค่าอะไรเลย

ขั้นตอนต่อไปเป็นกระบวนการเพจจิง โดยสร้างการการเชื่อมต่อไปยังอุปกรณ์ที่ได้เลือกไว้

แชนเนล L2CAP จะถูกสร้างขึ้นไปยัง SDP เซิร์ฟเวอร์ เมื่อ SDP ได้รับเลข RFCOMM เซิร์ฟเวอร์แชนเนลของการบริการการจำลองพอร์ตอนุกรมแล้ว ในขณะนี้ผู้ใช้จะได้รับข้อมูลเกี่ยวกับชื่อของการบริการที่มีให้ใช้ซึ่งได้รับมาจาก SDP

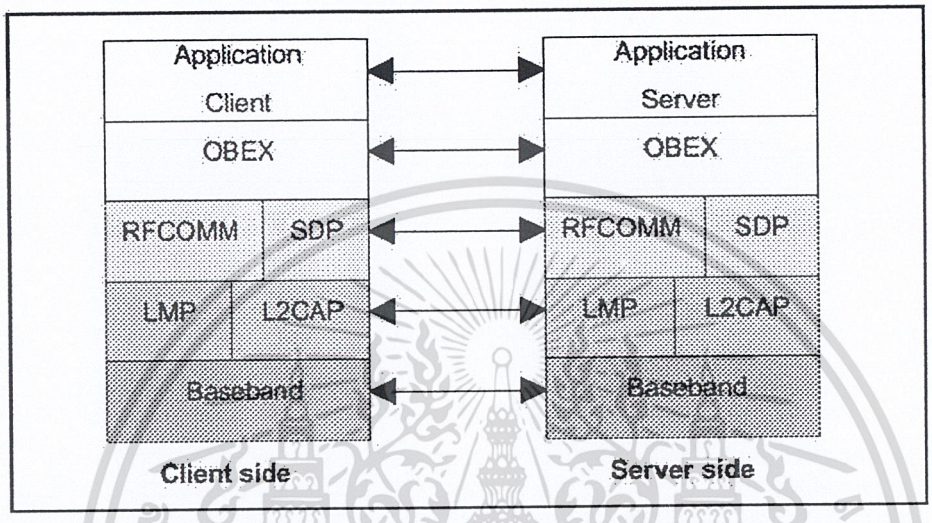
แชนเนล L2CAP จะถูกสร้างขึ้นไปยัง RFCOMM ของฝ่ายตัวตอบสนอง RFCOMM จะทำการมัลติเพลกสัญญาณการควบคุมซึ่งถูกสร้างขึ้นระหว่างแชนเนล L2CAP

ค่าพารามิเตอร์ของการเชื่อมต่อ RFCOMM จะถูกติดต่อก่อนที่การเชื่อมต่อ RFCOMM จะเกิดขึ้น หลังจากนั้นโปรแกรมประยุกต์สามารถติดต่อสื่อสารผ่านพอร์ตอนุกรมที่จำลองไว้ได้บนแชนเนล RFCOMM โดยมีอัตราการรับส่งข้อมูลสูงสุดอยู่ที่ 128 กิโลบิตต่อวินาที

โพรไฟล์การแลกเปลี่ยนวัตถุ (GOEP : Generic Object Exchange Profile)

โพรไฟล์ 3 โพรไฟล์ที่ขึ้นอยู่กับ GOEP คือ

- การรับ-ส่งไฟล์
- การส่งออบเจ็กต์
- การทำข้อมูลให้ตรงกัน



รูปที่ 2-17 โมเดลของบลูทูธโพรโตคอลสแต็กแสดงการทำงานของการแลกเปลี่ยนวัตถุ

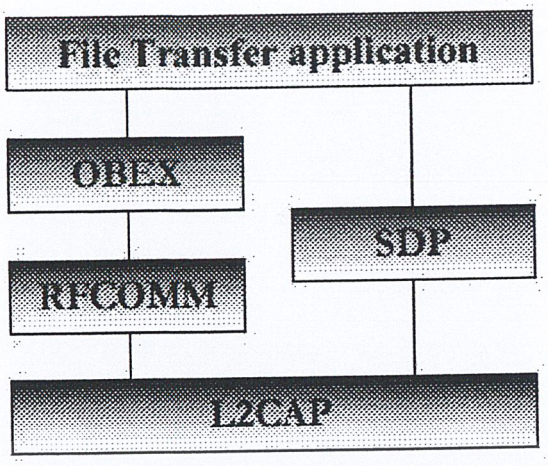
โพรไฟล์การแลกเปลี่ยนวัตถุใช้เลเยอร์ของ IrDA OBEX ซึ่งได้กำหนดวิธีใช้กับบลูทูธ นอกจากนี้ยังกำหนดวิธีการเริ่มต้นการติดต่อสื่อสารแบบไคลเอ็นต์-เซิร์ฟเวอร์ โพรไฟล์นี้ไม่ได้กำหนดถึงเลเยอร์ของโปรแกรมประยุกต์ แต่เลเยอร์โปรแกรมประยุกต์นี้จะกล่าวในโพรไฟล์การรับ-ส่งไฟล์, โพรไฟล์การส่งออบเจ็กต์ และโพรไฟล์การทำข้อมูลให้ตรงกัน

โพรไฟล์การแลกเปลี่ยนวัตถุกำหนดหน้าที่ 2 อย่าง คือ อุปกรณ์ที่เป็นเซิร์ฟเวอร์ในการที่จะส่งและรับออบเจ็กต์ และอุปกรณ์ที่เป็นไคลเอนต์ในการที่จะส่งและรับออบเจ็กต์ โดยจะใช้โหมดการทำงาน คือ การเชื่อมต่อ, ยกเลิกการเชื่อมต่อ, ส่งออบเจ็กต์, รับออบเจ็กต์ และการกำหนดเส้นทาง

2.5.2 ภาพการนำบลูทูธไปใช้งาน

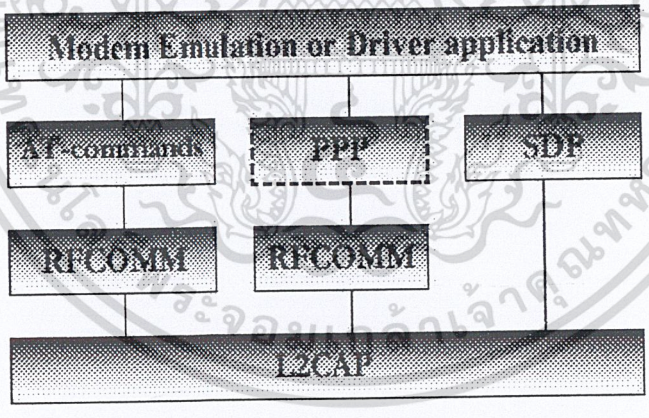
เนื่องจากบลูทูธเหมาะที่จะใช้งานในลักษณะของเครือข่ายส่วนบุคคล ดังนั้นเราสามารถนำบลูทูธมาใช้งานในลักษณะต่างๆดังนี้ คือ

- การรับส่งไฟล์ : การรับส่งไฟล์เป็นการรับส่งข้อมูลจากอุปกรณ์บลูทูธตัวหนึ่งไปยังอีกตัวหนึ่งๆได้ ความหมายของไฟล์ในที่นี้อาจหมายรวมถึง ไฟล์ข้อมูล, ไดเร็กทอรีหรือสายข้อมูลก็ได้ การรับส่งไฟล์นี้สามารถที่จะดูถึงเนื้อหาข้อมูลภายในโฟวเดอร์ที่อยู่ในอุปกรณ์อีกฝั่งหนึ่งได้ด้วย นอกจากนี้การส่งและการแลกเปลี่ยนข้อมูล เช่น การแลกเปลี่ยนนามบัตรทางธุรกิจยังกำหนดรายละเอียดไว้ในโพรไฟล์นี้เช่นกัน



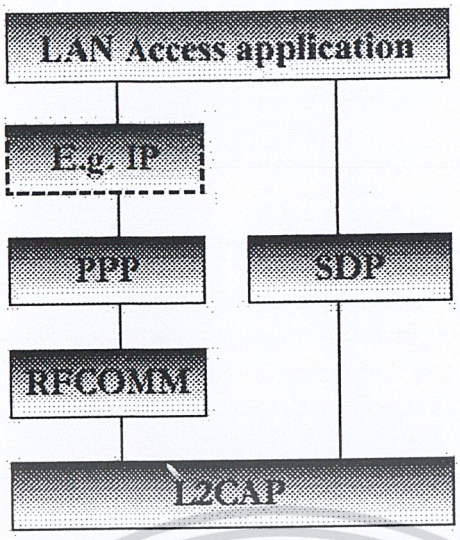
รูปที่ 2-18 โพรโตคอลสแต็กสำหรับการรับส่งไฟล์

● อินเทอร์เน็ตบริดจ์: ในโพรไฟล์นี้ โทรศัพท์มือถือหรือโมเด็มไร้สายจำทำหน้าที่เป็นโมเด็มที่เชื่อมต่อไปยังคอมพิวเตอร์ ซึ่งจะจัดเตรียมความสามารถในการติดต่อไปยังเครือข่ายและความสามารถในการรับส่งแฟกซ์ไว้ให้ โดยที่ไม่ต้องเชื่อมต่อกับคอมพิวเตอร์จริงๆเลย การใช้งานการติดต่อไปยังเครือข่ายนั้นจำเป็นต้องใช้โพรโตคอลอีกสองตัวมาช่วย คือ AT-commands ใช้ในการควบคุมการทำงานของโทรศัพท์มือถือหรือโมเด็ม และ PPP ที่ช่วยในการส่งผ่านข้อมูล



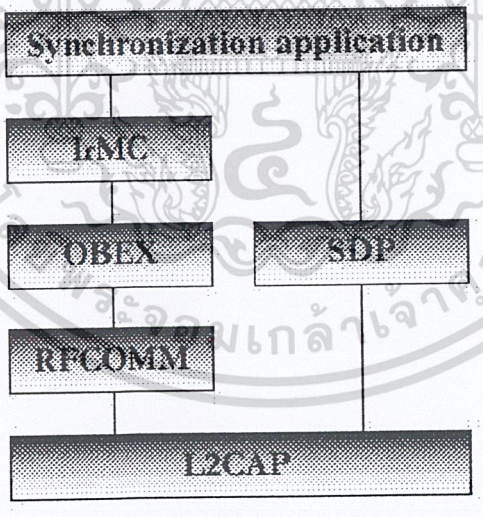
รูปที่ 2-19 โพรโตคอลสแต็กสำหรับอินเทอร์เน็ตบริดช์

● การเข้าถึงระบบแลน : ทำให้เทอร์มินัลสามารถที่จะเข้าถึงระบบแลนโดยปราศจากการเชื่อมต่อสายเคเบิล ลักษณะจะเป็นเหมือนระบบแลนไร้สาย เมื่อเทอร์มินอลต่างๆสามารถที่จะเชื่อมต่อได้แล้วจะทำงานราวกับว่ามันเชื่อมต่อไปยังแลนผ่านการติดต่อไปยังเครือข่าย เทอร์มินัลสามารถที่จะเรียกใช้งานการบริการต่างๆที่มีอยู่ในแลนได้



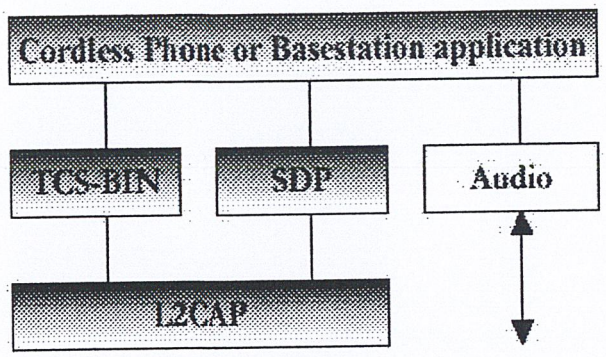
รูปที่ 2-20 โพรโตคอลสแต็กสำหรับการเข้าถึงระบบแลน

- การทำข้อมูลให้ตรงกัน : เป็นการทำให้ข้อมูลที่มีอยู่ในอุปกรณ์ต่างๆถูกต้องตรงกันโดยที่ไม่ต้องคำนึงถึงแหล่งที่ผลิตของตัวอุปกรณ์ อุปกรณ์ที่สามารถใช้ประโยชน์ในส่วนนี้ได้ เช่น คอมพิวเตอร์, โทรศัพท์มือถือ, คอมพิวเตอร์พกพา และโน้ตบุ๊ก ข้อมูลที่จะนำมาทำให้ตรงกันจะเป็นพวกข้อมูลการนัดหมายในปฏิทิน, ข้อมูลชื่อ ที่อยู่, รายการสิ่งที่ต้องทำในแต่ละวัน เป็นต้น



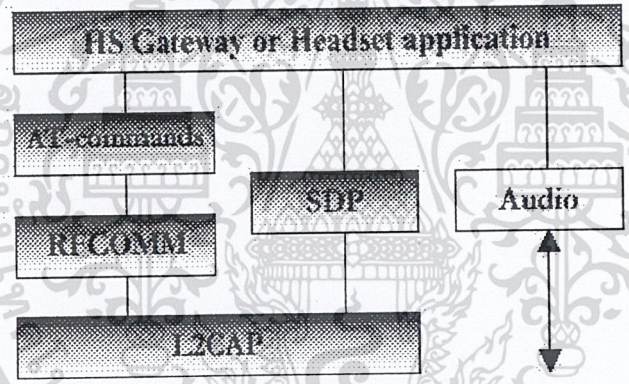
รูปที่ 2-21 โพรโตคอลสแต็กสำหรับการทำข้อมูลให้ตรงกัน

- โทรศัพท์ 3 ฟังก์ชัน : ทำให้โทรศัพท์เครื่องเดียวกันสามารถเป็นโทรศัพท์ธรรมดาเมื่อใช้งานอยู่ที่บ้าน โดยจะติดต่อไปยัง PSTN ดังนั้นการคิดเงินจะคิดเหมือนใช้โทรศัพท์บ้านทั่วไป ,เป็นวิทยุสื่อสาร (walkie-talkie) ไม่เสียค่าบริการ และสุดท้ายเป็นโทรศัพท์ไร้สายซึ่งจะคิดเงินตามเวลาการใช้งาน



รูปที่ 2-22 โพรโตคอลสแต็กสำหรับโทรศัพท์ 3 ฟังก์ชัน

● เฮดเซทไร้สาย : สามารถใช้งานเฮดเซทไร้สายนี้กับเครื่องคอมพิวเตอร์หรือโทรศัพท์มือถือ โดยในการเชื่อมต่ออาจจะทำหน้าที่เป็นตัวรับสัญญาณหรือตัวส่งสัญญาณก็ได้ ทำให้ผู้ใช้มีอิสระในการเคลื่อนไหวที่จะเดินไปยังที่ต่างๆ ได้โดยที่ยังอยู่ภายในระยะเครือข่ายเท่านั้น



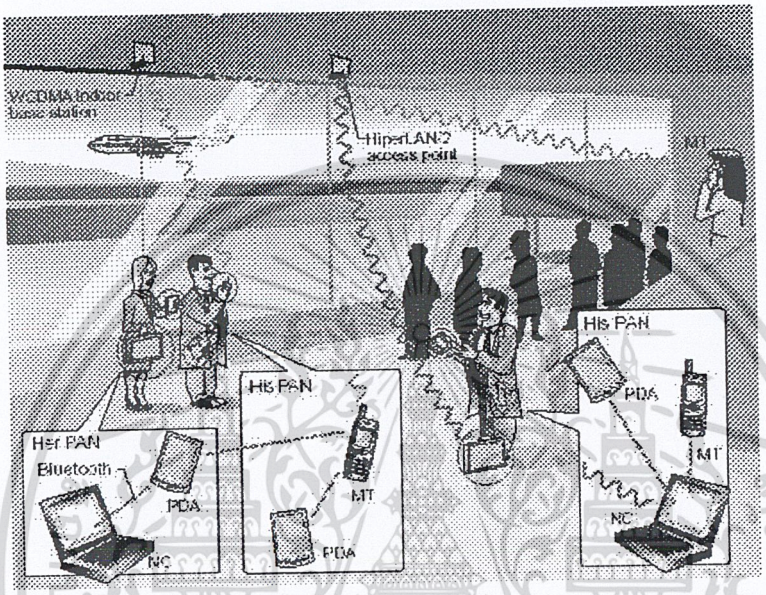
รูปที่ 2-23 โพรโตคอลสแต็กสำหรับเฮดเซทไร้สาย

2.6 ระบบเครือข่ายแบบแอดฮอค

ระบบเครือข่ายแบบแอดฮอคเป็นเครือข่ายที่ก่อรูปขึ้นโดยไม่มีการอำนวยการส่วนกลาง ประกอบด้วยโหนดเคลื่อนที่ที่มีการเชื่อมต่อกันแบบไร้สาย เพื่อส่งข้อมูลในรูปของแพ็กเกตระหว่างกัน เครือข่ายในลักษณะนี้สามารถที่จะมาช่วยให้การดำรงชีวิตในประจำวันของเราง่ายขึ้นมาก โดยการที่อุปกรณ์ต่างๆ สามารถที่ติดต่อสื่อสารและสร้างการเชื่อมต่อขึ้นมาเองโดยไม่ต้องมีการดูแลจากส่วนกลาง พร้อมทั้งสิ้นสุดการเชื่อมต่อทันทีในกรณีที่อุปกรณ์ตัวใดตัวหนึ่งออกนอกเครือข่าย

การใช้งานของเครือข่ายแบบแอดฮอคในปัจจุบันจะแสดงดังรูป เป็นการใช้งานเครือข่ายแบบแอดฮอคผ่านบลูทูธ ภายในสนามบิน หลายคนสามารถติดต่อเข้าใช้เครือข่ายแลน (LAN) และแวน (WAN) ได้ โดยทำการเชื่อมโยงอุปกรณ์สื่อสารต่างๆที่พกพาไปยังจุดติดต่อของเครือข่ายที่ติดตั้งอยู่ใกล้เคียงในบริเวณนั้น และมีการพกพาอุปกรณ์ต่างๆที่สามารถติดต่อเชื่อมโยงสื่อสารกันเป็นเครือข่ายได้ อุปกรณ์ของ

ผู้ใช้งานสามารถเชื่อมโยงติดต่อระหว่างกัน หรือเชื่อมโยงไปยังจุดติดต่อของเครือข่าย ซึ่งตัวอย่างที่แสดงในรูปนั้น ที่ประตูทางออกจะมีจุดติดต่อของเครือข่าย โดยเป็นจุดที่ผู้ใช้สามารถที่จะใช้อุปกรณ์พกพาต่างๆ เพื่อทำการเรียกดูข้อมูลของสายการบินได้ และเมื่อผู้ใช้เรียกดูข้อมูลเสร็จสิ้นแล้วก็จะสามารถทำการยกเลิกการเชื่อมต่อได้ในทันที ซึ่งจะเห็นได้ว่าเป็นเครือข่ายที่มีความยืดหยุ่นสูง อุปกรณ์ต่างๆไม่ได้ถูกกำหนดค่าขึ้นมาเพื่อใช้งานในเครือข่ายใดๆก่อนเลย และเนื่องจากไม่ได้มีการจัดการจากส่วนกลางนั้น ทำให้อุปกรณ์ต่างๆต้องมีความสามารถในการจัดการการเชื่อมต่อและการปรับแต่งค่าต่างๆได้ด้วยตนเอง



รูปที่ 2-24 แสดงการใช้งานระบบเครือข่ายแบบแอดฮอคผ่านบลูทูธภายในสนามบิน

แนวคิดของระบบเครือข่ายแบบแอดฮอค ประกอบด้วยหลักการ 4 อย่าง คือ

- Spontaneous

อุปกรณ์ที่ทำงานในระบบเครือข่ายแบบแอดฮอคนั้นจะทำการติดต่อสื่อสารกับอุปกรณ์อื่นด้วยตนเอง โดยที่ไม่ต้องมีอุปกรณ์อื่นๆมากระตุ้นให้เริ่มทำงาน ซึ่งในการใช้งานนั้นอุปกรณ์จะทำการเชื่อมต่อด้วยตนเองโดยที่ไม่จำเป็นต้องรอการสั่งงานจากผู้ใช้ ซึ่งจะแตกต่างจากการทำงานในระบบเครือข่ายแบบเดิม เนื่องจากต้องรอคำสั่งจากผู้ใช้ถึงจะเริ่มต้นทำงาน

- Autonomous

อุปกรณ์ที่ทำงานในระบบเครือข่ายแบบแอดฮอคนั้นจะทำงานด้วยตนเอง เป็นอิสระ ไม่ต้องถูกควบคุมจากส่วนกลาง ดังนั้นอุปกรณ์ที่ทำงานในระบบเครือข่ายแบบนี้นั้นต้องมีความสามารถในการจัดการการเชื่อมต่อและการปรับแต่งค่าต่างๆได้ด้วยตนเอง เมื่อใช้งานในเครือข่ายเรียบร้อยแล้ว ก็จะสามารถยกเลิกการเชื่อมต่อกับอุปกรณ์อื่นได้ด้วยตนเองเช่นกัน

- Dynamic

ระบบเครือข่ายแบบแอดฮอคเป็นระบบเครือข่ายที่มีความยืดหยุ่นสูง เนื่องจากอุปกรณ์ต่างๆไม่ได้มีการปรับแต่งค่าเพื่อถูกทำงานในระบบเครือข่ายใดๆมาก่อน แต่อุปกรณ์ต่างๆสามารถสร้างการเชื่อมต่อ

กับอุปกรณ์อื่นเป็นระบบเครือข่ายแบบเฉพาะกิจขึ้นมา ณ เมื่อใด เวลาใดได้ และสามารถยกเลิกการเชื่อมต่อทันทีเมื่อใช้งานเสร็จเรียบร้อยแล้ว

- Mobile

เนื่องจากความยืดหยุ่นของระบบเครือข่ายแบบแอดฮอค ที่สามารถสร้างการเชื่อมต่อ ณ เวลาใดที่ใดได้ ทำให้อุปกรณ์ที่ทำงานในเครือข่ายนี้สามารถที่จะเคลื่อนที่ไปยังที่ต่างๆได้โดยที่ยังอยู่ภายในระยะเครือข่าย ซึ่งก่อให้เกิดการใช้งานที่สะดวกสบาย และเป็นธรรมชาติมากยิ่งขึ้น

จากแนวคิดของระบบเครือข่ายแบบแอดฮอคนั้นจะเห็นได้ว่าต่างจากแนวคิดของระบบเครือข่ายแบบเดิม จากเดิมที่อุปกรณ์ต่างๆถูกจัดการจากส่วนกลาง ต้องถูกปรับแต่งค่าก่อนที่จะใช้งานในเครือข่าย และเมื่อต้องเปลี่ยนรูปแบบเครือข่ายนั้นต้องทำการปรับแต่งค่าใหม่ทุกครั้ง ทำให้อุปกรณ์ต่างๆถูกจำกัดการใช้งานเฉพาะในเครือข่ายที่กำหนดมาเท่านั้น ขาดความยืดหยุ่นในการใช้งาน ซึ่งต่างจากแนวคิดของระบบเครือข่ายแบบแอดฮอคที่ไม่ได้ถูกควบคุมจากส่วนกลาง ทำให้อุปกรณ์ต่างๆต้องมีความสามารถในการจัดการการเชื่อมต่อและปรับแต่งค่าต่างๆด้วยตนเอง ทำให้เกิดความยืดหยุ่นในการใช้งานที่สูงขึ้น

การทำให้อุปกรณ์สามารถทำงานในระบบเครือข่ายแบบแอดฮอคได้นั้น จะมีประเด็นที่ต้องพิจารณาเพิ่มขึ้นมาจากแนวความคิดของระบบเครือข่ายแบบเดิม ซึ่งประเด็นหนึ่งที่น่าสนใจ คือ การเลือกเส้นทางในติดต่อสื่อสารกับอุปกรณ์ต่างๆ เนื่องจากอุปกรณ์ที่ทำงานในระบบเครือข่ายแบบแอดฮอคนี้ มักเป็นอุปกรณ์ที่สามารถพกพา หรือเคลื่อนที่ขณะที่ใช้งานได้ ทำให้ต้องมีกระบวนการในการเลือกเส้นทางที่จะทำการติดต่อสื่อสารกับอุปกรณ์ตัวอื่นๆ ซึ่งรวมไปถึงในกรณีที่อุปกรณ์ต่างๆเคลื่อนที่ออกจากเครือข่ายเดิม แล้วเข้าไปยังเครือข่ายใหม่ โพรโตคอลในการเลือกเส้นทางนี้ต้องสามารถทำการยกเลิกการเชื่อมต่อกับเครือข่ายเดิมแล้วทำการเชื่อมต่อกับเครือข่ายใหม่ที่กำลังเข้าไปใช้งาน

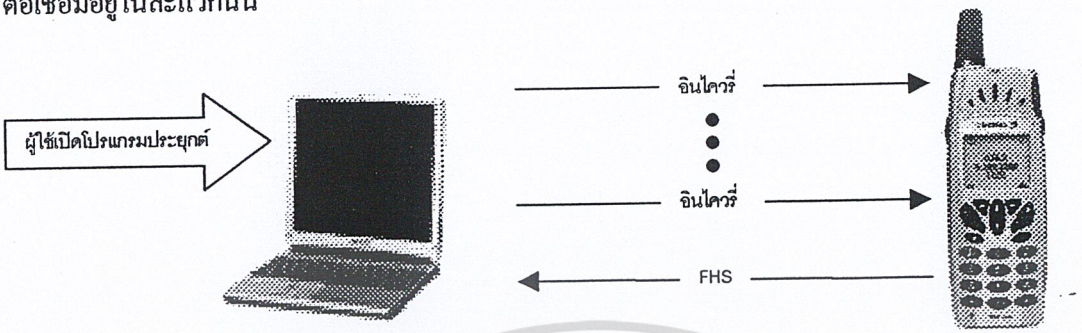
การนำบลูทูธมาใช้งานเป็นระบบเครือข่ายแบบแอดฮอคนั้น ต้องมีความเข้าใจพื้นฐานลักษณะการทำงานของบลูทูธเสียก่อน เนื่องจากบลูทูธนั้นไม่เหมือนกับระบบเครือข่ายที่ใช้สายเคเบิลในการเชื่อมต่อ เราสามารถที่จะรู้ได้ว่ากำลังติดต่อสื่อสารอยู่กับอุปกรณ์ตัวไหนและสามารถใช้ประโยชน์อะไรจากอุปกรณ์เหล่านั้นได้บ้าง โดยที่ไม่ต้องการการเชื่อมต่อทางกายภาพระหว่างตัวอุปกรณ์ ในการที่บลูทูธจะสามารถทำตามหลักการดังกล่าวได้ ต้องประกอบไปด้วยขั้นตอนหลักๆคือ การทำอินไควรี่ การทำเพจจิงและโพรโตคอลค้นหาการบริการ (SDP : Service Discovery Protocol) ซึ่งเราสามารถมองภาพการทำงานออกเป็น 2 ส่วน คือ

- การค้นหาอุปกรณ์บลูทูธ

ในที่นี้จะจำลองภาพการทำงานระหว่างอุปกรณ์บลูทูธ 2 ตัว คือ โน้ตบุ๊กและโทรศัพท์มือถือ โดยที่ตัวโทรศัพท์มือถือมีความสามารถในการเชื่อมต่อไปยังเครือข่ายอินเทอร์เน็ตเหมือนกับโมเด็ม และคอยค้นหาอยู่เป็นระยะๆว่ามีใครต้องการใช้ความสามารถในส่วนนี้ของมันบ้าง

เริ่มจากการที่ผู้ใช้เปิดโน้ตบุ๊กเปิดใช้งานแอพพลิเคชันที่ต้องการการเชื่อมต่อไปยังเครือข่ายอินเทอร์เน็ต เพื่อที่จะให้ใช้งานได้ โน้ตบุ๊กรู้ว่าจะต้องเชื่อมต่อไปยังอุปกรณ์บลูทูธตัวอื่นๆที่สามารถทำการเชื่อมต่อไปยังเครือข่ายอินเทอร์เน็ตได้ สิ่งแรกที่ทำคือ เริ่มสร้างการเชื่อมต่อที่จะหาว่ามีอุปกรณ์บลูทูธอะไรบ้างอยู่ในละแวกนั้น ดังนั้น โน้ตบุ๊กจึงเริ่มกระบวนการอินไควรี่เพื่อหาอุปกรณ์ที่อยู่ข้างเคียง โดยการ

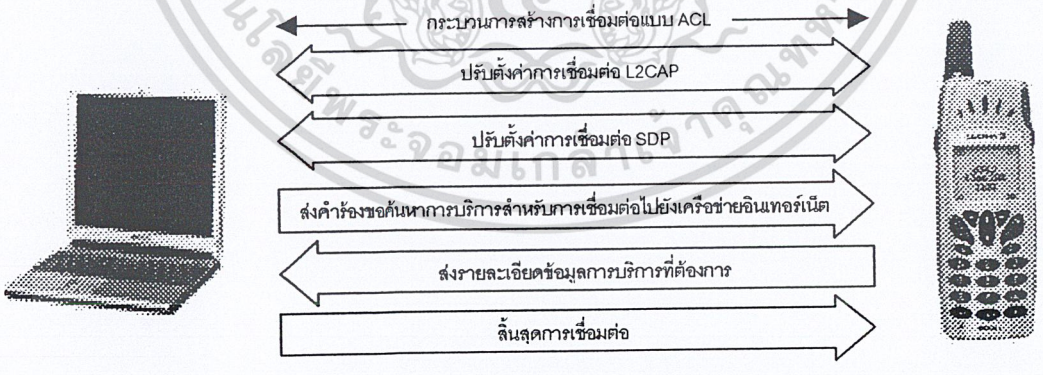
ส่งแพ็กเกตอินไควรี่ออกไปเป็นชุดๆ เมื่อโทรศัพท์มือถือได้รับการอินไควแล้ว ก็จะตอบรับกลับมาเป็นแพ็กเกต FHS (Frequency Hop Synchronization) ในแพ็กเกตนี้จะประกอบไปด้วยข้อมูลต่างๆที่โน้ตบุ๊กจะสามารถนำไปใช้สร้างการเชื่อมต่อได้ เมื่อเสร็จสิ้นกระบวนการนี้ โน้ตบุ๊กจะได้รายชื่อของอุปกรณ์ต่างๆที่ต่อเชื่อมอยู่ในละแวกนั้น



รูปที่ 2-25 การค้นหาอุปกรณ์บลูทูธ

- การค้นหาการบริการ

ในการที่จะรู้ว่าอุปกรณ์ต่างๆที่เชื่อมต่ออยู่นั้นสามารถนำไปใช้ประโยชน์อะไรได้บ้าง ตัวแอพพลิเคชันเองจะต้องติดต่อไปยังตัวอุปกรณ์และใช้โพรโตคอลค้นหาการบริการในการสื่อสาร จากตัวอย่างที่กล่าวมา เริ่มจากการที่โน้ตบุ๊กทำเพจจิงไปยังโทรศัพท์มือถือ เพื่อสร้างช่องทางการสื่อสารระหว่าง 2 อุปกรณ์ แล้วจึงร้องขอไปยังส่วนค้นหาการบริการที่มีอยู่ในตัวโทรศัพท์มือถือว่าให้ส่งข้อมูลทั้งหมดที่เกี่ยวข้องกับการเชื่อมต่อไปยังเครือข่ายอินเทอร์เน็ตกลับมา หลังจากนั้นตัวค้นหาการบริการในโทรศัพท์มือถือจะค้นหาผ่านฐานข้อมูลในตัวเองแล้วส่งผลลัพธ์กลับไปให้



รูปที่ 2-26 การค้นหาการบริการ

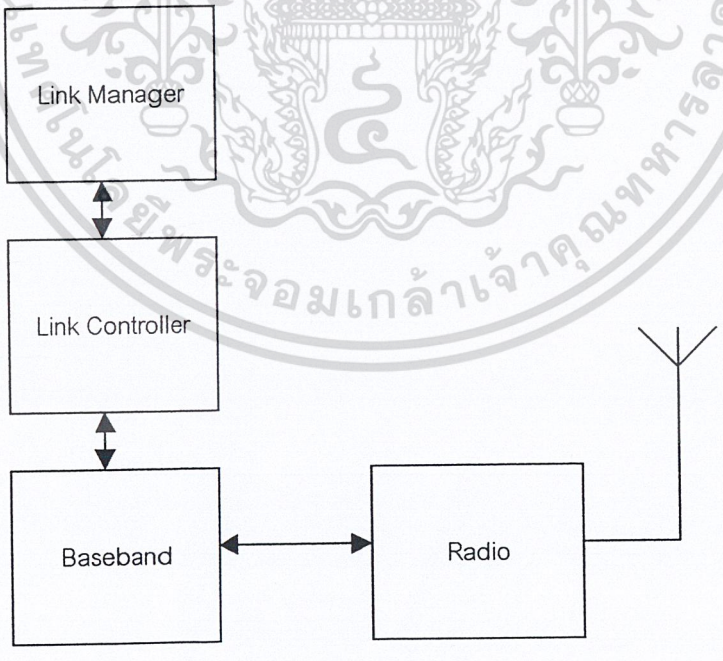
เมื่อทราบถึงกระบวนการสร้างเครือข่ายแอดฮอคแบบคร่าวๆแล้ว ในบทถัดไปนี้จะมาทำความเข้าใจในรายละเอียดของบลูทูธมากยิ่งขึ้น ซึ่งเป็นการอธิบายถึงการทำงานในชั้นต่างๆของบลูทูธโพรโตคอลสแต็ก แบ่งออกเป็นสองส่วนใหญ่ๆด้วยกัน คือ บลูทูธโมดูลซึ่งประกอบด้วยเลเยอร์ระดับล่าง และบลูทูธโฮสต์ซึ่งประกอบด้วยเลเยอร์ระดับบนของโพรโตคอลสแต็ก

บทที่ 3
บลูทูธโมดูล

3.1 เบสแบน (Baseband)

ในหัวข้อนี้จะเริ่มต้นด้วยการทำความเข้าใจระหว่างคำสองคำ คือ "ส่วนควบคุมการเชื่อมต่อ" (Link Controller) และ "เบสแบน" ในรายละเอียดบลูทูธเวอร์ชัน 1.0 นั้นได้ใช้คำ 2 คำนี้อย่างกำกวม เพื่อที่จะเข้าใจหลักการทำงานที่จะกล่าวถึงต่อไปจะกล่าวถึงความหมายของคำสองคำนี้ก่อน

ส่วนควบคุมการเชื่อมต่อมีหน้าที่รับผิดชอบในการส่งการทำงานในระดับของการเชื่อมต่อบนแพ็กเกตของข้อมูลหลายๆอันภายในช่วงเวลาที่ตอบสนองกับคำสั่งที่อยู่ระดับบนจากส่วนจัดการการเชื่อมต่อ (Link Manager) ส่วนควบคุมการเชื่อมต่อทั้งระยะใกล้และระยะไกลนั้นจะต้องจัดการแพ็กเกตโดยกระบวนการทำงานของแพ็กเกตในการสร้างการเชื่อมต่อจากส่วนจัดการการเชื่อมต่อ และต้องคอยรักษาการเชื่อมต่อนี้ไว้ให้คงอยู่ต่อไป ในชั้นกายภาพของ OSI นั้นแทนได้ด้วยคลื่นวิทยุและเบสแบน ส่วนเชื่อมต่อคลื่นวิทยุระหว่างตัวกลางที่อยู่ในอากาศและเบสแบนนั้นจะทำให้เกิดรูปแบบของข้อมูลที่สร้างมาจากส่วนควบคุมการเชื่อมต่อเพื่อความคงทนของสัญญาณและความน่าเชื่อถือในการส่งข้อมูลบนแชนแนลและรับข้อมูลจากแชนแนลส่งกลับไปยังแสตก เบสแบนมีหน้าที่ในการเข้ารหัสแชนแนล, ถอดรหัส, ควบคุมเวลาการทำงานในระดับล่าง และจัดการการเชื่อมต่อเมื่อแพ็กเกตข้อมูลแต่ละแพ็กเกตถูกส่งออกไป



รูปที่ 3-1 ส่วนควบคุมการเชื่อมต่อและเบสแบน

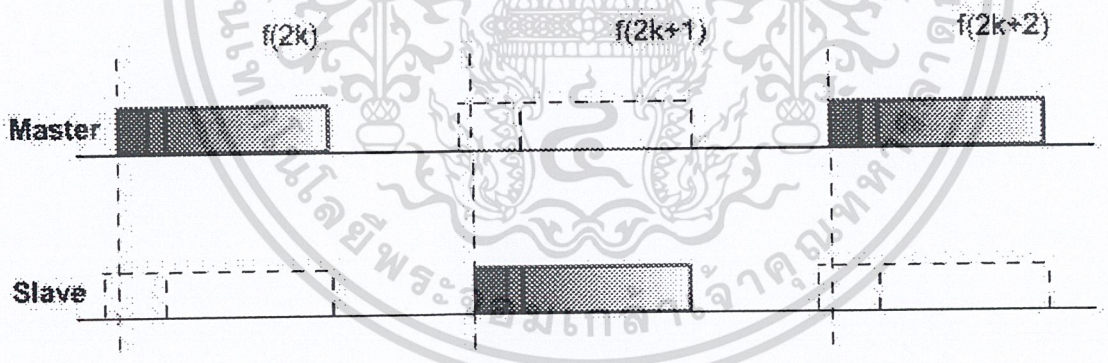
3.1.1 แอดเดรสอุปกรณ์บลูทูธ

แต่ละอุปกรณ์บลูทูธจะมีเลข 48 บิต (IEEE MAC address) ซึ่งเรียกกันว่าเป็นแอดเดรสอุปกรณ์บลูทูธ (BD_ADDR) สามารถแบ่งออกเป็น 3 ส่วนคือ แอดเดรสส่วนที่ไม่มีความสำคัญ (NAP : Non-significant Address Part), แอดเดรสส่วนบน (UAP : Upper Address Part) และแอดเดรสส่วนล่าง (LAP : Lower Address Part)

- BD_ADDR[47:32] – NAP[15:0]
  ใช้ในการเริ่มต้นการเข้ารหัส
- BD_ADDR[31:24] – UAP[7:0]
  ใช้ในการเริ่มต้นการคำนวณ HEC และ CRC และใช้ในการกระโดดของความถี่
- BD_ADDR[23:0] – LAP[23:0]
  ใช้ในการเข้าจังหวะของข้อมูลและการกระโดดของความถี่

3.1.2 มาสเตอร์ สเลฟ และพิโคเน็ต

บลูทูธมีการมัลติเพล็กสัญญาณโดยใช้เวลาเป็นตัวแบ่ง (TDM : Time Division Multiplexed) โดยช่วงเวลาการทำงานในแต่ละสล็อตคือ 625 μs ในการติดต่อสื่อสารการรับและส่งข้อมูลจะเกิดขึ้นใน 1,3,5 สล็อต ถ้าเป็นการติดต่อสื่อสารก่อนที่จะมีการเชื่อมต่อนั้น การรับส่งข้อมูลอาจเกิดขึ้นได้ในครึ่งสล็อต แพ็กเกตต่างๆสามารถที่นำมาเชื่อมต่อกันในการรับส่งข้อมูลเป็นคู่ได้ เช่น 2,4,6,8,10 เป็นต้น



รูปที่ 3-2 เวลาการทำงานในแพ็กเกตสล็อตเดี่ยว

อุปกรณ์บลูทูธแต่ละตัวอาจจะเป็นมาสเตอร์หรือสเลฟในขณะใดขณะหนึ่งเท่านั้น ซึ่งบทบาทของ 2 หน้าที่นี้ คือ

- มาสเตอร์ – เป็นอุปกรณ์ที่เริ่มต้นแลกเปลี่ยนข้อมูลก่อน
- สเลฟ – เป็นอุปกรณ์ที่ตอบสนองการทำงานของตัวมาสเตอร์

นอกจากนี้ในขณะที่ทำการติดต่อสื่อสารอยู่นั้น อุปกรณ์ที่เป็นสเลฟจะใช้ประโยชน์จากเวลาของตัวมาสเตอร์เพื่อใช้ในการกระโดดความถี่ ทำให้อุปกรณ์ทั้งคู่สามารถติดต่อสื่อสารกันได้อย่างถูกต้อง จาก

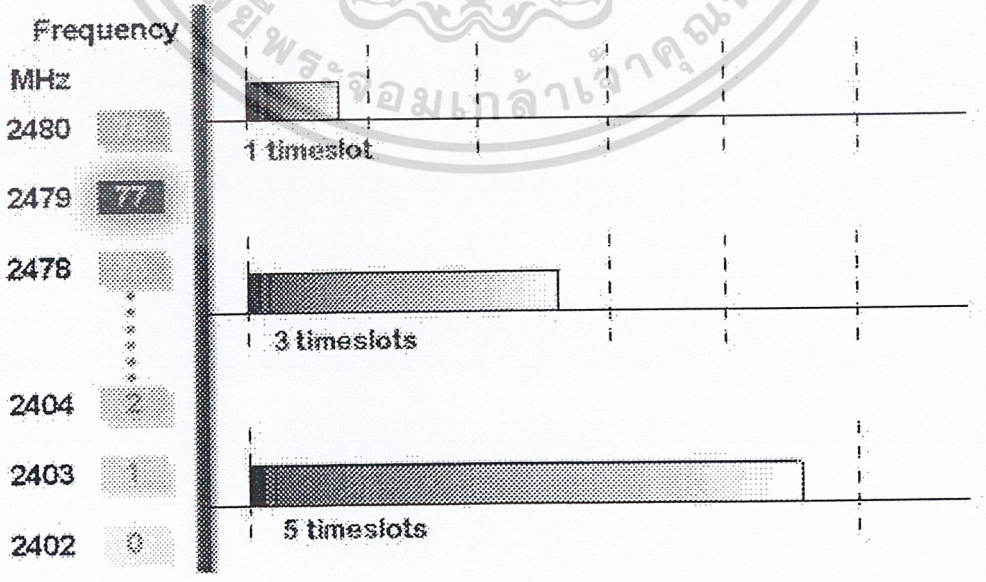
รูปที่ 3-2 นั้น มาสเตอร์เป็นตัวที่เริ่มทำการติดต่อไปยังอุปกรณ์ที่เป็นสเลฟ ซึ่งเหตุการณ์นี้จะเกิดขึ้นเมื่ออุปกรณ์ทั้งคู่ใช้แชนแนลคลื่นวิทยุ f(2k) อีก 625 μs ต่อมา อุปกรณ์ทั้งสองจะทำการปรับคลื่นวิทยุของตนหรือทำการกระโดดความถี่ที่ใช้งานอยู่ไปยัง f(2k+1) ในขณะนี้สเลฟไม่ได้รับอนุญาตให้ทำการส่งข้อมูล แต่จะเป็นการตอบกลับไปยังตัวมาสเตอร์ว่าได้รับแพ็กเกตสุดท้ายที่ส่งไปให้หรือไม่ หลังจากนั้นสเลฟจะคอยรับฟังว่ามีข้อมูลส่งมาหามันหรือไม่ อย่างไรก็ตามมาสเตอร์อาจเลือกที่จะส่งข้อมูลไปยังอุปกรณ์ตัวอื่นหรือไม่ส่งเลยก็เป็นได้ โดยที่สเลฟจะไม่ส่งข้อมูลไปหามาสเตอร์อีกจนกว่ามาสเตอร์จะส่งข้อมูลมาหามันอีกครั้ง

อุปกรณ์แต่ละตัวจะทำการกระโดดความถี่หนึ่งครั้งต่อข้อมูลหนึ่งแพ็กเกต ซึ่งเป็นหลักการพื้นฐานในการส่งข้อมูลของบลูทูธ ข้อดีในการส่งแบบนี้คือ

- ความปลอดภัย – เนื่องจากการกระโดดของความถี่นี้ถูกกำหนดจากลำดับที่สุ่มขึ้นมาโดยสร้างจากแอดเดรสของอุปกรณ์ที่เป็นมาสเตอร์
- ความน่าเชื่อถือ – เนื่องจากถ้าการรบกวนของสัญญาณทำให้ข้อมูลแพ็กเกตเสียหายตรง Ch (n) โอกาสที่การรบกวนจะเกิดตรง Ch(n+m) จะเป็นไปได้น้อย โดย n+m จะรับประกันว่าเป็นระยะห่างจาก n โดยการกำหนดของอัลกอริทึมในการสุ่มการกระโดดของความถี่

บลูทูธกำหนดแพ็กเกตข้อมูลซึ่งมีขนาด 1,3,5 สล็อต รูป 3-3 แสดงเวลาการทำงานที่แตกต่างกันของแพ็กเกตเหล่านี้ ในการใช้แพ็กเกตที่มีขนาดยาวจะทำให้ได้ความเร็วในการส่งข้อมูลที่สูงขึ้น ทุกๆแพ็กเกตจะมีส่วนหัวของข้อมูลที่เหมือนกัน ดังนั้นจะเห็นได้ว่าแพ็กเกตที่เป็นแบบหลายสล็อตนั้นจะมีประสิทธิภาพในการส่งข้อมูลมากกว่า เนื่องจากสามารถส่งข้อมูลทีละมากๆโดยมีส่วนหัวข้อมูลเพียงครั้งเดียว

มาสเตอร์เป็นผู้ที่คอยควบคุมการรับส่งข้อมูล โดยจะเป็นตัวตัดสินใจว่าจะส่งข้อมูลด้วยสล็อตแบบใดและเมื่อไหร่ที่จะทำการส่งข้อมูล สเลฟเป็นตัวที่จะคอยฟังอย่างเงียบๆภายในช่วงเวลาที่คาดว่าจะได้รับข้อมูล โดยการรับข้อมูลที่เป็นแอดเดรสของสเลฟเอง และตอบกลับไป

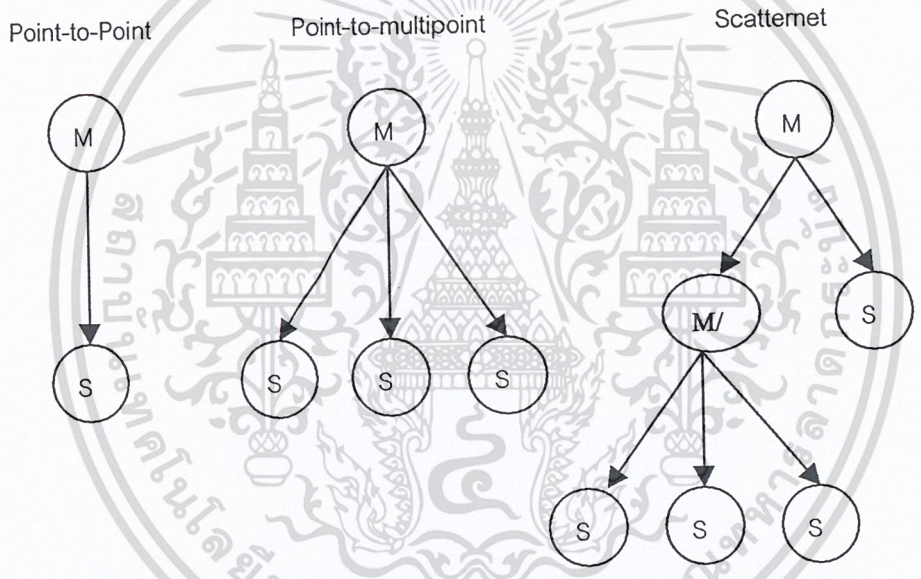


รูปที่ 3-3 รูปแบบต่างๆของสล็อต

บลูทูธเป็นเทคโนโลยีที่มีประโยชน์ถ้าสิ่งที่มันทำได้ คือการติดต่อสื่อสารแบบจุดต่อจุด (Point-to-Point) แต่บลูทูธสามารถทำได้มากกว่านั้น โดยการสร้างเป็นเครือข่ายเล็กๆขึ้นมา ที่เรียกกันว่า พิโคเน็ต

พิโคเน็ตเป็นกลุ่มของอุปกรณ์บลูทูธที่อยู่ในระยะที่สามารถทำการติดต่อสื่อสารกันได้เพื่อทำการแลกเปลี่ยนข้อมูลซึ่งกันและกัน พิโคเน็ตถูกสร้างขึ้นโดยอุปกรณ์ที่เป็นมาสเตอร์ ซึ่งเป็นตัวที่ควบคุมดูแลอุปกรณ์ต่างๆที่ต่ออยู่ในพิโคเน็ตด้วยกันอีก1 – 7 ตัวที่เรียกกันว่า สเลฟ สเลฟทุกตัวจะปรับเวลาเข้าหาตัวมาสเตอร์เพื่อให้สามารถติดต่อสื่อสารกันได้

รูปที่ 3-4 แสดงให้เห็นถึงรูปแบบการเชื่อมต่อแบบจุดต่อจุดระหว่างอุปกรณ์ที่เป็นมาสเตอร์และสเลฟเพื่อที่จะสร้างขึ้นมาเป็นพิโคเน็ต ในรูปที่ 3 แสดงให้เห็นบางสิ่งที่ต่างกันออกไป จากรูปมีมาสเตอร์หนึ่งตัวที่ทำงานกับสเลฟสามตัว แต่ตัวมาสเตอร์นั้นก็ทำหน้าที่เป็นสเลฟให้กับอุปกรณ์ที่เป็นมาสเตอร์อีกตัวหนึ่ง รูปแบบนี้เรียกกันว่า สแคตเตอร์เน็ต เนื่องจากบลูทูธถูกจำกัดให้มีอุปกรณ์ที่เป็นสเลฟในหนึ่งพิโคเน็ตได้ไม่เกิน 7 ตัว ดังนั้นจึงนำพิโคเน็ตแต่ละตัวมาต่อกันเพื่อให้คลอบคลุมการใช้งานที่กว้างมากขึ้น



รูปที่ 3-4 พิโคเน็ตและสแคตเตอร์เน็ต

ขณะที่อุปกรณ์เคลื่อนที่ไปยังที่ต่างๆ มันอาจจะหลุดออกไปจากระยะเครือข่ายได้ทำให้ขาดการติดต่อกับพิโคเน็ตเดิม ดังนั้นแต่ละการเชื่อมต่อจะมีเวลาที่กำหนดไว้ตรวจสอบว่าอุปกรณ์ยังเชื่อมต่ออยู่ในระบบหรือไม่

จากความสามารถของอุปกรณ์บลูทูธที่สามารถนำไปใช้งานแบบเคลื่อนที่ได้ ทำให้บลูทูธพิโคเน็ตแตกต่างไปจากเครือข่ายท้องถิ่นอื่นๆทั่วไป เพราะความยืดหยุ่นที่มากขึ้นนั่นเอง อุปกรณ์ที่อยู่ในเครือข่ายบลูทูธสามารถที่จะเปลี่ยนไปมาได้ตลอดเวลา ซึ่งเรียกเครือข่ายที่มีลักษณะเช่นนี้ว่าเป็น เครือข่ายแบบแอดฮอค

3.1.3 การเชื่อมต่อทางกายภาพ

การเชื่อมต่อถูกสร้างขึ้นระหว่างอุปกรณ์ที่ทำหน้าที่เป็นมาสเตอร์ และอุปกรณ์ที่ทำหน้าที่เป็นสเลฟ เมื่อการเชื่อมต่อถูกสร้างขึ้นแล้ว จะมีชนิดของข้อมูลแพ็กเกตพื้นฐานสองชนิดที่จะใช้ในการส่งข้อมูล คือ ACL และ SCO

- ACL

การเชื่อมต่อแบบ ACL นี้จะเกิดขึ้นทันทีเมื่อมาสเตอร์และสเลฟทำการติดต่อกัน มาสเตอร์สามารถมีจำนวนของการเชื่อมต่อที่เป็น ACL กับสเลฟได้หลายตัวได้ แต่ในเวลาหนึ่งๆสามารถมีเพียงการเชื่อมต่อเดียวระหว่างอุปกรณ์สองตัว ดังที่กล่าวมาแล้วว่ามาสเตอร์ไม้ได้ทำการติดต่อกับสเลฟเพียงตัวเดียวเหมือนกับการติดต่อทั่วไปเท่านั้น ดังนั้นวิธีการของแพ็กเกตสวิตชิ่งจึงนำมาใช้ในการรับส่งข้อมูลเพื่อให้สามารถติดต่อกับอุปกรณ์หลายๆตัว โดยเสมือนว่าสามารถติดต่อกับอุปกรณ์เหล่านั้นได้หลายๆตัวพร้อมกัน ในการที่จะเลือกว่าสเลฟตัวใดจะรับหรือจะส่งข้อมูลขึ้นอยู่กับมาสเตอร์



รูปที่ 3-5 การเชื่อมต่อแบบ ACL

การเชื่อมต่อแบบ ACL นั้นจะส่งข้อมูลไปและรับจากเลเยอร์ L2CAP หรือ LMP การปรับแต่งค่าและการควบคุมการเชื่อมต่อจะถูกทำโดย LM และคำสั่งที่ใช้ในการควบคุมข้อมูลจะถูกส่งไปยังเบสแบนในรูปแบบของแพ็กเกต LMP

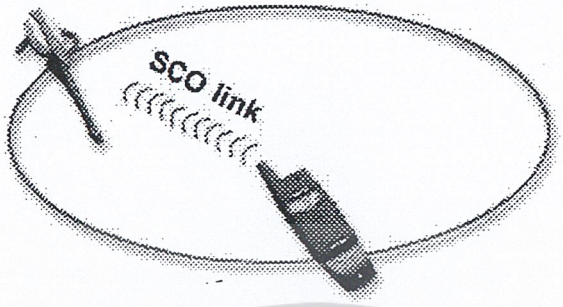
สเลฟจะตอบกลับด้วยแพ็กเกต ACL ในสล็อตของสเลฟ-มาสเตอร์ ถ้าแพ็กเกตที่ส่งมาจากมาสเตอร์ก่อนหน้านี้ระบุแอดเดรสของสเลฟ ในกรณีที่สเลฟไม่สามารถถอดรหัสส่วนหัวแพ็กเกตที่ระบุแอดเดรสของสเลฟได้นั้น สเลฟจะไม่สามารถยืนยันแอดเดรสที่ส่งมาได้ว่าเป็นของตนหรือไม่ ทำให้สเลฟไม่ตอบกลับแพ็กเกตนั้นไป

การบรอดแคสแพ็กเกตจะเป็นการส่งแพ็กเกตแบบ ACL ซึ่งไม่ได้มีการระบุแอดเดรสลงไป ทำให้สเลฟทุกตัวได้รับข้อมูลในแพ็กเกตนั้น

- SCO

การเชื่อมต่อแบบ SCO นี้เป็นลักษณะการเชื่อมต่อแบบเซอร์กิตสวิตชิ่ง ซึ่งใช้ในการรับส่งข้อมูลที่เป็นเสียง มาสเตอร์สามารถส่งข้อมูลที่เป็นเสียงโดยรองรับได้ถึงสามการเชื่อมต่อแบบ SCO ไปยังสเลฟ

ตัวเดียวกันหรือคนละตัวก็ได้ ในทางกลับกัน สเลฟสามารถส่งข้อมูลที่เป็นเสียงโดยรองรับได้ถึงสามการเชื่อมต่อไปยังมาสเตอร์ตัวเดียวกัน แพ็กเกตแบบ SCO นี้จะไม่มีการส่งแพ็กเกตซ้ำไปใหม่



รูปที่ 3-6 การเชื่อมต่อแบบ SCO

3.1.4 โครงสร้างแพ็กเกตบลูทูธ

แพ็กเกตที่ใช้ในการติดต่อสื่อสารระหว่างอุปกรณ์ไม่ว่าจะเป็นการติดต่อแบบ ACL หรือ SCO นั้นจะมีโครงสร้างที่แตกต่างกันออกไป โครงสร้างแพ็กเกตของบลูทูธประกอบด้วย 3 ส่วน คือ ส่วนที่ใช้ในการเข้าถึง, ส่วนหัวข้อมูล และส่วนข้อมูล ในส่วนที่ใช้ในการเข้าถึงนี้จะใช้ค้นหาสถานะของแพ็กเกตและกำหนดแอดเดรสของอุปกรณ์ปลายทางให้กับแพ็กเกตนั้น ตัวอย่างเช่น สเลฟตรวจพบสถานะของแพ็กเกตโดยเปรียบเทียบกับส่วนที่ใช้ในการเข้าถึงกับส่วนที่คัดลอกมาจากส่วนที่ใช้ในการเข้าถึงของมาสเตอร์ ส่วนหัวข้อมูลจะเป็นส่วนที่ควบคุมข้อมูลข่าวสารของแพ็กเกตและการเชื่อมต่อ เช่น แอดเดรสของสเลฟ และส่วนสุดท้าย คือ ส่วนข้อมูล เป็นส่วนที่บรรจุข้อมูลข่าวสารจริงๆที่ต้องการจะส่งไป

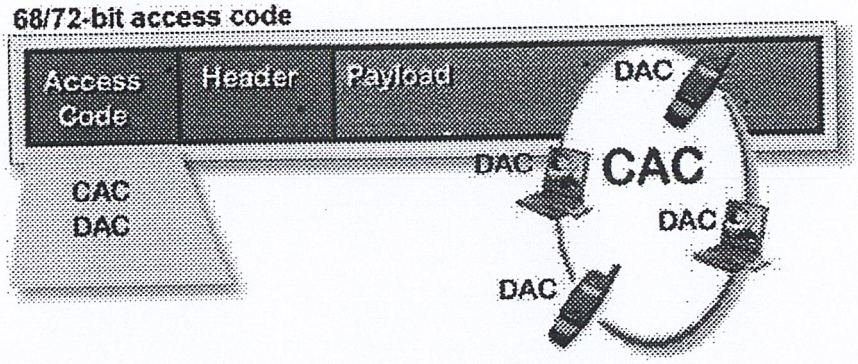


รูปที่ 3-7 โครงสร้างแพ็กเกตของบลูทูธ

3.1.4.1 ส่วนที่ใช้ในการเข้าถึง

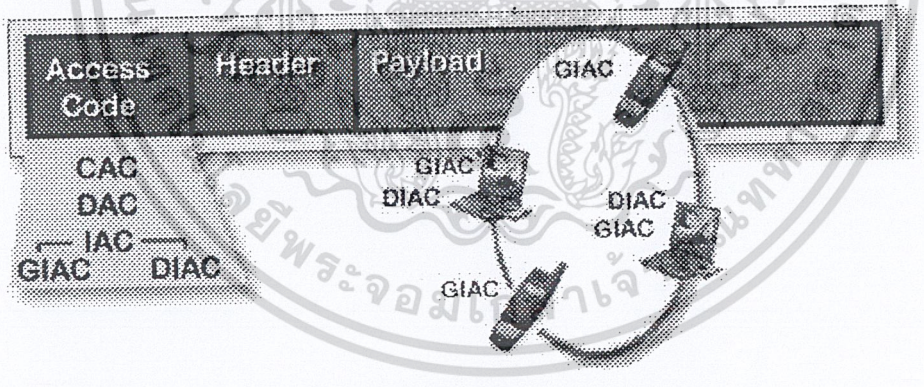
ส่วนที่ใช้ในการเชื่อมต่อสามารถแบ่งออกเป็น 4 ประเภทคือ

- ส่วนที่ใช้ในการเข้าถึงแชนแนล (CAC : Channel Access Code) ส่วนที่ใช้ในการเข้าถึงแชนแนลนี้ได้รับมาจาก LAP ของมาสเตอร์ และจะถูกใช้ในอุปกรณ์ทุกตัวใน พิโคเน็ต ระหว่างที่มีการแลกเปลี่ยนข้อมูลหรือในระหว่างที่มีการติดต่อสื่อสาร



รูปที่ 3-8 ส่วนที่ใช้ในการเข้าถึง CAC และDAC

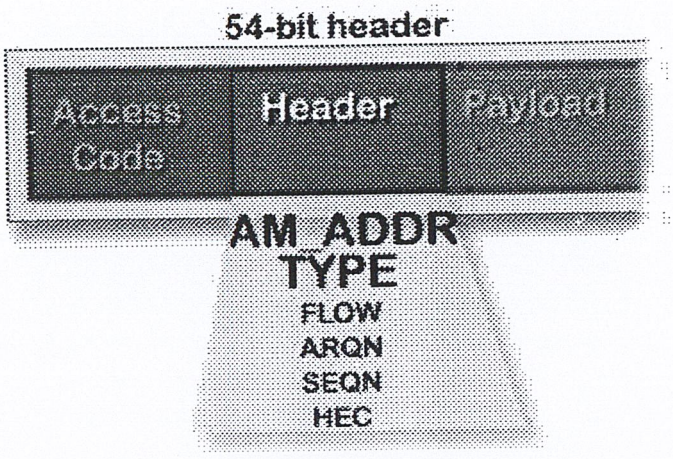
- ส่วนที่ใช้ในการเข้าถึงอุปกรณ์ (DAC : Device Access Code)ส่วนที่ใช้ในการเข้าถึงอุปกรณ์ได้รับมาจาก LAP ของอุปกรณ์ที่ระบุไว้ จะถูกใช้เมื่อทำการเพจจิงไปยังอุปกรณ์ที่ระบุไว้และอุปกรณ์ตัวนั้นจะใช้ในกระบวนการทำเพจสแกนขณะที่กำลังรับฟังว่ามีอุปกรณ์ตัวใดที่ทำการเพจมายังอุปกรณ์ของตน
- ส่วนที่ใช้ในการเข้าถึงทั่วไป (GIAC : General Inquiry Access Code) ส่วนที่ใช้ในการเข้าถึงทั่วไปถูกใช้ในอุปกรณ์ทุกตัวที่ทำกระบวนการอินไควรี่
- ส่วนที่อุทิศให้ในการเข้าถึง (DIAC : Dedicated Inquiry Access Code) ในส่วนนี้จะมีช่วงของเลข DIAC ที่สงวนไว้ให้กับอุปกรณ์ในการทำกระบวนการอินไควรี่



รูปที่ 3-9 ส่วนที่ใช้ในการเข้าถึง IAC

3.1.4.2 ส่วนหัวข้อมูล

ส่วนหัวข้อมูลประกอบไปด้วยข้อมูลที่ใช้ในการควบคุมการทำงานของแพ็กเกต ส่วนหัวข้อมูลนี้จะมีข้อมูล 18 บิตซึ่งจะป้องกันความผิดพลาดโดยใช้ FEC (Forward Error Correction) ส่วนประกอบของส่วนหัวข้อมูลคือ



รูปที่ 3-10 ส่วนหัวของข้อมูล

- AM_ADDR

ในช่วงของการทำกระบวนการเพจจิง มาสเตอร์จะกำหนดแอดเดรสของสมาชิกที่กำลังทำงานอยู่ (Active Member Address) ให้กับอุปกรณ์ต่างๆที่เป็นสเลฟ หลังจากนั้นจึงสร้างการเชื่อมต่อและติดต่อไปยังอุปกรณ์ที่เป็นสเลฟ AM_ADDR ประกอบไปด้วยเลข 3 บิต ซึ่งเพียงพอสำหรับสเลฟ 7 ตัว โดยเลข 000 หมายถึงการบรอดแคส ดังนั้นสเลฟทุกตัวจะได้รับข้อมูลข่าวสารในแพ็กเกตนี้

- ชนิดของแพ็กเกต (Packet Type)

ชนิดของแพ็กเกตนี้จะกำหนดว่าข้อมูลที่ใช้ส่งเป็นแพ็กเกตแบบใด (SCO, ACL, NULL, POLL)

- การไหล (Flow)

เป็นแฟลกที่ถูกกำหนดโดยอุปกรณ์ เมื่ออุปกรณ์ไม่สามารถที่จะรับข้อมูลต่อไปได้ เนื่องจากบัฟเฟอร์มีข้อมูลอยู่เต็มแล้ว

- ARQN และ SEQN

แฟลก ARQN ถูกกำหนดโดยอุปกรณ์เพื่อที่จะบอกว่าข้อมูลที่ได้รับมาก่อนหน้านี้เรียบร้อยแล้ว โดยการตรวจเช็คของ CRC อย่างไรก็ตามระหว่างที่ส่ง ARQN กลับไปอาจจะไม่ถึงไปยังอุปกรณ์อีกฝ่ายหนึ่งก็เป็นได้ ดังนั้น ถ้าอุปกรณ์ที่ส่งข้อมูลไปไม่ได้รับ ARQN ตอบกลับมา จะถือว่าเป็น NAK (Negative Acknowledge) และจะส่งข้อมูลมาใหม่อีกครั้ง ในแต่ละครั้งที่แพ็กเกตถูกส่งออกไปนั้น แฟลก SEQN จะกลับค่าตัวมันเอง

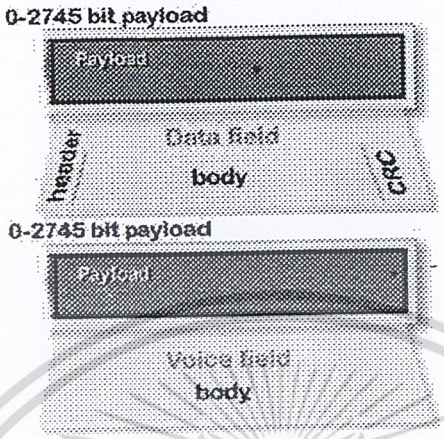
- ส่วนที่เช็คส่วนหัวข้อมูล HEC (Header Error Check)

HEC จะเป็นการทำ CRC ของส่วนหัวข้อมูล HEC จะช่วยให้ผู้ใช้สามารถที่จะละเลยข้อมูลที่ส่งมาหลังจากนี้ เนื่องจากเกิดความผิดพลาดในส่วนของ HEC ได้

3.1.4.3 ส่วนข้อมูล

ส่วนของข้อมูลสามารถแบ่งออกได้เป็น 3 ส่วน คือ

- ส่วนหัวข้อมูล
- ข้อมูล
- CRC



รูปที่ 3-11 ส่วนข้อมูล

3.1.5 ชนิดของแพ็กเกต

ในตารางที่ 3-1 นี้แสดงถึงแพ็กเกตชนิดต่างๆที่ใช้ในบลูทูธ

Segment	Timeslots	SCO Link	ACL Link
1	1 1 1 1	NULL POLL FHS DM1	NULL POLL FHS DM1
2	1 1 1 1	HV1 HV2 HV3 DV	undefined undefined undefined undefined
3	3 3	undefined	DM3 DH3
4	5 5	undefined	DM5 DH5

ตารางที่ 3-1 ชนิดของแพ็กเกตที่ใช้ในบลูทูธ

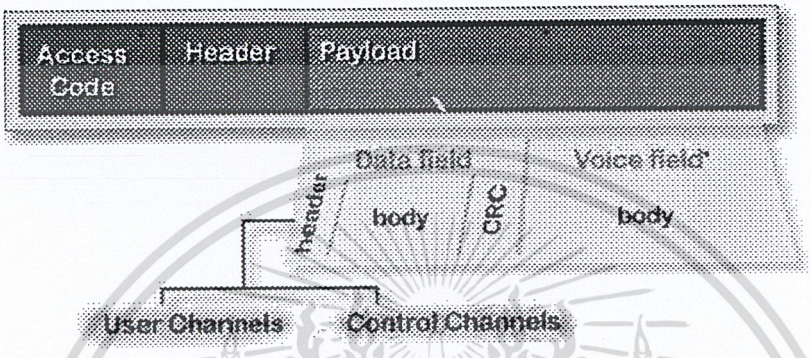
เป็นหน้าที่ของส่วนจัดการการเชื่อมต่อที่จะคอยตรวจสอบดูคุณภาพและความน่าเชื่อถือของการเชื่อมต่อ และตัดสินใจว่าควรจะส่งแพ็กเกตชนิดใดกับข้อมูลแบบใดในช่วงเวลาต่างๆ อุปกรณ์บางชนิดอาจไม่รองรับแพ็กเกตทุกชนิด ดังนั้น ช่วงเวลาในการติดตั้งการเชื่อมต่อเพื่อเจรจาต่อรองจึงเป็นสิ่งสำคัญ

การส่งแบบสมมาตรจะเกิดขึ้นเมื่อส่งข้อมูลที่เป็นแพ็กเกตเดียวกันทั้ง 2 ฝั่ง อย่างไรก็ตามในโปรแกรมประยุกต์บางตัวจำเป็นต้องส่งข้อมูลแบบไม่สมมาตร ซึ่งมีความเร็วสูงสุดในการรับส่งข้อมูลอยู่ที่

723.2 กิโลบิตต่อวินาทีซึ่งสามารถทำได้โดยการส่งแบบห้าสล็อตแพ็กเกตในทิศทางเดียว และรับแพ็กเกตกลับในทิศทางตรงกันข้ามที่ความเร็ว 57.6 กิโลบิตต่อวินาทีและถ้าเป็นการส่งแบบสมมาตร จะส่งได้ที่ความเร็ว 433.9 กิโลบิตต่อวินาทีโดยใช้แพ็กเกตแบบ DH5 ทั้งสองทิศทาง

3.1.6 ลอจิคัลแชนแนล

ในรายละเอียดของบลูทูธเวอร์ชัน 1.0 ได้กำหนดลอจิคัลแชนแนลไว้ 5 ประเภทด้วยกัน คือ



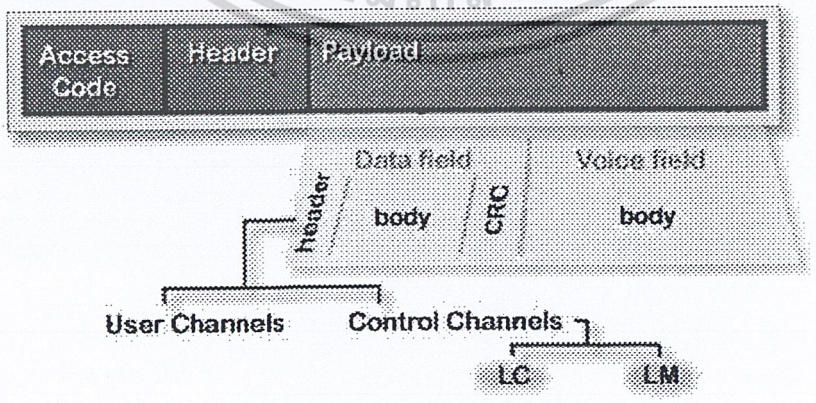
รูปที่ 3-12 ลอจิคัลแชนแนล

- ส่วนควบคุมการเชื่อมต่อ

จะถูกส่งไปกับส่วนหัวของแพ็กเกต ประกอบไปด้วย ARQ, SEQ และส่วนที่ใช้ในการควบคุมข้อมูลอื่นๆ ข้อมูลในส่วนนี้เป็นสิ่งจำเป็นเพื่อใช้ในการควบคุมและรักษาการเชื่อมต่อเอาไว้

- ส่วนจัดการการเชื่อมต่อ

จะถูกส่งไปกับส่วนที่อุทิศให้ในการเข้าถึงข้อมูลและประกอบไปด้วยส่วนที่ใช้ควบคุมส่วนจัดการการเชื่อมต่อซึ่งจะถูกแลกเปลี่ยนข้อมูลระหว่าง LM สองตัว



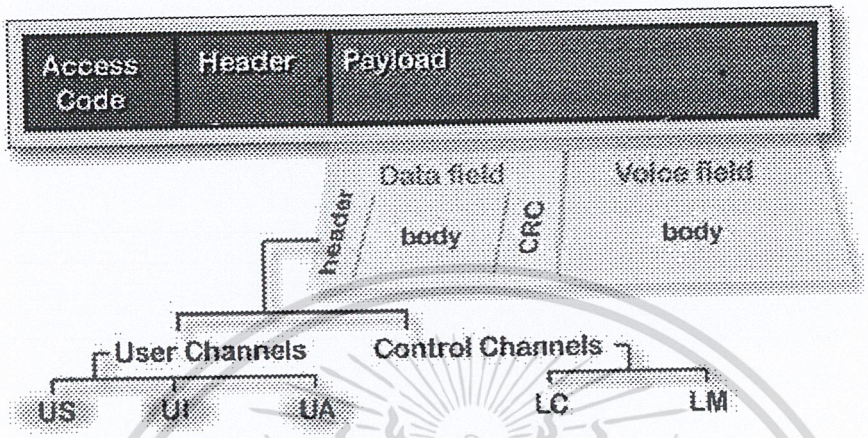
รูปที่ 3-13 ส่วนควบคุมแชนแนล

- UA / UI (User Asynchronous หรือ User Isochronous data)

จะถูกส่งไปกับข้อมูล ACL และประกอบด้วยข้อมูลของผู้ใช้ในระดับ L2CAP

- US (User Synchronous data)

การทำงานในส่วนนี้จะถูกซ่อนโดยส่งผ่านไปกับข้อมูลในแชนแนล SCO



รูปที่ 3-14 ส่วนแชนแนลของผู้ใช้

3.1.7 สรุปการทำงานของเบสแบน

เบสแบนมีหน้าที่รับผิดชอบในการเข้ารหัสและถอดรหัสแชนแนล ควบคุมเวลาการทำงานในระดับล่างและจัดการเกี่ยวกับการเชื่อมต่อในการส่งข้อมูลแต่ละแพ็กเกต โดยที่จะเพิ่มแอดเดรสและส่วนที่ควบคุมการเชื่อมต่อเข้าไปยังข้อมูลที่จะส่ง ซึ่งจะทำให้สามารถตรวจสอบและแก้ไขข้อผิดพลาดในกรณีที่ตรวจพบได้

อุปกรณ์จะมีโหมดการทำงานอยู่ 2 โหมดคือ มาสเตอร์และสเลฟ ซึ่งติดต่อสื่อสารกันภายในเครือข่ายที่เรียกว่าพิโคเน็ต ประกอบไปด้วยอุปกรณ์ที่เป็นสเลฟมากสุด 7 ตัวที่ถูกควบคุมการทำงานโดยอุปกรณ์ที่เป็นมาสเตอร์ ข้อมูลที่ส่งออกไปจะมีอยู่ 2 ชนิด คือ SCO และ ACL ซึ่งแต่ละชนิดก็จะมีชนิดของแพ็กเกตที่แตกต่างกับออกไป

บลูทูธรับประกันว่าอุปกรณ์ทุกตัวสามารถที่จะติดต่อสื่อสารกับอุปกรณ์อื่นๆได้ โดยการส่งสัญญาณติดต่อไปยังมาสเตอร์เป็นระยะๆ เพื่อยืนยันว่าตนเองยังอยู่ในระบบ

3.2 ส่วนควบคุมการเชื่อมต่อ (LC : Link Controller)

ในเลเยอร์ของส่วนการควบคุมการเชื่อมต่อนี้จะดูแลในส่วนของการทำงานในระดับที่สูงขึ้นกว่าเบสแบน เช่น กระบวนการอินไควรี่, กระบวนการเพจจิง, การจัดการกับหลายๆการเชื่อมต่อกับอุปกรณ์หลายๆตัวหรือพิโคเน็ตหลายๆเครือข่าย

3.2.1 สถานะของส่วนควบคุมการเชื่อมต่อ

ในขณะใดขณะหนึ่งนั้น อุปกรณ์บลูทูธจะอยู่ในสถานะใดสถานะหนึ่งจากหลายๆสถานะที่มีอยู่ โดยมีสถานะต่างๆดังนี้

- แสตนบาย

ในสถานะนี้ อุปกรณ์จะอยู่ในโหมดไม่ทำงาน ไม่มีการส่งข้อมูลระหว่างอุปกรณ์ และคลื่นวิทยุไม่ได้ถูกเปิดใช้งาน ดังนั้นอุปกรณ์ที่อยู่ในสถานะนี้จะไม่สามารถตรวจสอบข้อมูลการเข้าถึงใดๆได้ สถานะนี้ถูกใช้ในกรณีที่อุปกรณ์อยู่ในช่วงของการประหยัดพลังงาน

- อินไควรี่

อินไควรี่เป็นกระบวนการที่อุปกรณ์มีความพยายามที่จะค้นหาอุปกรณ์ตัวอื่นๆที่อยู่ในระยะเครือข่าย ในชั้นของเบสแบนนั้นจะจบอยู่ที่โพรโตคอลการค้นหาการบริการ หลังจากนั้นมันจะลิสต์รายชื่อของอุปกรณ์ขึ้นมาเพื่อให้เลือกว่าอุปกรณ์ใดที่ต้องการจะติดต่อด้วย ระหว่างกระบวนการของการอินไควรี่นี้ อุปกรณ์ที่ถูกอินไควจะส่งแพ็กเกต FHS ไปยังอุปกรณ์ที่ทำการอินไคว ในแพ็กเกต FHS นี้จะมีข้อมูลที่สำคัญที่จะทำให้อุปกรณ์ที่อินไควนั้นสามารถที่จะสร้างการเชื่อมต่อขึ้นมาได้ เช่น CLKN ซึ่งใช้ในการควบคุมการกระโดดของความถี่, BD_ADDR ใช้ในการควบคุมการกระโดดของความถี่และรูปแบบของการเข้าถึงข้อมูล และข้อมูลพาริตี BCH ใช้ในการสร้างส่วนของการเข้าถึงข้อมูล

- อินไควรี่สแกน

อินไควรี่สแกนเป็นกระบวนการครึ่งที่เหลือของกระบวนการอินไควรี่ ถึงแม้ว่าการสแกนจะเป็นความสามารถเสริมและขึ้นอยู่กับโปรแกรมประยุกต์ที่ใช้งาน แต่อุปกรณ์ส่วนใหญ่มักเข้าสู้การทำงานในโหมดนี้เพื่อให้ตนเองถูกค้นพบโดยอุปกรณ์อื่น อุปกรณ์ที่อยู่สถานะนี้จะคอยฟังในช่วงเวลาที่กำหนดเพื่อคอยรับแพ็กเกตอินไควรี่ที่ใช้ส่วนที่ใช้ในการเข้าถึงทั่วไปหรือส่วนที่อุทิศให้ในการเข้าถึง (GIAC, DIAC) เมื่ออุปกรณ์ได้รับข้อความอินไควรี่ที่สมบูรณ์แล้ว ก็จะเข้าสู่โหมดการตอบสนองต่ออินไควรี่ ซึ่งเป็นสถานะย่อยและตอบกลับโดยใช้ข้อมูลจากแพ็กเกต FHS ที่ได้รับมา สถานะการอินไควรี่และการอินไควรี่สแกนจะใช้ประโยชน์จากการกระโดดของความถี่โดยใช้การกระโดดแบบรวดเร็วกับกระบวนการอินไควรี่ และการกระโดดแบบช้ากับกระบวนการอินไควรี่สแกน ซึ่งออกแบบมาเพื่อลดเวลาที่ต้องเสียไปก่อนที่ความถี่ของอุปกรณ์ทั้งสองจะมาเจอกัน

- เพจ

ในการสร้างการเชื่อมต่อขึ้นมานั้น อุปกรณ์ที่เป็นมาสเตอร์จะถูกสั่งโดยโปรแกรมประยุกต์ให้ทำกระบวนการเพจจิง อุปกรณ์ที่เป็นมาสเตอร์จะต้องเข้าสู่สถานะของการเพจก่อน ซึ่งจะส่งข้อ

ความเพจจิงไปยังอุปกรณ์ที่ระบุไว้ โดยใช้ข้อมูลที่ได้รับมาจากขั้นตอนการทำอินไควรี่ สเลฟจะตอบกลับข้อความเพจจิงและมาสเตอร์จะเข้าสู่สถานะการตอบสนองต่อมาสเตอร์ ซึ่งเป็นสถานะย่อยและตอบกลับด้วยแพ็กเกต FHS

- เพจสแกน

เหมือนกับสถานะอินไควรี่สแกน อุปกรณ์ต้องเข้าสู่สถานะเพจสแกนเป็นช่วงๆเพื่ออนุญาตให้อุปกรณ์ที่ทำการเพจจิงสร้างการเชื่อมต่อขึ้นมาได้ เมื่ออุปกรณ์อยู่ในสถานะเพจสแกนและได้รับแพ็กเกตที่เพจจิงแล้ว ก็จะเข้าสู่โหมดการตอบสนองต่อสเลฟ ซึ่งเป็นสถานะย่อยและจะตอบกลับและคอยแพ็กเกต FHS เมื่อได้รับแพ็กเกต FHS แล้วก็จะทำการอัพเดตข้อมูลก่อนที่จะเข้าสู่สถานะการเชื่อมต่อ ดังเช่นกระบวนการอินไควรี่ กระบวนการเพจจิงจะใช้การกระโดดของความถี่แบบพิเศษ คือ เร็วสำหรับการทำการเพจ และช้าสำหรับการทำเพจสแกน ซึ่งออกแบบมาเพื่อลดเวลาที่ต้องเสียไปก่อนที่ความถี่ของอุปกรณ์ทั้งสองจะมาเจอกัน

ในการทำการเพจสแกนนี้ สามารถที่จะเข้าสู้กระบวนการเพจได้เลยโดยที่ไม่ต้องผ่านขั้นตอนของกระบวนการอินไควรี่มาก่อนได้เลย ถ้ารู้แอดเดรสของอุปกรณ์อยู่แล้ว

- **การเชื่อมต่อ – ทำงาน**

ในสถานะการเชื่อมต่อนี้ สเลฟจะสวิตไปยังคล็อกของมาสเตอร์และกระโดดไปยังความถี่ที่ตัวมาสเตอร์ใช้ มาสเตอร์จะส่งแพ็กเกต POLL มาเพื่อพิสูจน์ว่าการเชื่อมต่อได้เกิดขึ้นอย่างเสร็จสมบูรณ์แล้ว จากนั้นสเลฟจะทำการตอบไปยังมาสเตอร์ด้วยแพ็กเกตชนิดใดก็ได้ โดยส่วนใหญ่แล้วจะเป็นแพ็กเกต NULL แฟลก ARQN ของอุปกรณ์ทั้งสองตัวจะถูกเซ็ตให้เป็น NAK แต่หลังจากนั้นเมื่อการส่งข้อมูลครั้งต่อไปเริ่มขึ้น ตัวมาสเตอร์จะทำการเซ็ต ARQ และ SEQ เพื่อใช้งานจริง ถ้าการเชื่อมต่อไม่ประสบผลสำเร็จ หลังจากเวลาที่เหมาะสมผ่านไป อุปกรณ์ทั้งคู่จะเข้าสู่กระบวนการเพจและเพจสแกนอีกครั้ง โดยกระบวนการทั้งหมดจะเริ่มขึ้นใหม่

เมื่อสเลฟได้รับแพ็กเกตมาจะทำการตรวจสอบกับส่วนหัวของข้อมูลว่า AM_ADDR ตรงกับแอดเดรสของมันหรือไม่ ถ้าไม่ตรงก็จะไม่สนใจข้อมูลส่วนที่เหลืออีกต่อไป เนื่องจากชนิดของแพ็กเกตที่อยู่ในส่วนหัวจะเป็นตัวบอกว่าแพ็กเกตที่ส่งมานี้จะมีข้อมูลต่ออีกกี่แพ็กเกต สเลฟจะใช้ประโยชน์จากข่าวสารที่ได้รับมานี้เพื่อเข้าสู่โหมดประหยัดพลังงานขณะที่มาสเตอร์กำลังส่งข้อมูลให้กับอุปกรณ์อื่นๆอยู่

ในช่วงของสถานะการเชื่อมต่อนี้ ข้อมูลต่างๆจะถูกแลกเปลี่ยนซึ่งกันและกัน อย่างไรก็ตามอุปกรณ์ที่อยู่ในสถานะนี้อาจจะย้ายไปสู่โหมดประหยัดพลังงานตามที่ได้กล่าวมาแล้ว

- **การเชื่อมต่อ – พัก**

ในโหมดนี้ อุปกรณ์จะหยุดการรับส่งข้อมูลที่เป็น ACL ภายในช่วงเวลาที่กำหนดเพื่อทำให้มีแบนด์วิดธ์ว่างพอที่จะไปทำงานอื่นๆ เช่น การสแกน, การเพจ, การอินไควรี่ หรือแม้กระทั่งเข้าสู่

โหมดประหยัดพลังงาน หลังจากที่เวลาในการหยุดพักหมดลง อุปกรณ์จะทำการเข้าจังหวะและคอยรับฟังการสื่อสารอีกครั้งหนึ่ง

- **การเชื่อมต่อ – สนิฟ**

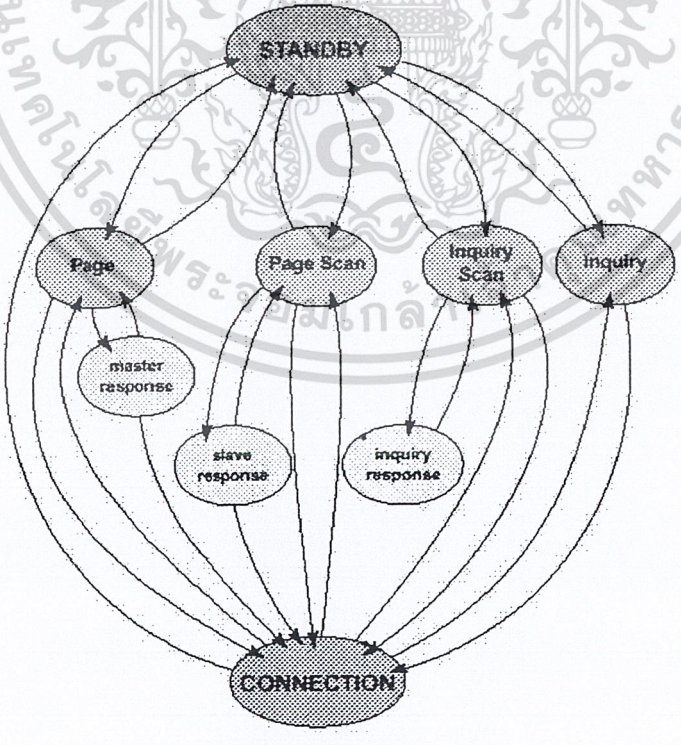
ในโหมดนี้อุปกรณ์ที่เป็นสเลฟจะถูกกำหนดช่วงเวลาใดเวลาหนึ่งเพื่อคอยฟังการสื่อสารที่เกิดขึ้น ถ้ามีการรับแพ็กเกตในช่วงเวลานี้ สเลฟจะคอยฟังจนกระทั่งแพ็กเกตที่มีแอดเดรสของมันหยุดไป

- **การเชื่อมต่อ – หยุดรอ**

ในโหมดนี้สเลฟจะยกเลิก AM_ADDRESS และคอยฟังการสื่อสารเป็นช่วงๆ โดยทั่วไปแล้ว จะเป็นในกรณีที่อุปกรณีเข้าสู่โหมดประหยัดพลังงาน

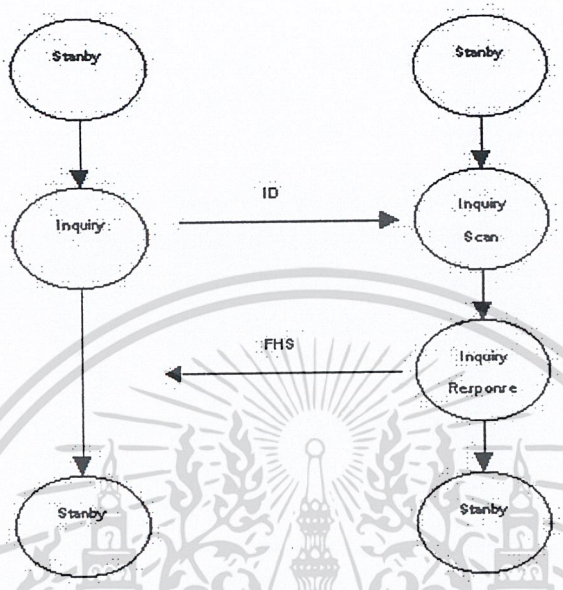
3.2.2 การทำงานของสถานะควบคุมการเชื่อมต่อ

สถานะการทำงานของอุปกรณ์บลูทูธสามารถแสดงได้ดังรูปที่ 3-15 ไดอะแกรมนี้แสดงให้เห็นถึงการทำงานเริ่มจากโหมดแสตนบาย ไปยังการอินไควรี่ แล้วจึงไปยังการเชื่อมต่อ แท้จริงแล้ว ไม่มีวิธีใดที่เป็นเส้นทางตรง ในการที่จะสร้างการเชื่อมต่อนั้น จะต้องเริ่มจากการเพจจิงหรือเพจสแกนก่อน ไดอะแกรมนี้เพียงตั้งใจที่จะแสดงให้เห็นว่าสถานะอินไควรี่สามารถมาได้จากแสตนบายหรือสถานะการเชื่อมต่อก็ได้ ไม่ได้แสดงให้เห็นถึงวิธีการใดที่จะไปถึงสถานะการเชื่อมต่อ

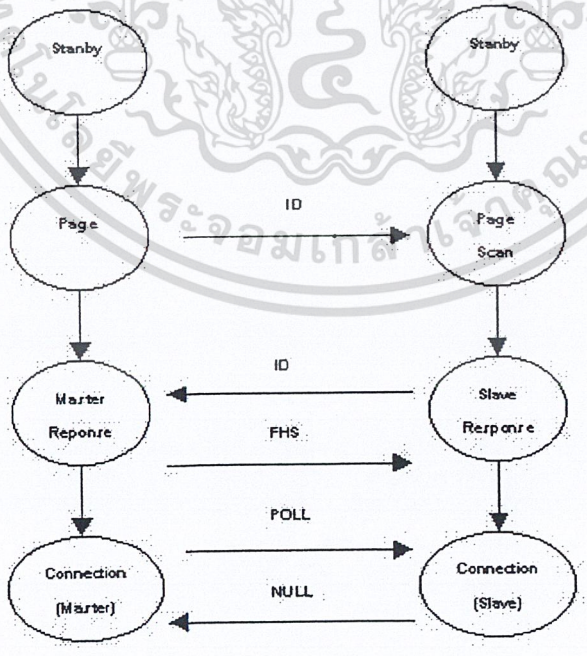


รูปที่ 3-15 ไดอะแกรมแสดงสถานะของส่วนควบคุมการเชื่อมต่อ

อีกรูปแบบหนึ่งที่แสดงให้เห็นถึงการเดินทางของสถานะไดอะแกรมแสดงได้ดังรูปที่ 3-16 และ 3-17 จากรูปนี้จะช่วยให้มีความเข้าใจในการที่อุปกรณีจะสร้างการเชื่อมต่อขึ้นมา อย่างไรก็ตาม สถานะไดอะแกรมอาจมีความซับซ้อนมากกว่านี้ก็เป็นได้



รูปที่ 3-16 ไดอะแกรมแสดงสถานะของกระบวนการอินไคว

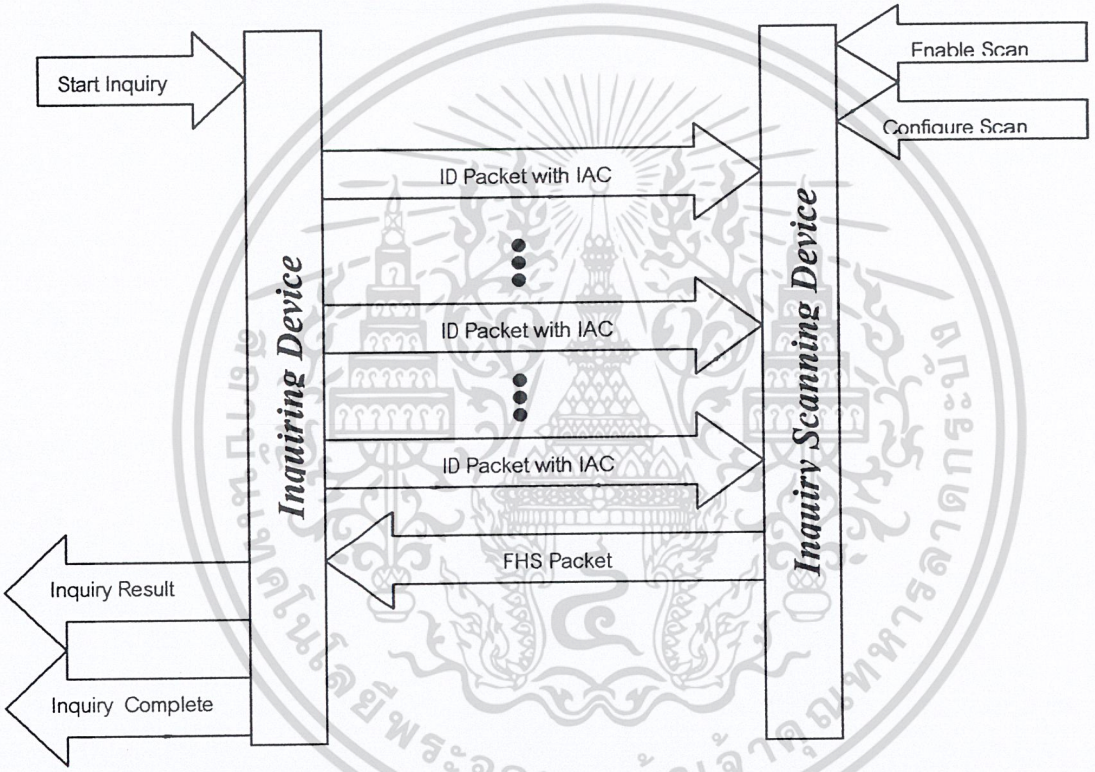


รูปที่ 3-17 ไดอะแกรมแสดงสถานะของกระบวนการเพจจิง

3.2.2.1 การค้นหาอุปกรณ์และการอินไควรี่

สืบเนื่องมาจากลักษณะที่เป็นธรรมชาติของเครือข่ายบลูทูธแบบแอดฮอค ที่ซึ่งอุปกรณ์พกพาต่างๆสามารถที่จะเคลื่อนที่เข้าและออกจากเครือข่ายเมื่อใดก็ได้ มีผลให้รูปร่างของเครือข่ายและอุปกรณ์ที่อยู่ในเครือข่ายมีการเปลี่ยนแปลงอยู่เสมอ ในรายละเอียดบลูทูธเวอร์ชัน 1.0 ได้กำหนดวิธีการในการค้นหาอุปกรณ์ โดยใช้โพรโตคอลการค้นหาการบริการมาช่วย ส่วนควบคุมการเชื่อมต่อใช้กระบวนการอินไควรี่และอินไควรี่สแกนในการค้นหาอุปกรณ์และอีปกรณ์ที่ต้องการถูกค้นพบ ตามลำดับ

รูปที่ 3-18 แสดงให้เห็นถึงข้อความที่มีการแลกเปลี่ยนกันระหว่างอุปกรณีสองตัวที่ทำการอินไควรี่และการอินไควรี่สแกน อุปกรณ์ที่ทำการอินไควรี่จะถามไปว่า "มีใครอยู่บ้างมั้ย?" ในขณะที่การสแกนจะเป็นการคอยรับฟังว่ามีใครที่ทำการอินไควอยู่หรือไม่



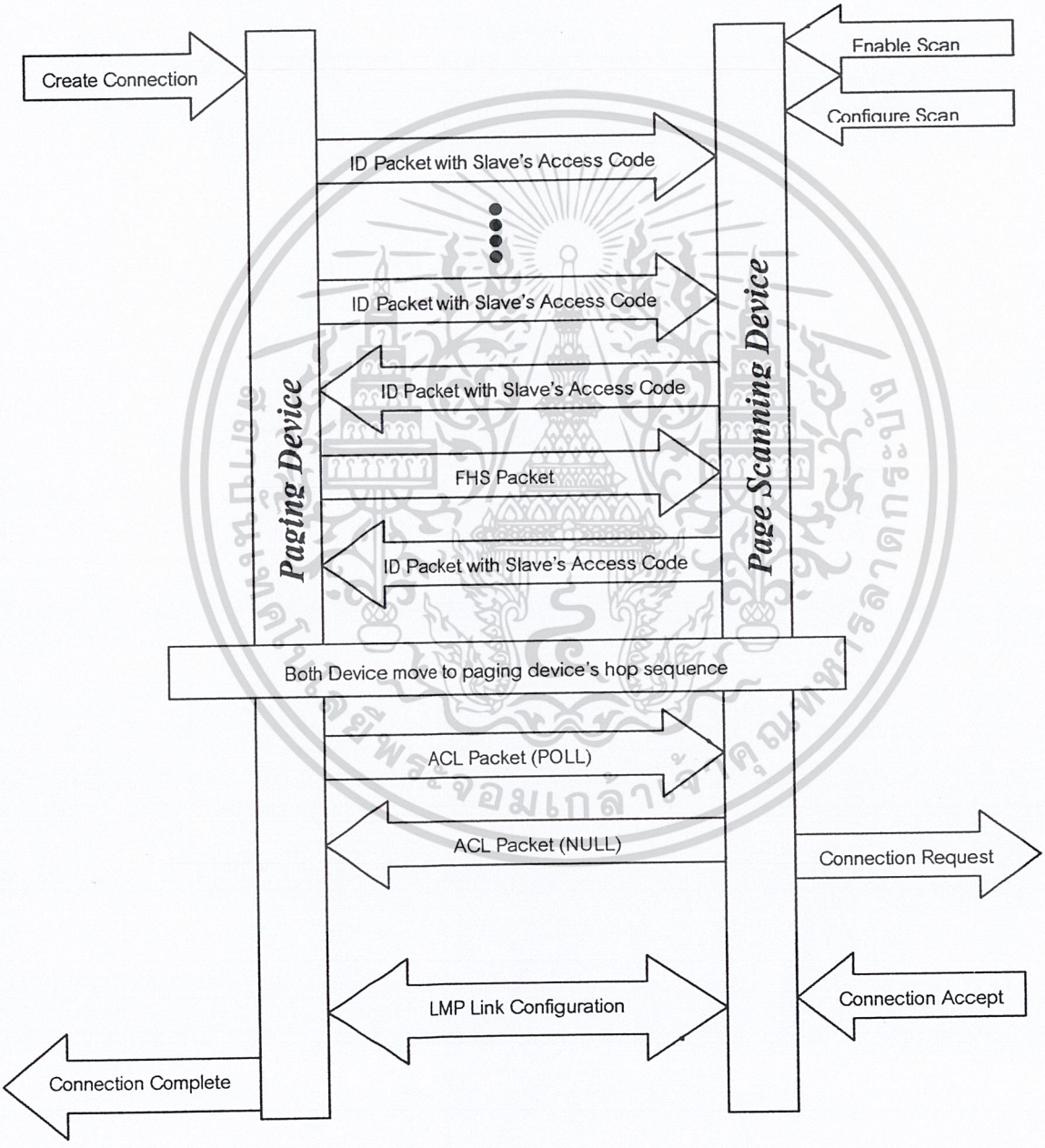
รูปที่ 3-18 ลำดับการส่งข้อความในการอินไควรี่และการอินไควรี่สแกน

อุปกรณ์ที่ทำการอินไควรี่จะส่งแพ็กเกต ID ออกไปซึ่งในนั้นประกอบด้วยส่วนเข้าถึงข้อมูลการอินไคว เมื่ออุปกรณ์ที่ทำการอินไควรี่สแกนได้รับการอินไควรี่แล้วสามารถที่จะตอบกลับในทันที แต่การทำแบบนี้อาจทำให้ผลลัพธ์ที่ตอบกลับไปจากหลายๆอุปกรณ์เกิดการรบกวนซึ่งกันและกันได้ ทำให้อุปกรณ์ที่ส่งอินไควรี่ออกไปไม่ได้รับการตอบกลับมาเลย ดังนั้น จึงมีการกำหนดเวลาที่ใช้ในการเลื่อนออกไปขึ้นมา ซึ่งเป็นค่าที่สุ่มขึ้นมา เมื่ออุปกรณ์ที่ทำการสแกนอยู่ได้รับข้อมูลการอินไควแล้ว แทนที่จะตอบกลับไปในทันที ก็จะรอเป็นเวลาช่วงหนึ่งก่อน แล้วจึงเข้าสู่สถานะสแกนอีกครั้งเพื่อที่จะคอยฟัง ID จากตัวอื่นๆ เมื่อได้รับการอินไควอีก ก็จะตอบกลับด้วยแพ็กเกต FHS ด้วยวิธีนี้ทำให้อุปกรณ์ต่างๆหลายๆตัวสามารถที่จะตอบกลับไปยังอุปกรณีที่ทำการอินไควได้โดยที่ไม่รบกวนกัน

3.2.2.2 การสร้างการเชื่อมต่อและการเพจจิง

ในการที่จะสร้างการเชื่อมต่อ อุปกรณ์ตัวหนึ่งจะต้องเริ่มต้นการเชื่อมต่อโดยการสอบถามไปยังอุปกรณ์ที่ต้องการติดต่อด้วยว่า "คุณต้องการติดต่อกับฉันมั้ย?" ซึ่งนั่นคือกระบวนการเพจจิงนั่นเอง อุปกรณ์อื่นๆที่คอยฟังคำถามนี้อยู่ จะเรียกว่ากำลังทำการเพจสแกน เมื่อสิ้นสุดกระบวนการเพจแล้ว อุปกรณ์ที่ทำการเพจจะเป็นมาสเตอร์ ส่วนอุปกรณ์ที่ทำการเพจสแกนจะเป็นสเลฟ

รูปที่ 3-19 แสดงให้เห็นถึงการรับส่งข้อความระหว่างอุปกรณ์สองตัวที่อยู่ในขั้นตอนของการสร้างการเชื่อมต่อโดยใช้กระบวนการเพจจิง



รูปที่ 3-19 ลำดับการส่งข้อความในการเพจจิงและเพจสแกน

เริ่มจากคำสั่งในการสร้างการเชื่อมต่อทำให้อุปกรณ์เข้าสู่สถานะเพจจิง โดยอุปกรณ์จะส่งชุดของแพ็กเกต ID ที่มีแอดเดรสของผู้รับออกไป ขณะเดียวกันเมื่ออุปกรณ์ที่ทำการเพจสแกนอยู่ได้รับแพ็กเกต ID นั้นแล้วก็จะตอบกลับด้วยแพ็กเกต ID ของตน ต่างกับขั้นตอนการทำอินไควรี่ เนื่องจากการในการเพจนั้นอุปกรณ์ที่จะตอบกลับมาข่อมมีเพียงตัวเดียวเท่านั้น ดังนั้นเมื่ออุปกรณ์ตอบกลับไปด้วยแพ็กเกต ID ของตนแล้วจะดึงค่าต่างๆที่ได้รับมาจากแพ็กเกต ID เพื่อใช้ในการสร้างการเชื่อมต่อขึ้นมา หลังจากนั้นอุปกรณ์ที่ทำการเพจสแกนจะสามารถคำนวณความถี่ที่ต้องใช้ในการติดต่อสื่อสารได้แล้ว ก็จะกระโดดไปใช้ความถี่นั้น การเชื่อมต่อก็จะเริ่มขึ้น อุปกรณ์ที่ทำการเพจจะเป็นมาสเตอร์ ส่วนอุปกรณ์ที่ทำการเพจสแกนจะเป็นสเลฟ

หลังจากนั้นกระโดดไปใช้ความถี่ที่ตกลงกันไว้แล้ว มาสเตอร์จะส่งแพ็กเกต POLL ออกไป เพื่อตรวจสอบว่าสเลฟสามารถรับข้อมูลได้อย่างถูกต้อง เมื่อสเลฟได้รับแพ็กเกตแล้วก็จะตอบกลับด้วยแพ็กเกต ACL ชนิดใดก็ได้ หลังจากที่มีการตรวจสอบเรียบร้อยแล้วแพ็กเกตการปรับแต่งค่า LMP จะถูกส่งถึงกันเพื่อปรับค่าต่างๆให้เรียบร้อยก่อนที่จะเริ่มส่งข้อมูลจริง

3.2.3 สรุปการทำงานของส่วนควบคุมการเชื่อมต่อ

ในเลเยอร์ควบคุมการเชื่อมต่อนี้มีหน้าที่รับผิดชอบในการจัดการเรื่องการค้นหาอุปกรณ์, การสร้างการเชื่อมต่อ และเมื่อเชื่อมต่อได้แล้วก็ดูแลการเชื่อมต่อนั้นให้สามารถติดต่อสื่อสารได้ โดยการทำงานแทนได้ด้วยสถานะการทำงานต่างๆ ซึ่งรับผิดชอบในส่วนต่างๆต่อไปนี้

- โฮสต์ร้องขอการอินไควรี่
- อินไควรี่ถูกส่งไปโดยใช้ลำดับความถี่กระโดดที่ใช้ในการอินไคว
- อุปกรณ์ที่ทำการอินไควรี่สแกนตอบกลับด้วยแพ็กเกต FHS ซึ่งประกอบไปด้วยข้อมูลต่างๆที่จำเป็นในการสร้างการเชื่อมต่อ
- ข้อมูลในแพ็กเกต FHS ถูกส่งกลับไปยังโฮสต์
- โฮสต์ร้องขอการเชื่อมต่อไปยังอุปกรณ์ที่ตอบรับการอินไควรี่
- กระบวนการเพจจิงใช้ในการเริ่มต้นการเชื่อมต่อไปยังอุปกรณ์ที่ต้องการติดต่อด้วย
- ถ้าอุปกรณ์ที่ต้องการติดต่อด้วยทำการเพจสแกนอยู่จะตอบกลับมา
- ถ้าอุปกรณ์ที่ทำการเพจสแกนอยู่ยอมรับการเชื่อมต่อ จะกระโดดไปยังใช้ความถี่ที่มาสเตอร์เป็นผู้กำหนดและติดต่อสื่อสารผ่านลำดับความถี่ที่ใช้ในการกระโดดนั้น

ในส่วนของการควบคุมการเชื่อมต่อนั้นขึ้นอยู่กับแต่ละอุปกรณ์ ถึงแม้ว่ามาสเตอร์จะมีอำนาจในการควบคุมเมื่อมีการเชื่อมต่อขึ้นมาแล้ว แต่มันขึ้นอยู่กับอุปกรณ์แต่ละตัวว่าต้องการจะติดต่อด้วยหรือไม่ ถ้าอุปกรณ์ต้องการที่จะไม่ถูกค้นพบก็ไม่ต้องทำการอินไควรี่สแกน หรือถ้าอุปกรณ์ไม่ต้องการที่จะติดต่อกับอุปกรณ์อื่นๆก็ไม่ต้องทำการเพจสแกน และในขณะที่ทำการเชื่อมต่ออยู่นั้น ก็สามารถที่จะยกเลิกการเชื่อมต่อในทันทีโดยที่ไม่จำเป็นต้องแจ้งให้มาสเตอร์ทราบก่อน

เมื่อการเชื่อมต่อได้ถูกสร้างขึ้นแล้ว มาสเตอร์และสเลฟสามารถที่จะแลกเปลี่ยนหน้าที่การทำงานกันได้ โดยที่สเลฟจะกลายเป็นมาสเตอร์และมาสเตอร์จะกลายเป็นสเลฟ

3.3 ส่วนจัดการการเชื่อมต่อ (LM : Link Manager)

โฮสต์สั่งงานอุปกรณ์บลูทูธผ่านทางคำสั่งที่อยู่ในส่วนเชื่อมต่อการควบคุมโฮสต์ (HCI) แต่ส่วนจัดการการเชื่อมต่อจะเป็นส่วนที่ทำหน้าที่ในการแปลคำสั่งเหล่านี้ไปเป็นการทำงานในระดับของเบสแบน ซึ่งจะจัดการการทำงานต่างๆดังนี้

- นำสเลฟมารวมเข้าไปในพิโคเน็ต และกำหนดเลข AM_ADDR ให้กับอุปกรณ์นั้น
- หยุดการเชื่อมต่อชั่วคราวเพื่อนำสเลฟมารวมเข้าไปในพิโคเน็ต
- ปรับแต่งค่าการเชื่อมต่อ โดยรวมถึงการสลับหน้าที่ระหว่างมาสเตอร์กับสเลฟ
- สร้างการเชื่อมต่อแบบ SCO และ ACL
- นำการเชื่อมต่อไปยังโหมดประหยัดพลังงาน เช่น พัก, สนิฟ หรือหยุดรอ
- ควบคุมโหมดการทดลอง

ส่วนจัดการการเชื่อมต่อบลูทูธนี้จะติดต่อสื่อสารกับส่วนจัดการการเชื่อมต่อที่อยู่ในอุปกรณ์บลูทูธอื่นโดยใช้โพรโตคอลการจัดการการเชื่อมต่อ

รายละเอียดบลูทูธเวอร์ชัน 1.0 ได้กำหนดการแลกเปลี่ยนข้อความใน LMP ระหว่างส่วนจัดการการเชื่อมต่อต่างๆ โดยข้อความเหล่านั้นจะใช้ในการควบคุมการจัดการของพิโคเน็ต, การปรับแต่งค่าการเชื่อมต่อ และฟังก์ชันในเรื่องของความปลอดภัย

3.3.1 หน่วยข้อมูลโพรโตคอล (PDU : Protocol Data Unit)

ส่วนจัดการการเชื่อมต่อจะติดต่อสื่อสารระหว่างอุปกรณ์ต่างๆโดยใช้โพรโตคอลการจัดการการเชื่อมต่อ ซึ่งจะระบุกลุ่มของข้อความที่ในการติดต่อระหว่างส่วนจัดการการเชื่อมต่อ

3.3.2 แชนแนลส่วนจัดการการเชื่อมต่อ

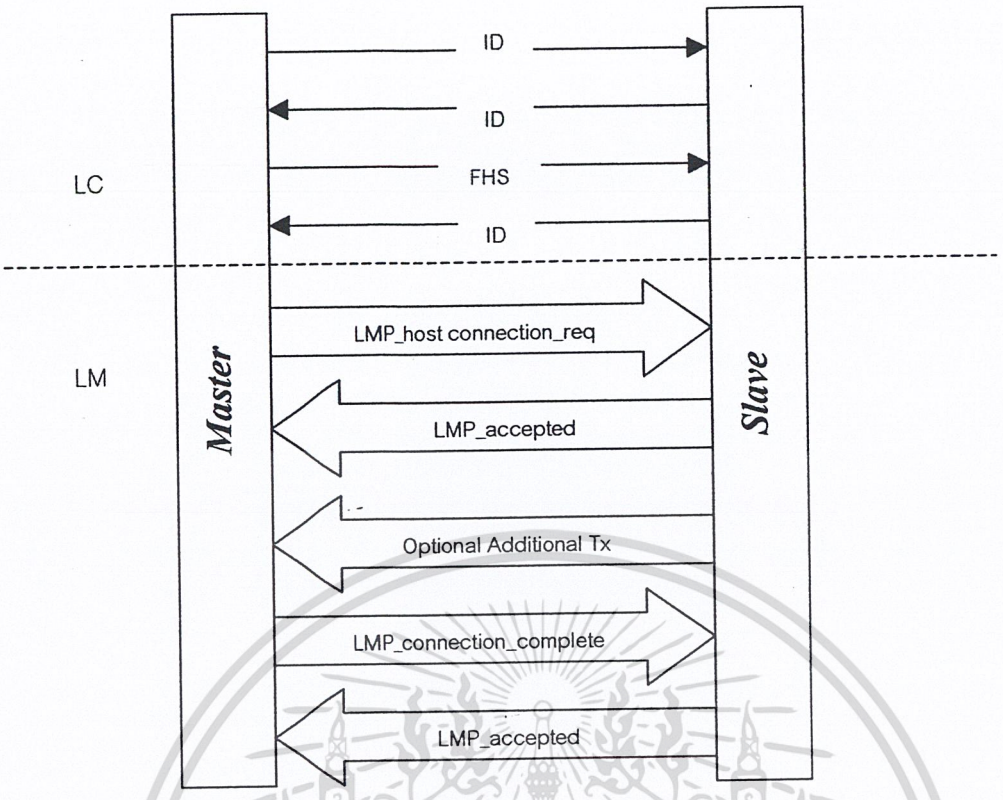
หน่วยข้อมูลโพรโตคอล LMP จะถูกส่งเป็นแพ็กเกตสล็อตเดี่ยวบนส่วนจัดการการเชื่อมต่อที่เป็นลอจิคัลแชนแนล แชนแนลนี้ถูกกำหนดโดยตั้งค่าในลอจิคัวแชนแนลที่อยู่ในส่วนหัวของแพ็กเกตเป็น 11

3.3.3 การเริ่มต้นการเชื่อมต่อ

ส่วนจัดการการเชื่อมต่อมีหน้าที่ในการเริ่มต้นการเชื่อมต่อและคอยดูแลการเชื่อมต่อนั้นให้อุปกรณ์บลูทูธสามารถติดต่อสื่อสารกันได้ ส่วนจัดการการเชื่อมต่อจะเริ่มสร้างการเชื่อมต่อแบบ ACL โดยการควบคุมเบสแบน จากนั้น ข้อความ LMP จะถูกใช้ในการสร้างการเชื่อมต่อ SCO บนการเชื่อมต่อ ACL ที่มีอยู่เดิม

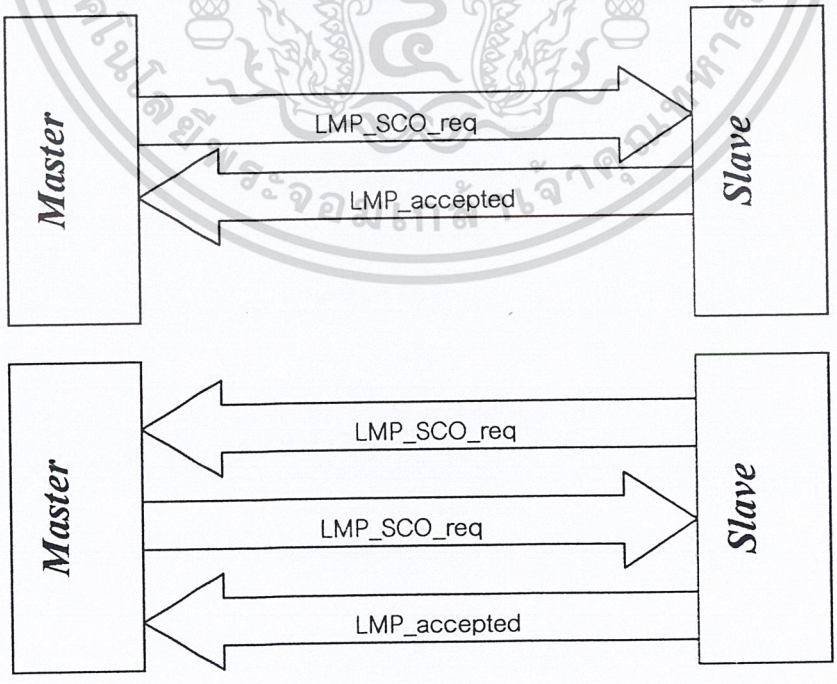
ส่วนจัดการการเชื่อมต่อจะดูแลข้อมูลบนอุปกรณ์ที่เป็นสเลฟซึ่งได้กำหนด AM_ADDR ให้

รูป 3-20 แสดงให้เห็นถึงข้อความที่เกี่ยวข้องในการเริ่มต้นการเชื่อมต่อ ACL ในส่วนควบคุมการเชื่อมต่อจะต้องสร้างการเชื่อมต่อระหว่างอุปกรณ์ก่อนที่ข้อความ LMP จะถูกแลกเปลี่ยน



รูปที่ 3-20 ลำดับข้อความ LMP ที่ใช้ในการเริ่มต้นการเชื่อมต่อแบบ ACL

เมื่อการเชื่อมต่อ ACL ถูกสร้างขึ้นมาได้แล้ว มาสเตอร์หรือสเลฟสามารถที่จะร้องขอการเชื่อมต่อแบบ SCO บนการเชื่อมต่อแบบ ACL ได้ ดังรูปที่ 3-21

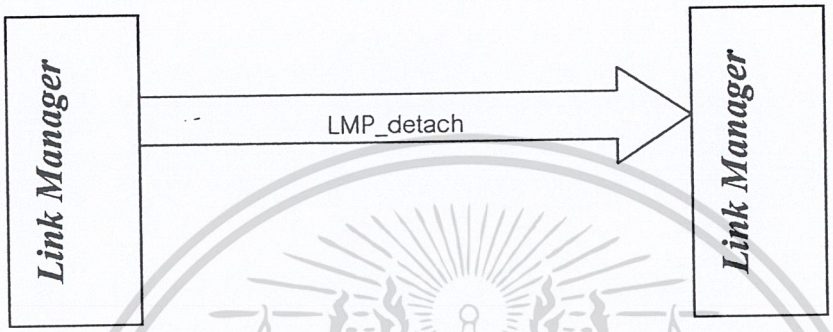


รูปที่ 3-21 ลำดับข้อความ LMP ที่ใช้ในการเริ่มต้นการเชื่อมต่อแบบ SCO

3.3.4 ปิดการเชื่อมต่อ

เวลาใดก็ตามที่มาสเตอร์หรือสเลฟต้องการที่จะปิดการเชื่อมต่อลง จะส่งคำสั่ง LMP_detach ดังรูปที่ 3-22 เหตุผลที่นำเอาการถอดออกมาเป็นคำสั่งด้วย คือ

- เพื่อให้ผู้ใช้ที่อยู่ปลายทางการติดต่อสามารถยกเลิกการเชื่อมต่อเองได้
- ประหยัดทรัพยากรของระบบ
- ต้องการที่จะปิดการทำงาน

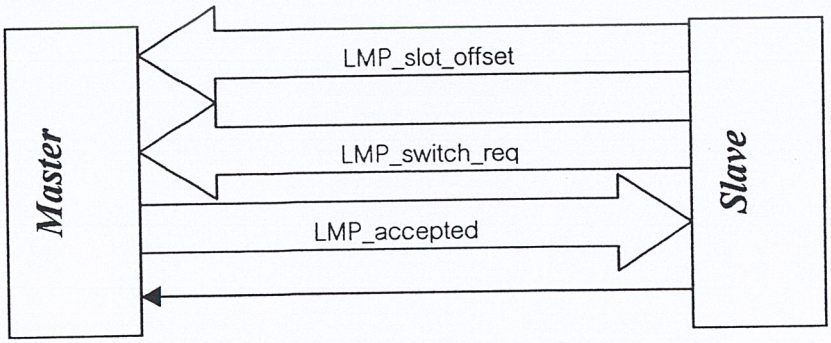


รูปที่ 3-22 ลำดับข้อความ LMP ที่ใช้ในการทำ LMP_detach

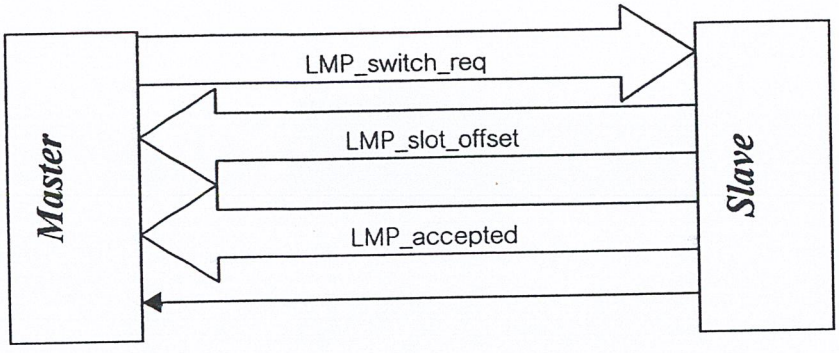
3.3.5 การสับเปลี่ยนหน้าที่

โดยทั่วไปแล้วอุปกรณ์ที่ทำการเพจจะเป็นมาสเตอร์ส่วนอุปกรณ์ที่ทำการเพจสแกนจะเป็นสเลฟ ในการส่งข้อมูลสเลฟสามารถส่งข้อมูลได้เพียงการตอบกลับเท่านั้น มาสเตอร์จะเป็นตัวควบคุมการทำงาน กล่าวได้ว่ามาสเตอร์เป็นตัวควบคุมการใช้แบนด์วิดธ์ของสเลฟ

ในบางกรณีการสับเปลี่ยนบทบาทระหว่างมาสเตอร์และสเลฟอาจไม่เหมาะสม เนื่องจากต้องทำการปิดการเชื่อมต่อก่อน จากนั้นจึงเริ่มต้นติดต่อใหม่อีกครั้งนึงโดยอุปกรณ์ที่เป็นสเลฟจะทำกระบวนการเพจจิง และอุปกรณ์ที่เป็นมาสเตอร์จะทำกระบวนการเพจสแกน ซึ่งจะทำให้เสียเวลาในการทำงานไปมาก โดยในช่วงเวลาที่การเชื่อมต่อครั้งใหม่กำลังถูกสร้างขึ้นนี้ จะไม่สามารถส่งข้อมูลใดๆได้ เพื่อไม่ให้เกิดการเสียเวลาดังกล่าวขึ้น LMP จึงจัดเตรียมวิธีที่มาสเตอร์และสเลฟจะทำการสับเปลี่ยนหน้าที่กัน ดังแสดงในรูปที่ 3-23 และรูปที่ 3-24



รูปที่ 3-23 ลำดับข้อความ LMP ที่ใช้ในการสับเปลี่ยนหน้าที่เมื่อสเลฟเป็นตัวเริ่มต้น



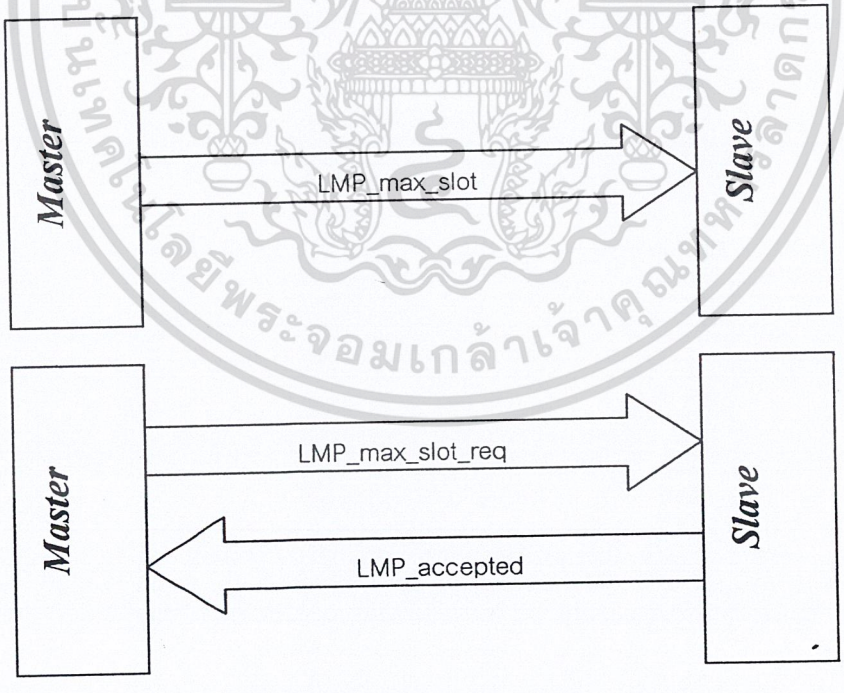
รูปที่ 3-24 ลำดับข้อความ LMP ที่ใช้ในการสับเปลี่ยนหน้าที่เมื่อมาสเตอร์เป็นตัวเริ่มต้น

3.3.6 การควบคุมการใช้มัลติสล็อตแพ็กเกต

เมื่อการเชื่อมต่อถูกสร้างขึ้นมาแล้ว เริ่มต้นจะใช้การรับส่งข้อมูลแบบแพ็กเกตสล็อตเดี่ยว การใช้มัลติสล็อตแพ็กเกตจะช่วยเพิ่มประสิทธิภาพการทำงานในการใช้แบนด์วิดธ์

เมื่อมาสเตอร์ต้องการที่จะใช้งานแพ็กเกตที่มีขนาดสูงสุดจากสเลฟ จะส่งคำสั่ง LMP_max_slot ดังรูปที่ 3-25 สเลฟไม่สามารถปฏิเสธการใช้งานได้ ดังนั้นจึงไม่ต้องมีการตอบกลับ

ในกรณีที่สเลฟต้องการใช้งานแพ็กเกตที่มีขนาดสูงสุดจากมาสเตอร์ ก็จะส่งคำสั่ง LMP_max_sloy_req ไป ถ้ามาสเตอร์อนุญาตให้สเลฟใช้ขนาดแพ็เกตที่ร้องขอมาก็จะตอบกลับด้วย LMP_accepted มิฉะนั้นจะส่ง LMP_not_accepted และสเลฟจะต้องใช้แพ็กเกตขนาดเท่าเดิมต่อไป



รูปที่ 3-25 ลำดับข้อความ LMP ที่ใช้ในการควบคุมการใช้มัลติสล็อตแพ็กเกต

3.3.7 ความปลอดภัย

บลูทูธเบสแบนได้ถึงการสร้างคีย์ที่ใช้ในการเข้ารหัสไว้ด้วย ในการเข้ารหัสและถอดรหัสนั้น การทำงานภายในจะถูกปรับแต่งค่าและถูกควบคุมโดยเลเยอร์ที่สูงขึ้นไป LMP จัดเตรียมกระบวนการในการเจรจาโหมดการทำงานของการเข้ารหัสและการเข้ารหัสคีย์ที่ใช้ในการรับส่งข้อมูล

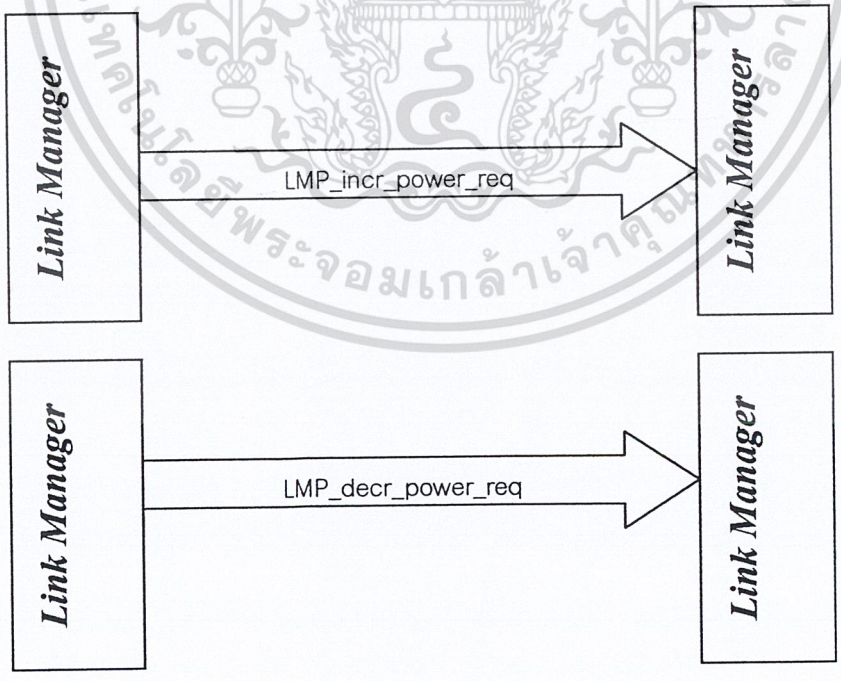
3.3.8 โหมดประหยัดพลังงาน

LMP รองรับการควบคุมเรื่องของโหมดการเชื่อมต่อแบบประหยัดพลังงาน ในโหมดนี้ สเลฟและมาสเตอร์จะยังคงเข้าจังหวะการทำงานกันอยู่ เพียงแต่จะประหยัดพลังงานเอาไว้โดยการมช้พลังงานที่น้อยลง

3.3.9 การควบคุมพลังงาน

พลังงานที่ใช้ในการส่งคลื่นวิทยุออกไปในการติดต่อสื่อสารนั้นจะถูกออกไปในจำนวนที่น้อยที่สุดเพื่อทำให้แบตเตอรีสามารถใช้งานได้นานที่สุด

เนื่องจากอุปกรณ์ที่ใช้ในบลูทูธมักเป็นอุปกรณ์ที่สามารถพกพาได้ ดังนั้นอุปกรณ์เหล่านี้จะมีขนาดเล็ก การประหยัดพลังงานจึงเป็นเรื่องที่สำคัญสำหรับอุปกรณ์ประเภทนี้ แต่อุปกรณ์บลูทูธจะรู้ได้อย่างไรว่าระดับพลังงานเพียงใดจึงจะต่ำเพียงพอที่ให้การส่งข้อมูลยังเป็นไปได้อย่างดี มีเพียงฝั่งที่รับข้อมูลเท่านั้น จึงจะทราบได้ว่าพลังงานที่ได้รับมาดีเพียงพอรึยัง ดังนั้นบลูทูธจึงกำหนดให้อุปกรณ์ที่เป็นตัวรับข้อมูลสามารถร้องขอการเปลี่ยนแปลงระดับพลังงานที่ใช้ในการส่งข้อมูลได้ ดังรูปที่ 3-26



รูปที่ 3-26 ลำดับข้อความ LMP ที่ใช้ในการเปลี่ยนระดับพลังงาน

3.3.10 คุณภาพของการบริการ

LMP รองรับการปรับแต่งคุณภาพของการทำงานในการเชื่อมต่อ โดยจะเปลี่ยนชนิดของแพ็กเกตที่ใช้ในการรับส่งข้อมูล ดังนั้นข้อมูลจะถูกส่งที่ความเร็วสูงเมื่อแชนแนลมีคุณภาพดี และจะส่งข้อมูลที่ความเร็วต่ำเพื่อป้องกันการเกิดข้อผิดพลาดไรการรับส่งข้อมูล

3.3.11 ข้อมูลข่าวสาร

LMP จัดเตรียมข้อความที่หลากหลายในการร้องขอข้อมูลจากอุปกรณ์ต่างๆที่ทำการเชื่อมต่ออยู่ ข้อมูลบางชนิดถูกร้องขอมาจากอุปกรณ์ที่เป็นโฮสต์ ส่วนข้อมูลบางชนิดถูกร้องขอมาจากส่วนจัดการการเชื่อมต่อเอง

3.3.12 คุณสมบัติที่รองรับ

คุณสมบัติบางตัวในมาตรฐานบลูทูธเป็นความสามารถเสริม LMP_features_req ใช้ในการสอบถามว่าคุณสมบัติใดบ้างที่รองรับ และ LMP_features_res จะตอบกลับด้วยข้อมูลต่างๆดังนี้

- มัลติสล็อตแพ็กเกต
- การเข้ารหัส
- โหมดประหยัดพลังงาน
- การควบคุมการใช้พลังงาน
- คุณภาพของแชนแนล
- การเชื่อมต่อ SCO
- การเข้ารหัสข้อมูลที่เป็นเสียง
- ความสามารถเสริมในการเพจจิง
- การสับเปลี่ยนหน้าที่ระหว่างมาสเตอร์และสเลฟ

3.3.13 เวอร์ชัน LMP

ใช้ในการสอบถามถึงข้อมูลของเวอร์ชันที่ใช้จากอุปกรณ์บลูทูธ ซึ่งข้อมูลเกี่ยวกับเวอร์ชันมีดังต่อไปนี้

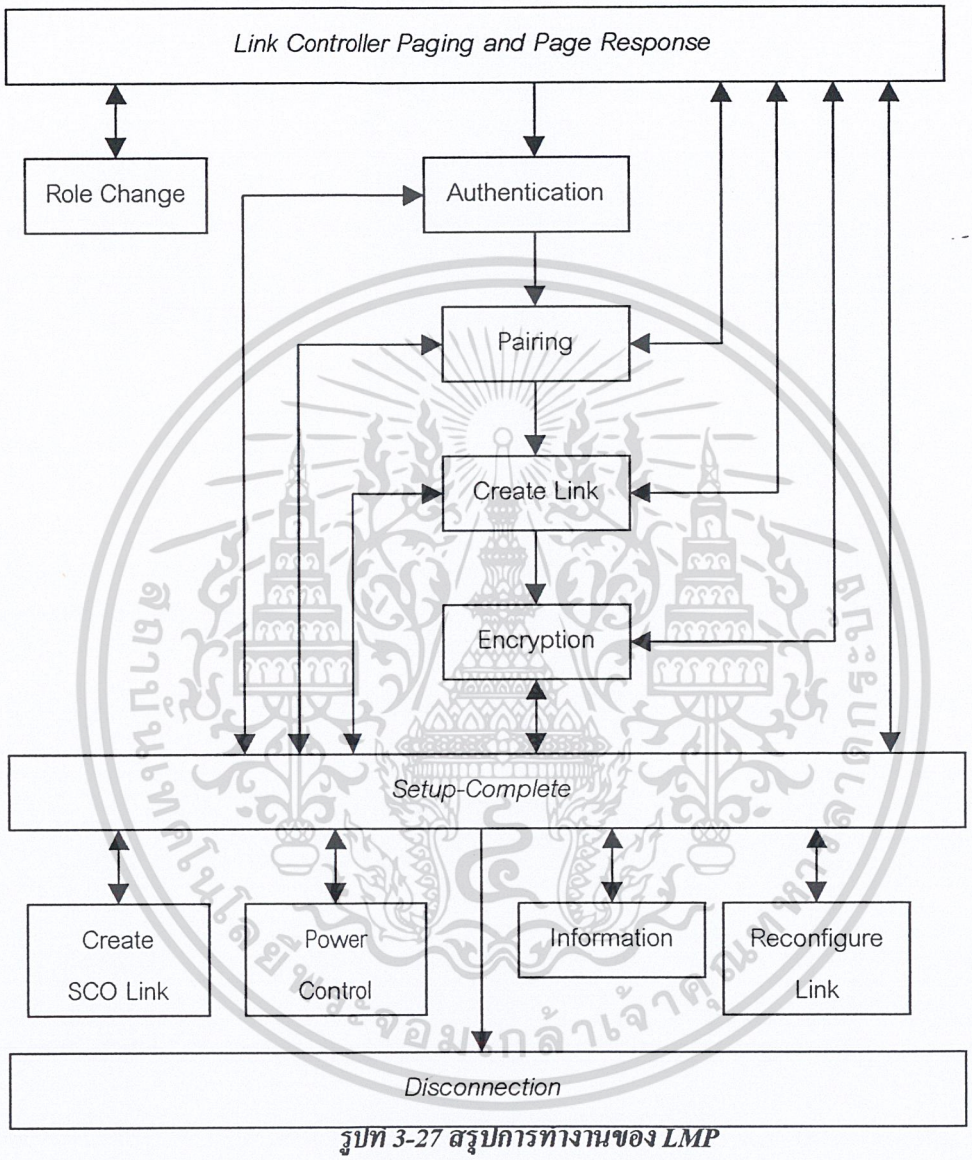
- VersNr – เป็นเวอร์ชันที่ LMP รองรับ
- CompId – เป็นเลข ID ของบริษัทที่สร้าง LMP ขึ้นมา
- SubVersNr – เป็นเลขเวอร์ชันของบริษัทที่เป็นผู้สร้าง LMP

3.3.14 การร้องขอชื่อ

อุปกรณ์บลูทูธแต่ละตัวจะมีชื่อเล่นซึ่งมีขนาดไม่เกิน 248 ไบต์ LMP รองรับการร้องขอชื่อเล่นของอุปกรณ์ที่ทำการติดต่ออยู่ด้วย

3.3.15 สรุปการทำงานของส่วนจัดการการเชื่อมต่อ

รูปที่ 3-27 เป็นการสรุปการทำงานของ LMP หลังจากที่ LC ได้สิ้นสุดกระบวนการเพจจิงและตอบสนองการเพจเรียบร้อยแล้ว ก็จะเริ่มทำการแลกเปลี่ยนข้อมูลโดยการส่งข้อความต่างๆเพื่อที่จะยืนยันการเริ่มต้นการเชื่อมต่อว่าเสร็จสมบูรณ์แล้ว



รูปที่ 3-27 สรุปการทำงานของ LMP

ภายในขั้นตอนนี้จะมีกระบวนการที่เกี่ยวกับเรื่องของความปลอดภัย เช่น การพิสูจน์สิทธิ์ การเพจจิง, การเข้ารหัส ซึ่งจะต้องทำให้เสร็จเรียบร้อยก่อนที่การเชื่อมต่อจะเสร็จสมบูรณ์

เมื่อการเชื่อมต่อเสร็จสมบูรณ์แล้ว จะมีการเชื่อมต่อแบบ SCO ให้ใช้งานได้มากสุดถึงสามการเชื่อมต่อ ซึ่งโหมดในการทำงานนี้สามารถปรับเปลี่ยนได้ ไม่ว่าจะเป็น คุณภาพของการบริการ ชนิดของแพ็กเกต หรือระดับของพลังงานที่ใช้ในการส่งข้อมูล เป็นต้น เมื่อไม่ต้องการการเชื่อมต่ออีกต่อไปแล้ว LMP ก็จะทำการปิดการเชื่อมต่อ

3.4 ส่วนควบคุมการเชื่อมต่อโฮสต์ (HCI : the Host Controller Interface)

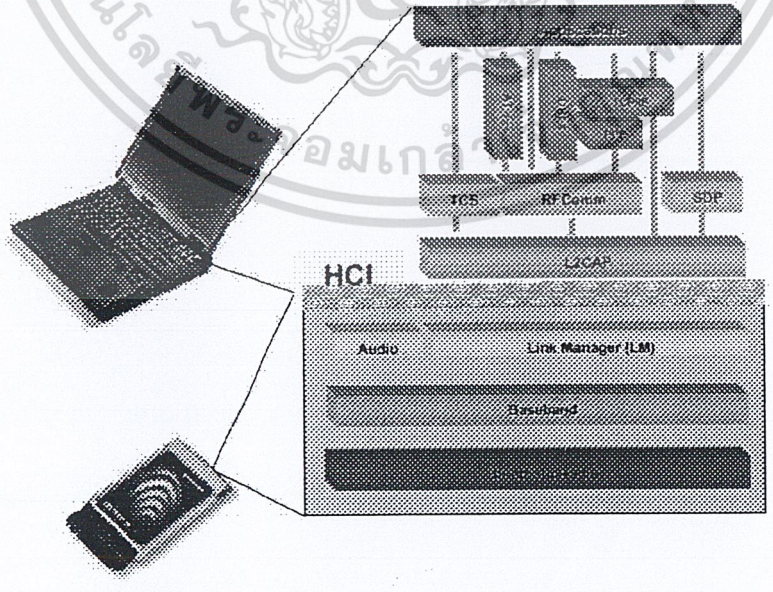
ในบลูทูธบางระบบนั้นจะมีเบสแบนและส่วนจัดการการเชื่อมต่อทำงานอยู่บนโพรเซสเซอร์ตัวหนึ่งกับเลเยอร์ในลำดับที่สูงขึ้นไป เช่น L2CAP, SDP, RFCOMM และโปรแกรมประยุกต์ต่างๆที่ทำงานอยู่บนโพรโซสเซอร์อีกตัวหนึ่งที่อยู่บนฝั่งของโฮสต์ การ์ดบลูทูธ PCMCIA เป็นตัวอย่างการทำงานในลักษณะนี้ โดยที่ส่วนจัดการการเชื่อมต่อและเบสแบนจะอยู่ที่การ์ด PCMCIA และเลเยอร์ที่สูงขึ้นไปจะอยู่บนโพรเซสเซอร์ของคอมพิวเตอร์

ในระบบที่เลเยอร์ระดับบนนั้นทำงานอยู่บนโฮสต์โพรเซสเซอร์และเลเยอร์ระดับล่างทำงานอยู่บนอุปกรณ์บลูทูธนั้น ส่วนเชื่อมต่อเป็นสิ่งจำเป็นเพื่อเชื่อมการทำงานระหว่างเลเยอร์ในสองชั้นนี้ มาตรฐานบลูทูธได้กำหนดถึงส่วนควบคุมการเชื่อมต่อโฮสต์ขึ้นมาเพื่อให้เป็นมาตรฐานการเชื่อมต่อ ดังเช่นตัวอย่างที่กล่าวมา จึงเป็นไปได้ที่เลเยอร์ระดับบนที่เป็นซอฟต์แวร์อยู่บนเครื่องคอมพิวเตอร์นั้นสามารถที่จะทำงานร่วมกับการ์ด PCMCIA ของผู้ผลิตใดก็ได้

อุปกรณ์เช่นบลูทูธการ์ดนั้นใช้ HCI เพื่อแยกเลเยอร์ระดับบนและระดับล่างออกจากกัน เพื่อเหตุผลดังต่อไปนี้

- โฮสต์ เช่น คอมพิวเตอร์มีความสามารถในการรองรับการทำงานของเลเยอร์ระดับบนได้ ทำให้อุปกรณ์บลูทูธสามารถใช้หน่วยความจำและโพรเซสเซอร์ที่มีความสามารถน้อยได้ ซึ่งเป็นการลดค่าใช้จ่ายในกระบวนการผลิต
- อุปกรณ์ที่เป็นโฮสต์นั้นสามารถที่จะหยุดพักและถูกปลุกขึ้นมาทำงานอีกครั้งเมื่อมีการเชื่อมต่อเข้ามา
- ส่วนเชื่อมต่อ HCI นั้นมีประโยชน์ในการทดสอบการทำงานกับอุปกรณ์บลูทูธ

รูปที่ 3-28 แสดงการทำงานของบลูทูธโพรโตคอลแสตก ซึ่งจะถูกแบ่งออกเป็นสองส่วนโดยมีส่วนเชื่อมต่อ HCI เป็นตัวเชื่อมต่อการทำงาน ดังรูป



รูปที่ 3-28 ส่วนเชื่อมต่อ HCI บนบลูทูธโพรโตคอลแสตก

สาเหตุที่เราไม่ทำการรวมแสตกทั้งหมดลงบนโพรเซสเซอร์ของโฮสต์เนื่องจาก การทำงานแบบสล็อต ซึ่งโฮสต์จะต้องมีความสามารถในการตอบสนองภายในเวลาที่รวดเร็วมากจากการเรียกของคลื่นวิทยุ ถึงแม้ว่าโฮสต์จะสามารถทำงานได้ในระดับของล้านคำสั่งต่อวินาที แต่ส่วนใหญ่แล้วไม่สามารถรับประกันได้ว่าโพรเซสเซอร์จะว่างเมื่อจำเป็นต้องเรียกใช้งาน ดังนั้น จึงเป็นการดีที่จะนำเลเยอร์ที่มีความสำคัญแยกออกมาจากโพรเซสเซอร์ของโฮสต์ เพื่อรับประกันถึงความรวดเร็วในการทำงาน

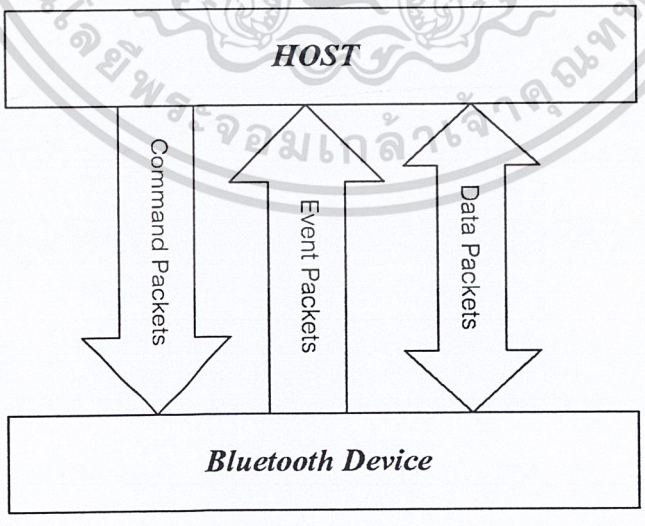
มาตรฐานของ HCI นั้นอนุญาตให้ตัวขับสามารถที่จะทำงานกับบลูทูธโมดูลที่ผลิตมาจากแหล่งต่างๆ นอกจากนี้ยังจัดเตรียมการทดสอบในส่วนการเชื่อมต่อเพื่อเป็นการทดสอบการใช้คลื่นวิทยุและการทำงานของโพรโตคอลแสตก

เป็นไปได้ที่จะทำให้ระบบบลูทูธมีโพรโตคอลแสตกที่ทำงานอยู่บนโพรเซสเซอร์เดียว ตัวอย่างเช่น หูฟังไร้สาย ซึ่งจำเป็นต้องมีขนาดเล็ก, ราคาไม่แพง, น้ำหนักเบา และกินพลังงานต่ำ การทำงานของโพรโตคอลแสตกบนโพรเซสเซอร์เดียวดูจะเป็นวิธีที่เหมาะสมกว่า อย่างไรก็ตาม อุปกรณ์ที่ทำงานบนโพรเซสเซอร์เดียวนี้สามารถที่จะรองรับส่วนเชื่อมต่อ HCI ได้ด้วย เพื่อใช้ในการทดสอบ

3.4.1 ชนิดของแพ็กเกตใน HCI

มาตรฐานบลูทูธได้กำหนดส่วนเชื่อมต่อ HCI ไว้ดังนี้

- แพ็กเกตคำสั่งที่ถูกใช้โดยโฮสต์ในการควบคุมการทำงานของโมดูล
- แพ็กเกตบอกเหตุการณ์ถูกใช้โดยโมดูลเพื่อแจ้งข่าวสารไปยังโฮสต์เกี่ยวกับความเปลี่ยนแปลงที่เกิดขึ้นในเลเยอร์ระดับล่าง
- แพ็กเกตข้อมูลที่ส่งผ่านข้อมูลธรรมดาและข้อมูลที่เป็นเสียงระหว่างโฮสต์กับโมดูล
- เลเยอร์ในการรับส่งข้อมูล HCI ซึ่งจะเป็นตัวส่งแพ็กเกต HCI



รูปที่ 3-29 ชนิดของแพ็กเกตใน HCI

3.4.2 เลเยอร์ในการรับส่งข้อมูล HCI

เลเยอร์ในการรับส่งข้อมูลนี้มีหน้าที่ในการนำแพ็กเกต HCI จากโฮสต์ไปยังโมดูลบลูทูธ ซึ่งบลูทูธได้กำหนดเลเยอร์ในการรับส่งข้อมูลไว้ดังนี้

- USB (Universal Serial Bus)
- RS-232 การเชื่อมต่อแบบอนุกรม ซึ่งมีความสามารถในการการแก้ไขข้อมูลที่ผิดพลาดด้วย
- UART (Universal Asynchronous Receiver Transmitter) เป็นการเชื่อมต่อแบบอนุกรมที่ไม่มีความสามารถในการการแก้ไขข้อมูลที่ผิดพลาด

3.4.2.1 โครงร่าง USB

USB นำแพ็กเกต HCI ชนิดต่างๆมาเทียบกับมาตรฐานของ USB ที่เป็นปลายทางต่างๆดังนี้

- คำสั่ง HCI – ส่วนควบคุมปลายทางของ USB
- ข้อมูลแบบ ACL ในชั้น HCI – ข้อมูลปลายทางขนาดใหญ่ของ USB
- ข้อมูลแบบ SCO ในชั้น HCI – ข้อมูลแบบ SCO ที่ปลายทางของ USB
- เหตุการณ์ใน HCI – การอินเทอร์รัพที่ปลายทางของ USB

ในการที่จะทำให้ตัวขับของ USB สามารถที่จะทำงานได้นั้น โค้ดที่เป็นคลาสของ USB จะต้องถูกกำหนดให้กับอุปกรณ์บลูทูธซึ่งจะระบุคลาส, คลาสย่อยและโค้ดของโพรโตคอลดังนี้

- bbDeviceClass = 0xE0 (ส่วนควบคุมไร้สาย)
- bbDeviceSubClass = 0x01 (ส่วนควบคุมคลื่นวิทยุ)
- bbDeviceProtocol = 0x01 (โปรแกรมมิ่งบลูทูธ)

3.4.2.2 เลเยอร์การรับส่งข้อมูลแบบอนุกรม

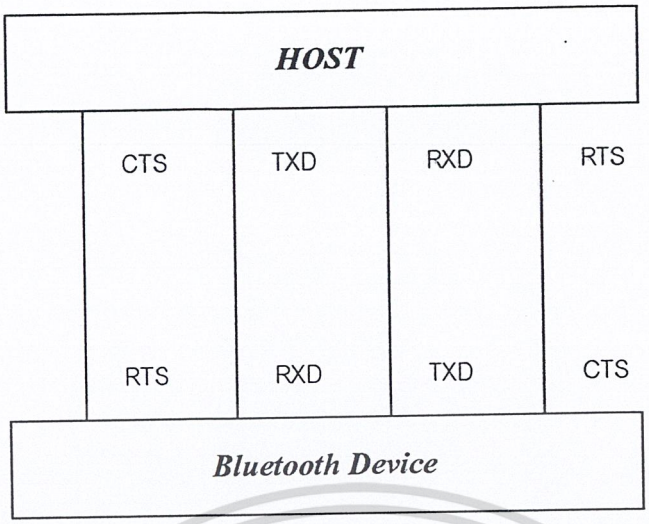
RS-232 และ UART ทั้งคู่เป็นการเชื่อมต่อแบบอนุกรม ซึ่งมีโครงสร้างดังต่อไปนี้

- 0x01 – แพ็กเกตคำสั่ง
- 0x02 – แพ็กเกตข้อมูลแบบ ACL
- 0x03 – แพ็กเกตข้อมูลแบบ SCO
- 0x04 – แพ็กเกตบอกเหตุการณ์

ขณะที่ UART ไม่จำเป็นต้องจัดการเรื่องการแก้ไขข้อผิดพลาดที่เกิดขึ้นในข้อมูล แต่ RS-232 จำเป็นต้องมีความสามารถในส่วนนี้ ดังนั้น RS-232 จึงจัดเตรียมส่วนต่อไปนี้เพิ่มขึ้นมา

- 0x05 – แพ็กเกตแสดงข้อความที่ผิดพลาด
- 0x06 – แพ็กเก็ตที่ใช้ในการเจรจาติดต่อ

เลเยอร์การรับส่งข้อมูลแบบ UART จะเป็นการเชื่อมต่อแบบลักษณะของนัลโมเด็มดังรูปที่ 3-30 ข้อมูลที่ใช้ส่ง (TXD : Transmit data) จะถูกเชื่อมต่อไปยังส่วนที่รับข้อมูล (RXD : Receive data) และสัญญาณเตรียมพร้อมที่จะส่ง (CTS : Clear to send) จะถูกเชื่อมต่อกับสัญญาณพร้อมที่จะส่ง (RTS : Ready to send)



รูปที่ 3-30 การเชื่อมต่อสำหรับเลเยอร์การรับส่งข้อมูลแบบ UART

3.4.3 ส่วนควบคุมการไหลของข้อมูล

ในส่วนเชื่อมต่อ HCI นี้อนุญาตให้มีการรับส่งข้อมูลที่ความเร็วมากกว่าความเร็วในการรับส่งของคลื่นวิทยุ ดังนั้นโฮสต์สามารถที่จะส่งข้อมูลลงไปยังเลเยอร์ที่อยู่ระดับล่างได้เร็วกว่าส่งข้อมูลออกไปนอกโมดูลปัญหาที่เกิดขึ้นก็คือ บลูทูธโมดูลมีหน่วยความจำอยู่น้อย ทำให้ถึงแม้ว่าโฮสต์จะส่งข้อมูลได้เร็วแต่โมดูลก็ไม่สามารถที่จะรับข้อมูลนั้นได้อย่างรวดเร็วด้วย ทางออกของปัญหานี้คือการใช้ส่วนควบคุมการไหลของข้อมูล

3.4.3.1 คำสั่งในการควบคุมการไหลของข้อมูล

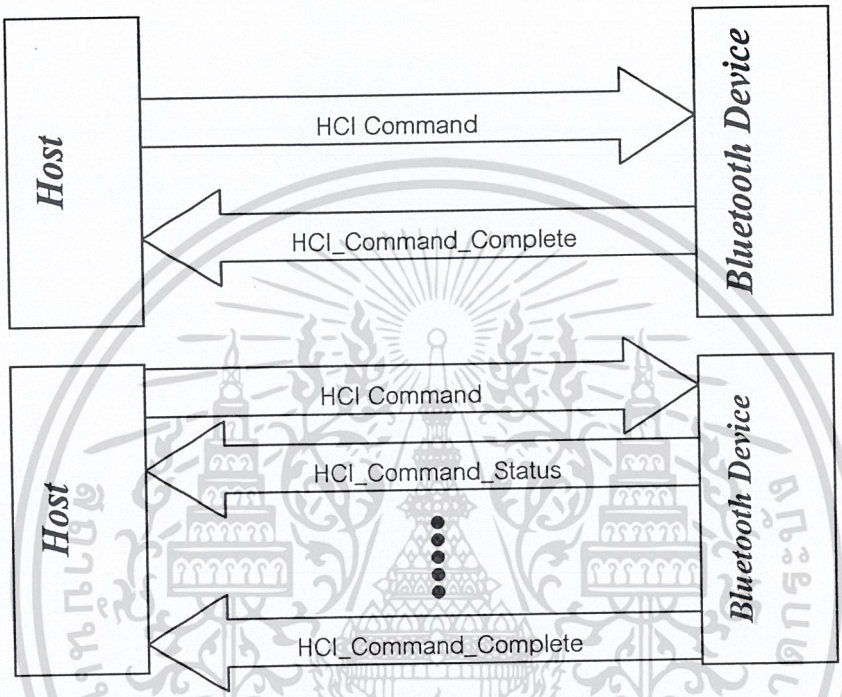
แต่ละคำสั่งที่ส่งออกไปยังบลูทูธโมดูลนั้น โมดูลจะตอบกลับด้วย HCI_Command_Complete หรือ HCI_Command_Status HCI_Command_Complete จะถูกใช้เมื่อคำสั่งสามารถทำงานได้ทันที มิฉะนั้น HCI_Command_Status และ HCI_Command_Complete จะถูกส่งเมื่อคำสั่งได้ถูกทำเสร็จสมบูรณ์แล้ว

3.4.3.2 ส่วนควบคุมการไหลของข้อมูลจากโฮสต์

คำสั่ง HCI_Read_Buffer_Size ใช้เพื่อถามถึงจำนวนบัฟเฟอร์ที่เหลืออยู่ในบลูทูธโมดูล โดยบลูทูธโมดูลจะตอบกลับด้วยจำนวนของแพ็กเกต SCO และ ACL ที่สามารถรองรับได้ และขนาดแพ็กเกต HCI สูงสุดของ SCO และ ACL เมื่อโฮสต์ทราบจำนวนแพ็กเกต SCO และ ACL ที่โมดูลสามารถรองรับได้ ก็จะส่งจำนวนของแพ็กเกตที่ระบุมาจาก HCI ไปยังบลูทูธโมดูล

3.4.3.3 ส่วนควบคุมการไหลของข้อมูลจากโมดูล

โดยทั่วไปแล้วมักจะไม่มีปัญหาสำหรับโฮสต์ที่จะรับข้อมูลจากโมดูลที่ส่งมา อย่างไรก็ตามโฮสต์บางตัวจะมีความสามารถจำกัด เช่น โทรศัพท์มือถือที่ไม่สามารถรับข้อมูลได้ที่อัตราสูงสุด ดังนั้นการควบคุมการไหลของข้อมูลจึงเป็นสิ่งจำเป็น คำสั่งที่ใช้ในการควบคุมการไหลของข้อมูลจากโมดูลบลูทูธ คือ HCI_Set_Host_Controller_To_Host_Flow_Control และคำสั่ง HCI_Host_Buffer_Size เป็นคำสั่งที่โฮสต์บอกโมดูลว่าสามารถรองรับแพ็กเกต SCO และ ACL ได้มากแค่ไหน และขนาดแพ็กเกต HCI สูงสุดของ SCO และ ACL ที่รองรับได้เป็นจำนวนเท่าไร



รูปที่ 3-31 ลำดับข้อความที่ใช้ในคำสั่งในการควบคุมการไหลของข้อมูล

3.4.4 การปรับแต่งค่าในโมดูล

HCI จัดเตรียมคำสั่งที่ใช้ในการปรับแต่งค่าไว้เพื่อจะได้ทราบลักษณะต่างๆที่มีอยู่ในโมดูล เนื่องจากคำสั่งต่างๆนี้ไม่เกี่ยงข้องกับการเชื่อมต่อเลย ดังนั้นจะส่งคำตอบกลับมาทันทีด้วยคำสั่ง HCI_Command_Complete

3.4.4.1 เวอร์ชัน

ก่อนที่โมดูลจะถูกใช้งานนั้นจำเป็นที่จะต้องทราบเวอร์ชันที่มาตรฐานรับรองเสียก่อน ซึ่งคำสั่ง HCI_Read_Local_Version จะส่งข้อมูลต่างๆเหล่านี้กลับมา

- ไบต์บอกสถานะ
- HCI ที่ปรับปรุงใหม่แล้ว
- เวอร์ชัน LMP
- ชื่อผู้ผลิต

- LMP เวอร์ชันย่อย

3.4.4.2 การร้องขอชื่ออุปกรณ์ระยะใกล้และไกล

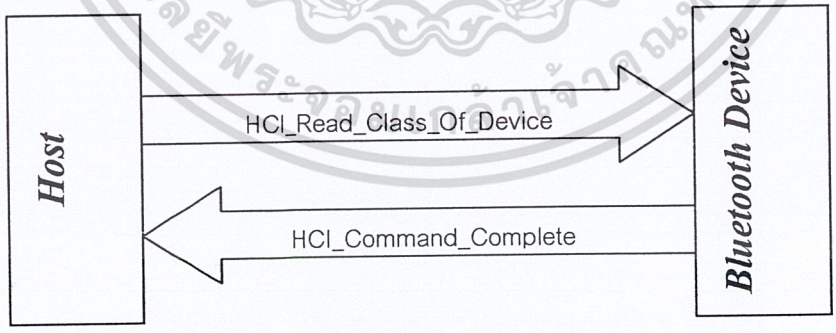
อุปกรณ์บลูทูธทุกตัวจะมีชื่อเล่นอยู่ซึ่งเป็นการเข้ารหัสด้วยวิธี UTF-8 มีความยาวได้สูงสุดไม่เกิน 248 ไบต์ คำสั่ง HCI_Read_Local_Name ใช้ในการค้นหาชื่อเล่นของโมดูลระยะใกล้ ส่วนการเปลี่ยนชื่อนั้นใช้คำสั่ง HCI_Change_Local_Name ชื่อเล่นนี้จะถูกส่งเริ่มจากไบต์แรกของข้อมูล

คำสั่ง HCI_Remote_Name_Request จะถูกใช้เมื่อต้องการทราบชื่อเล่นของอุปกรณ์ระยะไกล คำสั่งนี้จะใช้เมื่อไม่มีการเชื่อมต่อกับอุปกรณ์ใดๆอยู่ โดยการทำงาน เริ่มจากการที่อุปกรณ์จะสร้างการเชื่อมต่อขึ้นมา แล้วจึงใช้ LMP_name_req เพื่อสอบถามชื่อไปยังอุปกรณ์ที่อยู่ระยะไกล ดังนั้นอุปกรณ์จะถูกเชื่อมต่อและค่าพารามิเตอร์ต่างๆจะถูกส่งกลับมาใน HCI_Remote_Name_Request ซึ่งจะประกอบด้วย BD_ADDR, Page_Scan_Repetition_Mode, Page_Scan_Mode และ Clock_Offset

ในการสอบถามชื่ออุปกรณ์ที่อยู่ระยะไกลนั้น อาจจะต้องใช้เวลามากขึ้น เพื่อสร้างการเชื่อมต่อไปยังอุปกรณ์นั้นแล้วจึงรับชื่อนั้นกลับมา เมื่อโมดูลได้รับ HCI_Remote_Name_Request กลับมาแล้ว ก็จะตอบกลับด้วย HCI_Command_Status กลับไป เมื่ออุปกรณ์ที่อยู่ระยะไกลได้รับแล้ว โมดูลก็จะส่ง HCI_Remote_Name_Request_Complete กลับไปซึ่งจะมีฟิลด์ที่บอกว่าข้อมูลที่ได้รับมานั้นไม่ว่าจะเป็นแอดเดรสของอุปกรณ์ หรือชื่อเล่นของอุปกรณ์ที่ได้รับมานั้นสำเร็จหรือไม่

3.4.4.3 คลาสของอุปกรณ์

โดยการส่งคำสั่ง HCI_Read_Class_Of_Device ไปยังโมดูล เพื่อที่โฮสต์จะได้รู้คลาสของอุปกรณ์ซึ่งจะใช้ในการส่งแพ็กเกต FHS เมื่อมีการตอบสนองต่อการทำอินไควรี่ โมดูลบลูทูธจะตอบกลับด้วย HCI_Command_Complete ทันที ซึ่งพารามิเตอร์ที่อยู่ในคำสั่งนี้จะมีข้อมูลคลาสของอุปกรณ์ที่โฮสต์ร้องขอมา



รูปที่ 3-32 ลำดับข้อความที่ใช้ในคำสั่ง HCI_Read_Class_Of_Device

3.4.4.4 คุณสมบัติที่รองรับ

บลูทูธมีคุณสมบัติหลายอย่างที่เป็นความสามารถเสริม ดังนั้นจึงจำเป็นที่โฮสต์จะต้องทราบว่าความสามารถใดบ้างที่โมดูลมีอยู่ เพื่อจะใช้ติดต่อกับอุปกรณ์ที่อยู่อีกฝั่งหนึ่งของการเชื่อมต่อ บลูทูธได้จัดเตรียมคำสั่งที่ใช้ในการอ่านความสามารถเสริมของทั้งโมดูลและอุปกรณ์ที่อยู่ระยะไกลด้วย

คำสั่ง HCI_Read_Local_Supported_Features ใช้กับโมดูล และคำสั่งที่ใช้ติดต่อกับอุปกรณ์ระยะไกล คือ HCI_Read_Remote_Supported_Features ซึ่งความสามารถต่างๆที่เป็นความสามารถเสริมมีดังนี้

- แพ็กเกตข้อมูลแบบมัลติสล็อต
- การสับเปลี่ยนหน้าที่ระหว่างมาสเตอร์และสเลฟ
- ความแม่นยำของเวลาที่ใช้
- โหมดการประหยัดพลังงาน
- แชนแนล SCO
- แพ็กเกต HV2 และ HV3 บน SCO แชนแนล
- รูปแบบการเข้ารหัสข้อมูลเสียง
- ความสามารถเสริมในการทำการเพจจิง
- โหมดควบคุมการใช้พลังงาน

3.4.4.5 รหัสประเทศ

ในบางประเทศ เช่น ประเทศฝรั่งเศสนั้น ความถี่ในช่วง ISM ที่ถูกใช้โดยบลูทูธนั้นถูกสงวนไว้ใช้ในทางทหาร ดังนั้นช่วงความถี่ที่ใช้ในประเทศนี้จะแตกต่างจากประเทศอื่นๆ หรืออาจกล่าวได้ว่าฝรั่งเศสมีช่วงความถี่ที่ใช้ได้จำกัดกว่าการใช้ในประเทศอื่นๆ ดังนั้นรหัสประเทศจะเป็นตัวที่ใช้ในการบอกกับโมดูลว่าสามารถใช้ความถี่ได้เต็มช่วงหรือไม่

3.4.4.6 HCI_Read_BD_ADDR

แอดเดรสของอุปกรณ์บลูทูธทุกตัวนั้นไม่สามารถเปลี่ยนได้ เนื่องจากอุปกรณ์แต่ละตัวถูกผลิตมาจากที่ต่างๆกัน ดังนั้นถ้าอนุญาตให้มีการเปลี่ยนแอดเดรสได้นั้น จะเกิดกรณีที่ทีแอดเดรสซ้ำกันเกิดขึ้น ถึงแม้ว่าจะไม่สามารถทำการเปลี่ยนค่าแอดเดรสได้นั้น แต่สามารถที่จะอ่านค่าแอดเดรสขึ้นมาใช้งานได้ ซึ่งข้อมูลนี้เป็นประโยชน์มากในการที่จะเข้าไปใช้งานในโปรแกรมประยุกต์ของอุปกรณ์ที่อยู่ระยะไกล โดยการเพจจิงด้วยแอดเดรสของอุปกรณ์ตัวนั้น คำสั่ง HCI_Read_BD_ADDR จึงใช้เพื่ออนุญาตให้โฮสต์ดึงข้อมูลแอดเดรสของอุปกรณ์บลูทูธไปจากโมดูล

3.4.5 การอินไควรี่

ในหัวข้อของเบสแบนได้กล่าวถึงการอินไควรี่ที่ใช้ในการค้นหาอุปกรณ์บลูทูธตัวอื่นๆที่มีอยู่ในเครือข่าย ในกระบวนการอินไควรี่นั้นจะถูกควบคุมโดย HCI

3.4.5.1 การเริ่มต้นกระบวนการอินไควรี่

คำสั่ง HCI_Inquiry ใช้ในการเริ่มต้นกระบวนการอินไควรี่ ซึ่งประกอบด้วยพารามิเตอร์สามตัว คือ

- อินไควรี่ LAP – จะใช้ส่วนเข้าถึงการอินไควรี่
- ความยาวของการอินไควรี่ –เป็นความยาวทั้งหมดของการอินไควรี่
- จำนวนของการตอบกลับ – เป็นจำนวนของผลลัพธ์การอินไควรี่ที่รอคอย

3.4.5.2 ส่วนควบคุมการตอบกลับการอินไควรี่

การตอบกลับของอินไควรี่ที่ได้รับมานั้นจะถูกรายงานไปยังโฮสต์ใน HCI_Inquiry_Result โดยที่แต่ละ HCI_Inquiry_Result นั้นจะบรรจุข้อมูลที่จำเป็นในการเชื่อมต่อกับอุปกรณ์ที่ตอบกลับมา

- จำนวนของอินไควรี่ที่ตอบกลับจะอยู่รวมในแพ็กเกตบอกเหตุการณ์
- แอดเดรสบลูทูธของอุปกรณ์ที่ตอบกลับ ซึ่งจะใช้ในการเพจ
- โหมดการเพจสแกนซ้ำ มีอยู่สามโหมด คือ R0 ใช้ในกรณีที่ทำการเพจสแกนอย่างต่อเนื่อง, R1 ใช้ในกรณีที่ทำการเพจสแกนทุกๆ 1.28 s และ R2 ใช้ในกรณีที่ทำการสแกนทุกๆ 2.56 s
- โหมดของการเพจแสกเป็นคาบ
- โหมดของการเพจสแกน
- คลาสของอุปกรณ์ที่ทำการตอบกลับ
- ออฟเซ็ตของคล็อก

ในลักษณะงานบางอย่างนั้น การอินไควรี่ไม่จำเป็นต้องแสดงรายชื่ออุปกรณ์ที่มีอยู่ทั้งหมดในละแวกนั้นออกมา เราสามารถที่จะกำหนดได้ว่าให้แสดงเฉพาะรายชื่ออุปกรณ์ที่เราสนใจเท่านั้น โดยมีประเด็นตัดสินใจอยู่ 3 อย่างด้วยกัน คือ

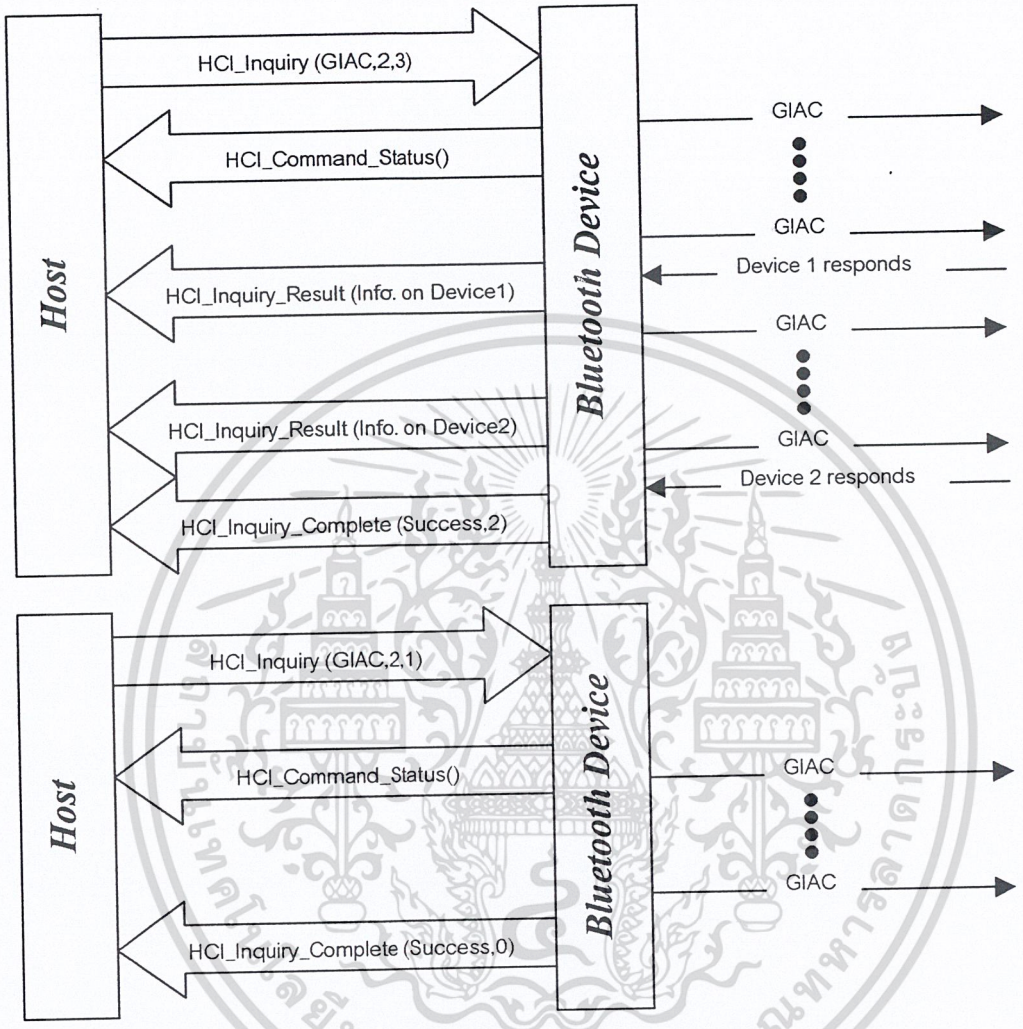
- ให้แสดงเฉพาะรายชื่ออุปกรณ์ใหม่ที่ไม่เคยเจอมาก่อน
- ให้แสดงเฉพาะรายชื่ออุปกรณ์ในประเภทนั้นๆ (โดยระบุชนิดของอุปกรณ์ที่ต้องการก่อน)
- ให้แสดงเฉพาะรายชื่ออุปกรณ์ที่มีแอดเดรสนั้นๆออกมา (โดยระบุแอดเดรสของอุปกรณ์ก่อน)

3.4.5.3 การหยุดกระบวนการอินไควรี่

การอินไควรี่จะหยุดเมื่อจำนวนอุปกรณ์ที่ตอบกลับการอินไควรี่เท่ากับจำนวนใน Num_response หรือ เมื่อช่วงเวลาในการอินไควรี่หมดลง กรณีใดกรณีหนึ่งที่เกิดขึ้นก่อนจะทำให้การอินไควรี่หยุดทำงาน รูปที่ 3-33 แสดงสองวิธีที่การอินไควรี่จะสามารถประสบผลสำเร็จ ในรูปบนนั้นการอินไควรี่เริ่มขึ้นโดยใช้ GIAC อุปกรณ์บลูทูธจะทำการส่งชุดของแพ็กเกต ID ที่ประกอบด้วย GIAC เมื่อมีอุปกรณ์สองตัวตอบกลับมา แต่ละข้อความจะตอบกลับมาด้วย HCI_Inquiry_Result เมื่ออุปกรณ์ตัวที่สองตอบกลับมาแล้วนั้น HCI_Inquiry_Complete จะถูกส่งออกไป ส่วนรูปข้างล่างนั้นไม่มีข้อความตอบกลับมา เมื่อเวลาหมดลงแล้ว HCI_Inquity_Complete จะถูกส่งกลับไปโดยไม่มีข้อความตอบกลับ

ในกรณีที่โฮสต์ต้องการที่จะหยุดการทำอินไควรี่ก่อนหน้านั้น สามารถทำได้โดยใช้คำสั่ง HCI_Inquiry_Cancel เมื่อการอินไควรี่เสร็จเรียบร้อยแล้ว หรือถูกหยุดเนื่องจากคำสั่ง

HCI_Inquiry_Cancel แพ็กเกต HCI_Inquiry_Complete จะถูกส่งออกไป ซึ่งในนั้นจะประกอบด้วยข้อมูลต่างๆ คือ จำนวนอุปกรณ์ทั้งหมดที่ตอบกลับมาในช่วงระยะเวลาการอินไควรี่ และสถานะที่บอกว่าการอินไควรี่เสร็จสมบูรณ์หรือไม่



รูปที่ 3-33 ลำดับข้อความที่ใช้ในการทำอินไควรี่โดยเสร็จสมบูรณ์

3.4.5.4 ช่วงเวลาในการอินไควรี่

โดยทั่วไปแล้วโฮสต์ต้องการที่จะทำการอินไควรี่เพื่อค้นหาอุปกรณ์อื่นๆที่อยู่ในระยะเครือข่าย การอินไควรี่อย่างต่อเนื่องนี้จะทำให้สิ้นเปลืองเปลืองพลังงานมาก ดังนั้นวิธีแก้ปัญหาคือการทำการอินไควรี่เป็นคาบๆในช่วงเวลาสั้นๆ โดยการใช้คำสั่ง HCI_Periodic_Inquiry_Mode

ถ้าการทำอินไควรี่ทำในช่วงเวลาที่ซ้ำๆกัน มีโอกาสเป็นไปได้ที่อุปกรณ์ที่ทำการอินไควรี่และอุปกรณ์ที่ทำการสแกนจะมีโอกาสที่ไม่ได้เจอกัน ดังนั้นการอินไควรี่จะเริ่มทำงานในช่วงเวลาที่สุ่มขึ้นมา ซึ่งค่าที่สุ่มขึ้นมานี้อยู่ระหว่างค่าสูงสุดและค่าต่ำสุดซึ่งถูกกำหนดไว้ใน HCI_Periodic_Inquiry_Mode

3.4.6 การอินไควรี่สแกน

อุปกรณ์บลูทูธอนุญาตให้อุปกรณ์อื่นๆค้นหามันได้โดยการทำอินไควรี่สแกน อุปกรณ์ที่ทำการอินไควรี่สแกนจะเป็นอุปกรณ์ที่ถูกค้นพบได้จะอยู่ในโหมดของการถูกค้นพบ อุปกรณ์ที่อยู่ในโหมดนี้จะตอบกลับไปยังอุปกรณ์ที่ทำการอินไคว เพื่อให้อุปกรณ์ที่ทำการอินไควนั้นรับทราบข้อมูลของอุปกรณ์ที่ถูกค้นพบได้

3.4.6.1 GIAC และ LIAC

เมื่ออุปกรณ์บลูทูธทำการอินไควรี่แล้ว มันจะส่งแพ็กเกต ID ที่ประกอบด้วยข้อมูล IAC โดยทั่วไปแล้ว GIAC จะถูกใช้ แต่ LIAC ก็สามารถนำมาใช้ได้ในช่วงเวลาสั้นๆ อุปกรณ์ที่ทำการอินไควรี่สแกนจะคอยฟัง IAC ที่อุปกรณ์ที่ทำการอินไควรี่ส่งมา

3.4.6.2 ช่วงเวลาในการอินไควรี่สแกน

ดังเช่นปัญหาการทำอินไควรี่ อุปกรณ์ที่ทำการอินไควรี่สแกนต้องการทราบว่ามีอุปกรณ์ใดบ้างที่ทำการอินไควรี่อยู่ เพื่อตัวอุปกรณ์นั้นจะได้ถูกพบ แต่การทำการอินไควรี่สแกนแบบต่อเนื่องนั้น จะทำให้เกิดปัญหาสองอย่าง คือ

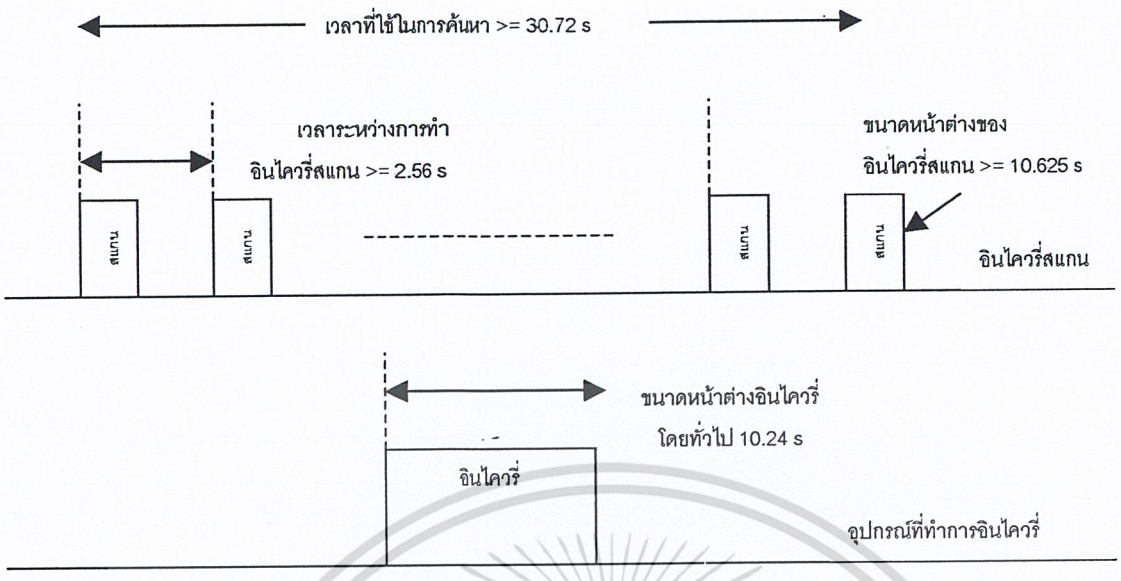
- ในการเปิดคลื่นวิทยุเพื่อใช้ในการรับสัญญาณตลอดเวลาจะทำให้สิ้นเปลืองพลังงาน
- อุปกรณ์บางตัวต้องการที่จะทำการสแกน ในขณะที่กำลังเชื่อมต่อกับอุปกรณ์อื่นๆอยู่ด้วยการทำอินไควรี่สแกนแบบต่อเนื่องนั้นจำทำให้การเชื่อมต่อที่มีอยู่หลุดออกไป

ทางออกของปัญหา คือ การทำอย่างรวดเร็วในช่วงเวลาสั้นๆ ในโพรไฟล์การเข้าถึงแบบทั่วไปแนะนำว่าอุปกรณีควรทำการสแกนในช่วงเวลา 10.625 มิลิวินาที ถ้าอุปกรณ์ทำการสแกนในช่วงระยะเวลาสั้นมากๆ โอกาสที่จะไม่พบการอินไควรี่อาจเกิดขึ้นได้ ดังนั้นจึงแนะนำว่าการอินไควให้ทำนาน 10.24 วินาที และอุปกรณ์ที่อยู่ในโหมดที่ถูกค้นพบได้นั้นจะมีช่วงเวลาในแต่ละครั้ง 2.56 วินาที ตั้งแต่เริ่มทำการอินไควรี่ครั้งแรก และเริ่มทำอินไควรี่ครั้งต่อไป นั่นหมายความว่าขณะที่อุปกรณ์ทำการอินไควรี่อยู่นั้น อุปกรณ์ที่ถูกค้นพบได้จะทำการอินไควรี่สแกนหลายๆครั้ง เพื่อให้ถูกค้นพบได้โดยง่าย

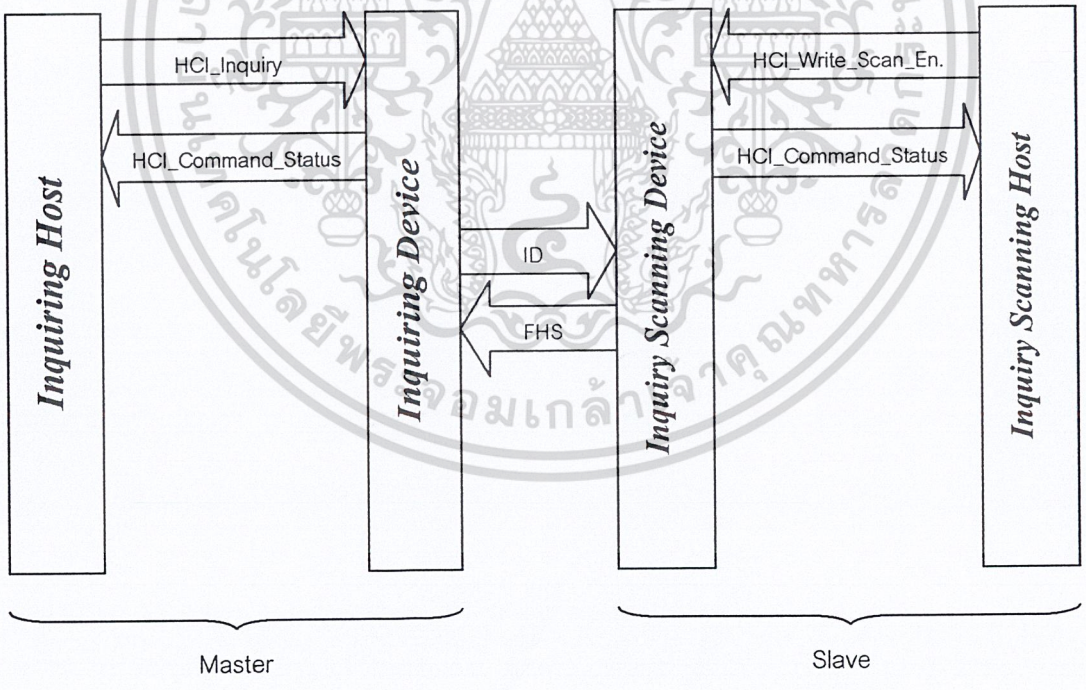
3.4.6.3 การเริ่มทำการอินไควรี่สแกน

อุปกรณ์สามารถทำการอินไควรี่สแกนได้โดยใช้คำสั่ง HCI_Write_Scan_Enable และคำสั่ง HCI_Read_Scan_Enable จะถูกใช้เมื่อต้องการตรวจสอบว่าการอินไควรี่สแกนถูกเปิดใช้อยู่หรือไม่

รูปที่ 3-34 แสดงลำดับข้อความที่ใช้ในการทำอินไควรี่และอินไควรี่สแกน เมื่ออุปกรณ์เริ่มต้นทำการอินไควรี่สแกน โมดูลจะตอบกลับด้วย HCI_Command_Status เพื่อยืนยันว่าได้รับคำสั่งเรียบร้อยแล้ว หลังจากนั้นโมดูลจะทำการอินไควรี่สแกนและตอบกลับเมื่อเมื่อมีการอินไควรี่เข้ามา



รูปที่ 3-34 เวลาที่ใช้ในการทำการอินไควรี่และการอินไควรี่สแกน



รูปที่ 3-35 ลำดับข้อความ HCI ในการอินไควรี่และอินไควรี่สแกน

3.4.7 การเพจจิง

กระบวนการเพจจิงเป็นการสร้างการเชื่อมต่อเพื่อสร้างช่องทางการสื่อสารในการรับส่งข้อมูลระหว่างอุปกรณ์ โดยเริ่มจากอุปกรณ์ที่เป็นมาสเตอร์ส่ง ID ไปยังสเลฟ ID ที่ส่งไปนี้จะเป็น SAC (Slave Access Code) ซึ่งได้มาจากขั้นตอนการทำกระบวนการอินไควรี่ ในกรณีที่มาสเตอร์รู้จักแอดเดรสของสเลฟแล้ว ก็สามารถมาเริ่มทำการเพจจิงได้เลยโดยที่ไม่ต้องทำในส่วนของอินไควรี่ เมื่อสเลฟอยู่ในสถานะเพจสแกนก็จะได้รับ ID นี้ แล้วจึงตอบกลับด้วย ID เดิมที่ส่งมาให้ เมื่อมาสเตอร์ได้รับ ID ที่ตอบกลับมาแล้วก็จะส่งแพ็กเกต FHS ไปให้อีกที เมื่อถึงขั้นตอนนี้อุปกรณ์ทั้ง 2 ฝั่งจะเคลื่อนไปใช้ความถี่ที่ได้ตกลงกันไว้โดยตัวมาสเตอร์ เพื่อเตรียมพร้อมในการสร้างช่องทางการสื่อสาร จากนั้นมาสเตอร์จะถามไปอีกที เพื่อตรวจสอบความพร้อมของสเลฟ เมื่อสเลฟพร้อมแล้วจึงส่งแพ็กเกตตอบกลับมา ซึ่งแพ็กเกตที่ตอบกลับมา ไม่มีการระบุว่าจะต้องเป็นแพ็กเกตแบบไหน แต่โดยส่วนใหญ่แล้วจะส่งแพ็กเกตนูล (NULL) ตอบกลับมา

สำหรับในส่วนของการสแกนนั้น ทั้งการทำอินไควรี่สแกนและเพจสแกน เราสามารถที่จะกำหนดเงื่อนไขได้ว่าจะเปิดใช้การทำงานในลักษณะใดเพื่อให้เหมาะกับงานที่จะทำ ซึ่งเงื่อนไขการสแกนมีดังนี้

เพจสแกน	อินไควรี่สแกน	ผลที่ได้
ใช้	ใช้	อุปกรณ์จะสามารถถูกเชื่อมต่อกับอุปกรณ์อื่นๆ ได้โดยที่ไม่จำเป็นต้องรู้แอดเดรสของอุปกรณ์มาก่อน
ใช้	ไม่ใช้	อุปกรณ์จะสามารถถูกเชื่อมต่อกับอุปกรณ์อื่นๆที่รู้จักแอดเดรสของอุปกรณ์มาก่อน
ไม่ใช้	ใช้	อุปกรณ์จะสามารถถูกเชื่อมต่อกับอุปกรณ์อื่นๆที่ไม่รู้จักแอดเดรสของอุปกรณ์มาก่อน เป็นการติดต่อสื่อสารกันครั้งแรกเท่านั้น
ไม่ใช้	ไม่ใช้	ไม่มีการเชื่อมต่อเกิดขึ้น อยู่ในโหมดประหยัดพลังงาน

ตารางที่ 3-2 การกำหนดเงื่อนไขการสแกน

3.4.8 สรุปการทำงานของส่วนควบคุมการเชื่อมต่อโฮสต์

อุปกรณ์บลูทูธสามารถแบ่งออกเป็น 2 ส่วนหลักๆคือ ส่วนของโฮสต์ที่มีเลเยอร์ระดับบน และโมดูลที่มีเลเยอร์ระดับล่าง ส่วนเชื่อมต่อ HCI นั้นมีเพื่อเชื่อมต่อการทำงานระหว่างแลเยอร์ที่อยู่ระดับบนและเลเยอร์ที่อยู่ระดับล่าง เนื่องจาก HCI เป็นมาตรฐาน ดังนั้นโฮสต์สามารถทำงานร่วมกับอุปกรณ์บลูทูธที่ผลิตมาจากที่ต่างๆกันได้

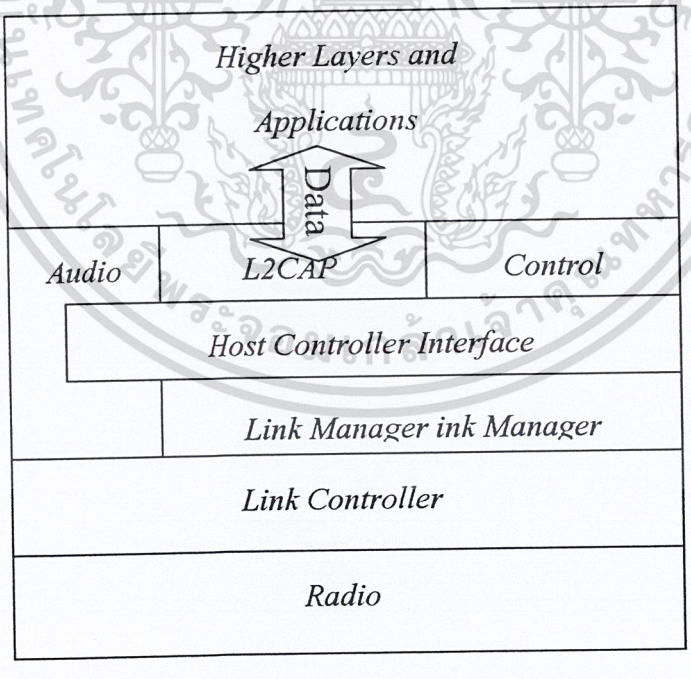
HCI ใช้แพ็กเกตสามชนิดด้วยกัน คือ แพ็กเกตคำสั่งที่ใช้ส่งจากโฮสต์ไปยังโมดูล, แพ็กเกตบอกเหตุการณ์ที่ใช้ส่งจากโมดูลไปยังโฮสต์ และแพ็กเกตข้อมูลซึ่งส่งทั้งสองทิศทาง ซึ่งแพ็กเกตทั้งสามชนิดนี้ใช้ในการควบคุมโมดูลบลูทูธและใช้ส่งข้อมูลที่ต้องการ

บทที่ 4
บลูทูธโฮสต์

4.1 L2CAP (Logical Link Control and Adaptation Protocol)

L2CAP เป็นเลเยอร์ที่นำข้อมูลจากเลเยอร์ระดับบนของบลูทูธโพรโตคอลแสตกและโปรแกรมประยุกต์มาส่งไปยังเลเยอร์ระดับล่างของโพรโตคอลแสตก L2CAP จะส่งแพ็กเกตจาก HCI ไปยัง LM รูปที่ 4-1 แสดงสถานะของ L2CAP ในบลูทูธโพรโตคอลแสตก L2CAP จะส่งข้อมูลที่เป็นข้อมูลธรรมดาเท่านั้น ไม่ส่งข้อมูลที่เป็นเสียง L2CAP มีหน้าที่การทำงานดังนี้

- มัลติเพล็กระหว่างเลเยอร์ที่อยู่ระดับบนของโพรโตคอล เพื่อให้สามารถใช้งานเลเยอร์ที่อยู่ระดับล่างร่วมกันได้
- การแตกข้อมูลออกเป็นแพ็กเกตย่อยๆและประกอบข้อมูลกลับให้เหมือนเดิม ซึ่งจะทำให้สามารถส่งข้อมูลที่มีขนาดใหญ่กว่าขนาดที่เลเยอร์ระดับล่างรองรับได้
- การจัดการในเรื่องของกลุ่ม โดยการจัดเตรียมการส่งข้อมูลแบบทางเดียวไปยังกลุ่มของอุปกรณ์บลูทูธตัวอื่นๆการจัดการเรื่องคุณภาพของการให้บริการสำหรับโพรโตคอลที่อยู่ในระดับบน



รูปที่ 4-1 สถานะของ L2CAP ในบลูทูธโพรโตคอลแสตก

L2CAP ทำงานบนการเชื่อมต่อแบบ ACL โดยการส่งข้อมูลที่เชื่อถือได้จากที่หนึ่งไปยังอีกที่หนึ่ง ฟังก์ชันการควบคุมต่างๆจะต้องเริ่มทำการเชื่อมต่อแบบ ACL เมื่อต้องการใช้งาน L2CAP และปิดการทำงานลงเมื่อเลิกใช้งาน L2CAP จะดูแลในเรื่องของคุณภาพของการให้บริการเมื่อมีการติดต่อกับเลเยอร์ที่อยู่ระดับบน

โปรแกรมประยุกต์ต่างๆจะต้องใช้ L2CAP ในการส่งข้อมูล รวมไปถึงเลเยอร์ที่อยู่ระดับบนของบลูทูธด้วย เช่น RFCOMM และ SDP ดังนั้น L2CAP จึงเป็นส่วนสำคัญที่จำเป็นในระบบบการทำงานแบบบลูทูธ

4.2 RFCOMM

พอร์ตอนุกรม RS-232 ประกอบด้วยขา 9 ขา ซึ่งใช้ในการรับส่งข้อมูลและส่งสัญญาณควบคุมต่างๆ RFCOMM เป็นโพรโตคอลที่ทำการจำลองการทำงานของสายเคเบิลที่รับส่งข้อมูลแบบอนุกรม และสถานะการทำงานของพอร์ตอนุกรม RS-232 นอกจากนี้ RFCOMM ยังรองรับการเชื่อมต่อหลายๆอันซึ่งถูกส่งมาจาก L2CAP ที่ทำการมัลติเพล็กข้อมูลการเชื่อมต่อต่างๆมาบนการเชื่อมต่อเพียงอันเดียวและจัดเตรียมการเชื่อมต่อไปยังอุปกรณ์หลายๆตัว

RFCOMM เชื่อถือการทำงานของเบสแบนโดยการรับส่งข้อมูลที่เชื่อถือได้ในรูปของสายข้อมูล ดังนั้นจึงไม่ต้องมีการแก้ไขข้อผิดพลาดที่เกิดขึ้น การควบคุมการไหลของข้อมูลใน RFCOMM จะถูกใช้ในชั้นของเบสแบนด้วย

ความเร็วในการรับส่งข้อมูลของ RFCOMM นี้ถูกจำกัดที่ตัวของอุปกรณ์ซึ่งเป็นการเชื่อมต่อทางกายภาพแบบพอร์ตอนุกรม (ชนิด 2) การนำไปใช้งานอาจใช้รับส่งข้อมูลบนพอร์ตจำลอง (ชนิด 1) ในการรับส่งข้อมูลนี้ RFCOMM จะส่งข้อมูลออกไปที่ความเร็วสูงสุดเท่าที่จะเป็นไปได้ ถึงแม้ว่าการส่งข้อมูลที่ความเร็วสูงสุดนี้จะทำให้เกิดประเด็นของความซับซ้อนที่เกิดขึ้นในการเชื่อมต่อแบบหลายๆการเชื่อมต่อ

RFCOMM ใช้งานง่าย, ทำงานบนโพรโตคอลที่เชื่อถือได้ และยังจัดการในเรื่องต่างๆดังนี้

- สถานะของโมเด็ม
- สถานะการเชื่อมต่อกับอุปกรณ์ระยะไกล
- การปรับแต่งค่าพอร์ตของอุปกรณ์ระยะไกล
- การเจรจาเพื่อปรับค่าพารามิเตอร์

4.2.1 พอร์ตอนุกรมและ UART

โดยทั่วไปแล้ว พอร์ตอนุกรมที่ใช้ในการส่งและรับข้อมูลจะถูกเชื่อมต่อกับ UART หน้าที่ของ UART คือ การเปลี่ยนระหว่างข้อมูลที่จะส่งไปเป็นแบบอนุกรมและข้อมูลที่ถูกประมวลผลแบบขนานที่อุปกรณ์ใช้งานอยู่ UART ใช้หน่วยความจำบัฟเฟอร์ในการเปลี่ยนระหว่างข้อมูลที่เป็นแบบอนุกรมและข้อมูลแบบขนาน ซึ่งจะช่วยลดภาระการทำงานของโพรเซสเซอร์ โดยแทนที่โพรเซสเซอร์จะถูกอินเทอร์รัพในข้อมูลทุกๆบิตที่ส่งมาจากสายเคเบิล แต่ UART จะส่งข้อมูลระหว่างสายเคเบิลกับบัฟเฟอร์ หลังจากนั้นโพรเซสเซอร์เพียงแค่ไปดึงข้อมูลมาจากบัฟเฟอร์เท่านั้น

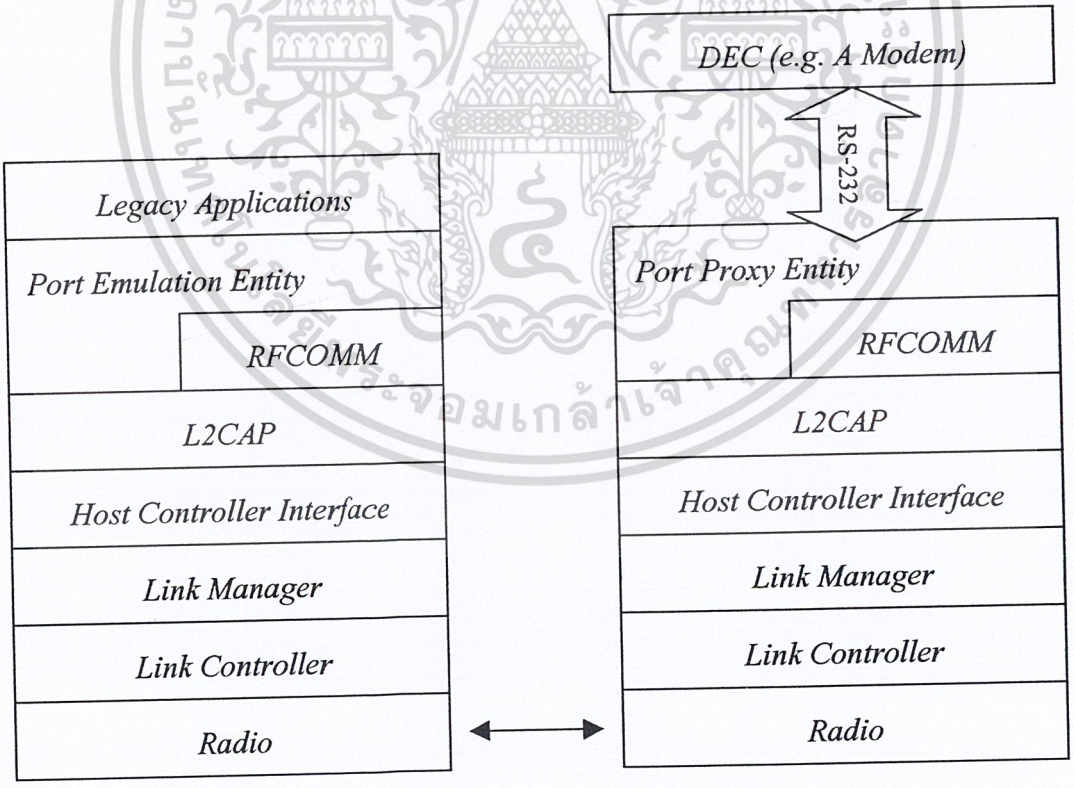
เมื่อสัญญาณจาก UART ถูกเชื่อมต่อ มันจะทำการเทียบเคียงกับแอดเดรสในระบบ โพรเซสเซอร์บางตัวจะสงวนช่วงแอดเดรสพิเศษไว้ใช้สำหรับ I/O ระบบอื่นๆสามารถเทียบเคียงแอดเดรสส่วนนี้กับหน่วยความจำของระบบได้ เนื่องจาก UART เปรียบเหมือนพื้นที่หน่วยความจำในมุมมองของไมโครโพรเซสเซอร์ ดังนั้นจึงเป็นไปได้ที่จะจำลองการทำงานของพอร์ตอนุกรมในซอฟต์แวร์โดยการใช้พื้นที่ของหน่วยความจำและปรับค่าต่างๆให้เหมือนกับการใช้ UART แบบปกติ

รายละเอียดบลูทูธในส่วนของ RFCOMM นั้นได้พูดถึงการจำลองการทำงานของพอร์ตอนุกรม RS-232 และระบุถึงวิธีที่ข้อมูลแบบสายอนุกรมจะถูกจำลองขึ้น แต่เนื่องจากข้อมูลแบบอนุกรมที่ผ่านพอร์ต RS-232 มานั้นถูกมองโดยไมโครโพรเซสเซอร์หลังจากที่ได้ผ่าน UART มาแล้ว ดังนั้นโปรแกรมที่ทำการติดต่อกับพอร์ตอนุกรมนั้นจะเหมือนทำงานกับข้อมูลที่เป็นแบบขนาน เช่นเดียวกัน ซอฟต์แวร์ RFCOMM จะทำงานกับข้อมูลแบบขนานที่ถูกส่งมาจากเลเยอร์ระดับล่างของบลูทูธโพรโตคอลแสตก

4.2.2 ชนิดของอุปกรณ์ RFCOMM

RFCOMM รองรับการทำงานของอุปกรณ์สองชนิด คือ

- ชนิด 1 – จำลองพอร์ตอนุกรม
- ชนิด 2 – อุปกรณ์เชื่อมต่อทางกายภาพด้วยพอร์อนุกรม



รูปที่ 4-2 อุปกรณ์ RFCOMM ชนิด 1 และ ชนิด 2

โพรโตคอลแสตกสำหรับอุปกรณ์ชนิดที่หนึ่งนั้น อุปกรณ์ RFCOMM จะอยู่ด้านซ้ายมือของรูปที่ 4-2 การจำลองพอร์ตจะถูกเทียบจากส่วนเชื่อมต่อที่ใช้ในการติดต่อสื่อสารของระบบไปยังการบริการ RFCOMM ซึ่งสามารถใช้เชื่อมต่อไปยังโปรแกรมประยุกต์ทั่วไปได้ ดังรูป หรือใช้ในการเชื่อมต่อไปยังโปรแกรมประยุกต์ที่ถูกเขียนขึ้นมาวำหรับบลูทูธก็เป็นได้ อุปกรณ์ชนิดที่หนึ่งนั้นมักจะเป็นอุปกรณ์ปลายทางของการติดต่อสื่อสาร เช่น คอมพิวเตอร์ หรือ พรินเตอร์ เป็นต้น

โพรโตคอลสแตกสำหรับอุปกรณ์ชนิดที่สองนั้น อุปกรณ์ RFCOMM จะแสดงอยู่ในรูปด้านขวามือของรูปที่ 4-2 อุปกรณ์ชนิดที่สองนี้มักเป็นอุปกรณ์ที่อยู่ตรงส่วนกลางของการเชื่อมต่อ เช่น โมเด็ม

4.2.3 ชนิดของเฟรมใน RFCOMM

RFCOMM อิงมาตรฐาน GSM TS 07.10 ซึ่งเป็นโพรโตคอลแบบไม่สมมาตรถูกใช้โดยโทรศัพท์เซลลูล่าร์แบบ GSM เพื่อใช้ในการมัลติเพล็กข้อมูลหลายๆข้อมูลบนการเชื่อมต่อเพียงอันเดียว RFCOMM เป็นโพรโตคอลแบบสมมาตร และส่งเฟรมแบบ TS 07.10 บนเลเยอร์ L2CAP โดยใช้คุณสมบัติเฟรมและคำสั่งของ TS 07.10 คุณสมบัติบางอย่างของ TS 07.10 ถูกนำมาปรับใช้เพื่อให้เหมาะสมกับบลูทูธ

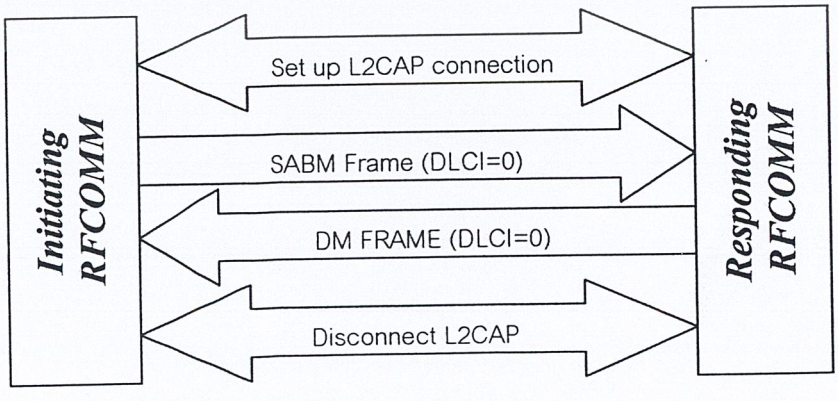
RFCOMM จะใช้เฟรมในการติดต่อสื่อสาร เฟรมใน RFCOMM จะกลายเป็นส่วนข้อมูลในแพ็กเกต L2CAP ซึ่งเฟรมมีด้วยกันห้าประเภท คือ

- SABM – Start Asynchronous Balance Mode (ใช้เริ่มต้นคำสั่ง)
- UA – Unnumbered Acknowledgement (ใช้ตอบกลับเมื่อมีการเชื่อมต่อ)
- DISC – Disconnect (คำสั่งในการหยุดการเชื่อมต่อ)
- DM – Disconnected Mode (ใช้ตอบกลับไปยังคำสั่งเมื่อมีการหยุดการเชื่อมต่อ)
- UIH – Unnumbered information with Header check

ใน GSM TS 07.10 จะมีเฟรม UI (Unnumbered Information) ด้วย ซึ่ง RFCOMM ไม่ได้นำมาใช้

4.2.4 การเชื่อมต่อและการหยุดการเชื่อมต่อ

เนื่องจากเฟรมของ RFCOMM นั้นจะถูกส่งไปเป็นแพ็กเกตข้อมูลของ L2CAP ดังนั้นการเชื่อมต่อของ L2CAP จะต้องเริ่มก่อนที่จะมีการเชื่อมต่อของ RFCOMM

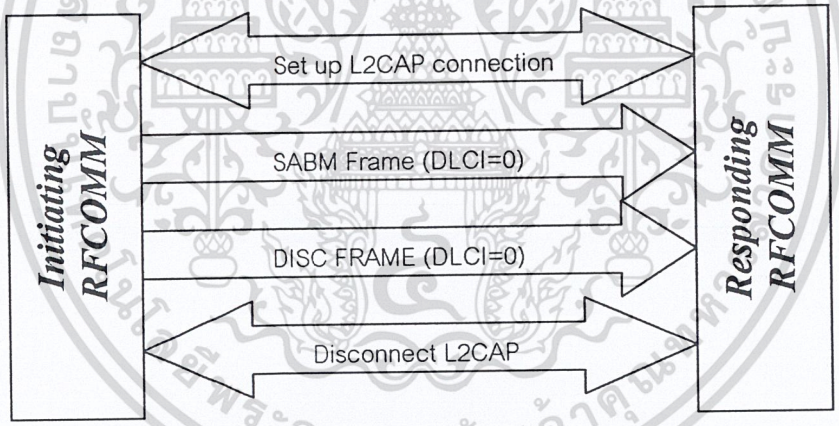


รูปที่ 4-3 การปฏิเสธการเชื่อมต่อ RFCOMM

RFCOMM เป็นโพรโตคอลที่สงวนไว้และค่าการบริการที่ใช้ในการมัลติเพล็กของ L2CAP ซึ่งกำหนดไว้ในมาตรฐานบลูทูธให้เป็น 0x0003 เฟรมของ L2CAP ใดๆที่รับค่านี้มาจะถูกส่งไปให้ RFCOMM เพื่อทำการประมวลผล

เฟรมแรกที่ถูกส่งไปบนแชนเนล RFCOMM คือ เฟรม SABM ซึ่งเป็นคำสั่งเริ่มต้นการทำงานในโหมดไม่สมมาตร ถ้าอุปกรณ์ RFCOMM ที่ต้องการจะเชื่อมต่อด้วย จะเข้าสู่โหมดไม่สมมาตรเช่นกัน และส่งเฟรม UA ตอบกลับมา ถ้าอุปกรณ์ RFCOMM ที่ไม่ต้องการจะเชื่อมต่อด้วยนั้น จะปฏิเสธการเชื่อมต่อโดยการส่งเฟรม DM ตอบกลับมา ดังรูปที่ 4-3

RFCOMM มีเวลา 60 s เมื่อคำสั่งถูกส่งออกไป ถ้าไม่ได้รับการตอบกลับภายในเวลาที่กำหนด ก็จะปิดการเชื่อมต่อทันที ซึ่งส่วนนี้จะแตกต่างจาก GSM 07.10 ซึ่งจะส่งคำสั่งไปอีกครั้งหนึ่งเมื่อเวลาหมดลงแล้ว ในกรณีของ RFCOMM นั้น เบสแบนของบลูทูธได้จัดเตรียมการเชื่อมต่อที่เชื่อถือได้ ดังนั้นถ้าคำสั่งที่ส่งออกไปครั้งแรกไม่ได้รับการตอบสนอง คำสั่งที่ส่งออกไปครั้งที่สองก็คงจะไม่ได้รับการตอบสนองเช่นเดียวกัน สำหรับคำสั่ง SABM นั้น เวลาที่ใช้คอยสามารถเพิ่มออกไปได้เนื่องจากกระบวนการที่เกี่ยวกับความปลอดภัยจะทำให้คำสั่งนี้ใช้เวลานานกว่าปกติ ถ้า RFCOMM หมดเวลาแล้วและต้องการปิดการเชื่อมต่อ ก็จะส่งเฟรมของคำสั่ง DISC ไป รูปที่ 4-4 แสดงการทำงานของแชนเนลเมื่อถูกปิดลงเนื่องจากเวลาหมด

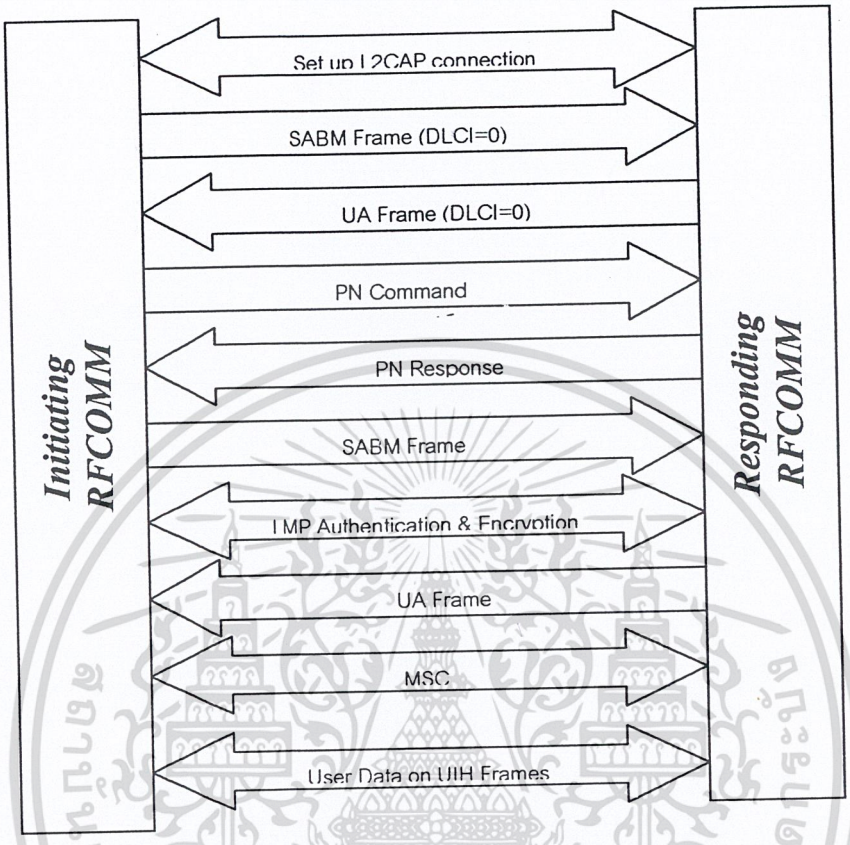


รูปที่ 4-4 แชนเนล RFCOMM เมื่อเวลาหมด

ถ้าการเชื่อมต่อประสบความสำเร็จ จะตอบกลับด้วยเฟรม SABM ที่มี เฟรม UA ด้วย ซึ่งจะตามด้วยคำสั่ง PN จากตัวเริ่มต้นและ PN ที่ตอบกลับจากอีกตัวหนึ่ง ดังรูปที่ 4-5

เมื่อการเชื่อมต่อด้วย DLCI=0 ได้ถูกเริ่มขึ้นแล้ว การส่งสัญญาณ RFCOMM ก็จะสามารถทำได้ในการส่งข้อมูลนั้น แชนเนลของ RFCOMM อื่นๆจะต้องถูกเริ่มต้นการทำงานก่อน รูปที่ 4-5 แสดงถึงแชนเนล RFCOMM ตัวที่สองที่เริ่มต้นการทำงานเพื่อจะส่งข้อมูล ในกรณีนี้แชนเนลจำเป็นต้องทำการพิสูจน์ตน ดังนั้นจะมีช่วงเวลาที่ต้องทำการพิสูจน์ตนและการเข้ารหัสของ LMP ระหว่างเฟรม SABM และเฟรมที่ตอบกลับ UA เมื่อเฟรม UA ได้รับตอบกลับมาแล้วนั้น คำสั่งที่เกี่ยวกับสถานะของโมเด็มจะถูกแลกเปลี่ยนเพื่อใช้ในการติดต่อสื่อสารถึงสถานะของสัญญาณควบคุม หลังจากนั้นข้อมูลจึงจะส่งผ่านติดต่อกัน

ได้ ดังรูป ในส่วนข้อมูลของผู้ใช้นั้นควรจะใส่คำสั่งที่เกี่ยวกับสถานะของโมเด็ม (MSC : Modem Status Command) ไปด้วย เพื่อใช้ในการติดต่อสื่อสารถึงสถานะของสัญญาณควบคุมการทำงานของพอร์ตอนุกรม



รูปที่ 4-5 ลำดับข้อความสำหรับการสร้างการเชื่อมต่อ RFCOMM

4.2.5 สรุปการทำงานของ L2CAP

RFCOMM เป็นการจำลองการทำงานของพอร์ตอนุกรม ซึ่งถูกใช้เชื่อมต่อกับโปรแกรมประยุกต์ต่างๆ และยังใช้ในการส่งข้อมูลระหว่างบลูทูธโพรไฟล์แต่ละตัวด้วย

RFCOMM รองรับการทำงานของอุปกรณ์สองชนิด คือ ชนิดที่หนึ่งสำหรับอุปกรณ์ที่อยู่ปลายทางของการเชื่อมต่อและรองรับโปรแกรมประยุกต์ต่างๆที่อยู่บนเลเยอร์ RFCOMM ด้วย ส่วนอุปกรณ์ชนิดที่สองนั้นใช้สำหรับอุปกรณ์ที่อยู่ตรงส่วนกลางของการติดต่อสื่อสารและมีการเชื่อมต่อทางกายภาพด้วย RS-232 บนเลเยอร์ของ RFCOMM

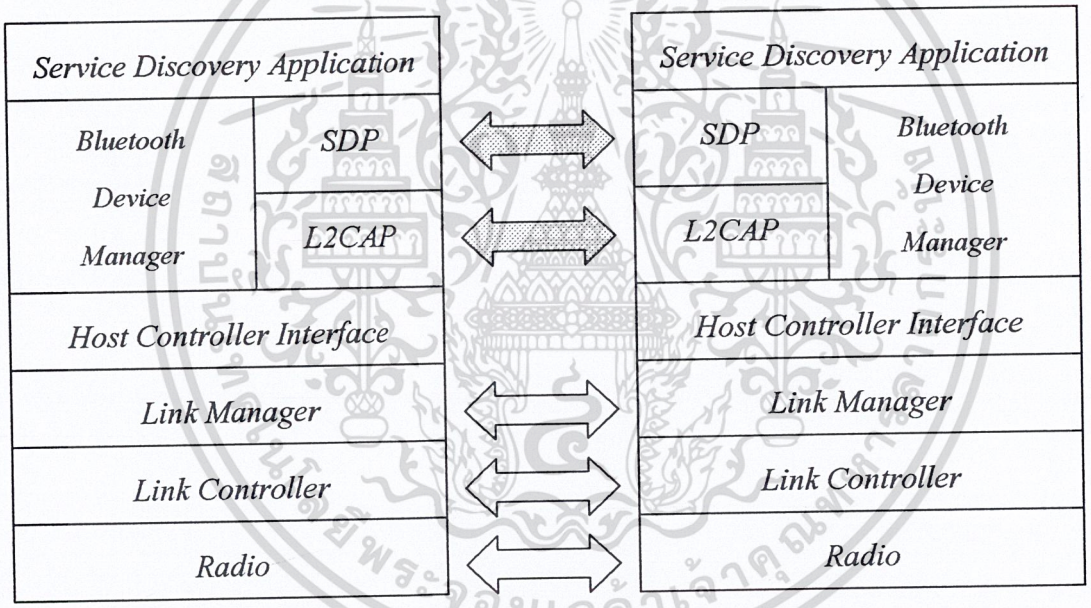
ในการเริ่มต้นการเชื่อมต่อแบบ RFCOMM นี้ การเชื่อมต่อ L2CAP จะต้องเริ่มขึ้นก่อนโดยที่เฟรมของ RFCOMM นั้นจะถูกส่งไปเป็นแพ็กเกตข้อมูลที่อยู่ในชั้นของ L2CAP เมื่อการเชื่อมต่อ L2CAP ได้เริ่มขึ้นแล้ว เฟรมควบคุม RFCOMM จะถูกส่งกลับและสร้างสัญญาณแชนเนลด้วย DLCI=0 หลังจากนี้แชนเนลย่อยจะถูกสร้างขึ้นมาเพื่อส่งผ่านข้อมูล ซึ่งสามารถมีได้มากสุดถึง 30 แชนเนลด้วยกัน

RFCOMM อิงมาตรฐานของ GSM 07.10 ซึ่งมีการเปลี่ยนแปลงเล็กน้อยเพื่อให้เหมาะสมกับการใช้งานในอุปกรณ์บลูทูธ

4.3 โพรโตคอลค้นหาการบริการ (SDP : Service Discovery Protocol)

ในบลูทูธพิโคเนตนั้นค่อนข้างที่จะแตกต่างจากลักษณะของ LAN ทั่วๆไป นอกจากจะต้องเชื่อมต่อไปยังเครือข่ายแล้วยังต้องเชื่อมต่อโดยตรงไปยังอุปกรณ์ด้วย การเชื่อมต่อสามารถเกิดการเปลี่ยนแปลงได้อย่างรวดเร็วในสภาพแวดล้อมที่ยืดหยุ่นนี้ ดังนั้นการทำงานแบบเก่าจึงไม่สามารถนำมาใช้ในสภาพแวดล้อมแบบนี้ได้

ใน LAN แบบทั่วไปนั้น เราสามารถที่จะเชื่อมต่อกับอุปกรณ์ต่างๆได้ เช่น พรินเตอร์ เมื่อเจออุปกรณ์ตัวนั้นแล้ว โดยทั่วไปมันจะอยู่อย่างนั้นเป็นเดือนหรือเป็นปี โดยที่ไม่มีการเปลี่ยนแปลง แต่สำหรับบลูทูธนั้นได้ถูกออกแบบมาเพื่อให้สามารถเดินไปยังพื้นที่ต่างๆเพื่อทำการค้นหาพรินเตอร์นั้น โดยที่ไม่จำเป็นต้องมีการปรับแต่งค่ามาก่อนแต่อย่างใด เมื่อใช้งานพรินเตอร์เสร็จเรียบร้อยแล้ว สามารถที่จะเดินออกไปโดยไม่ต้องสนใจรายละเอียดที่เหลือเลย โพรโตคอลค้นหาการบริการนี้เป็นส่วนหนึ่งของบลูทูธที่จะช่วยให้เหตุการณ์เหล่านี้สามารถเกิดขึ้นได้ SDP ได้จัดเตรียมวิธีการค้นหาอุปกรณ์ที่จะพิมพ์งานให้หรือทำการค้นหาการบริการที่มีอยู่ในระยะเครือข่ายว่ามีการบริการใดให้ใช้ได้บ้าง



รูปที่ 4-6 สถานะของ SDP ในบลูทูธโพรโตคอลแสตก

รูปที่ 4-6 แสดงสถานะของ SDP ในบลูทูธโพรโตคอลแสตก SDP ทำงานบนการเชื่อมต่อ L2CAP ซึ่งถูกสร้างขึ้นมาระหว่าง SDP ไคลเอ็นต์และ SDP เซิร์ฟเวอร์ เมื่อการเชื่อมต่อแบบ L2CAP ได้ถูกสร้างขึ้นแล้ว สามารถที่จะทำการค้นหาการบริการที่มีอยู่และวิธีที่จะใช้ติดต่อกับการบริการเหล่านั้น L2CAP ไม่ได้ทำการควบคุมการเชื่อมต่อเพื่อการใช้บริการ เพียงแต่เป็นตัวรวบรวมข้อมูลต่างๆ ดังนั้น ในการที่จะค้นหาการบริการและทำการเชื่อมต่อกับการบริการเหล่านั้นที่ให้ใช้โดย SDP เซิร์ฟเวอร์ ไคลเอ็นต์จะต้องทำตามลำดับดังต่อไปนี้

- สร้างการเชื่อมต่อแบบ L2CAP ขึ้นมายังอุปกรณ์ที่อยู่ระยะไกลโดยใช้แชนเนล PSM=0x0001
- ทำการค้นหาคลาสของการบริการที่กำหนด หรือค้นหาการบริการที่มีอยู่ทั้งหมด
- รับข้อมูลที่จำเป็นในการเชื่อมต่อกลับมาจากการบริการที่ได้เลือกไว้
- สร้างช่องทางการเชื่อมต่อที่จะใช้การบริการนั้น

แชนเนล L2CAP ที่ถูกใช้โดย SDP นั้นสามารถที่จะหยุดเมื่อไม่มีความจำเป็นต้องใช้อีกต่อไปแล้ว หรืออาจจะปล่อยไว้ในกรณีที่ไคลเอ็นต์ต้องการที่จะสอบถามการบริการจากเซิร์ฟเวอร์อีก

โพรโตคอล L2CAP และการมัลติเพล็กการบริการที่ใช้ใน SDP นั้นจะถูกกำหนดเป็น 0x0001 เมื่อทราบเลข PSM นี้แล้วหมายความว่าการเชื่อมต่อแบบ ACL ได้เกิดขึ้นแล้ว SDP ไคลเอ็นต์จะสามารถรู้เลขนี้โดยอัตโนมัติเพื่อใช้ในการสร้างการเชื่อมต่อไปยัง SDP เซิร์ฟเวอร์บนอุปกรณ์ที่อยู่ระยะไกล

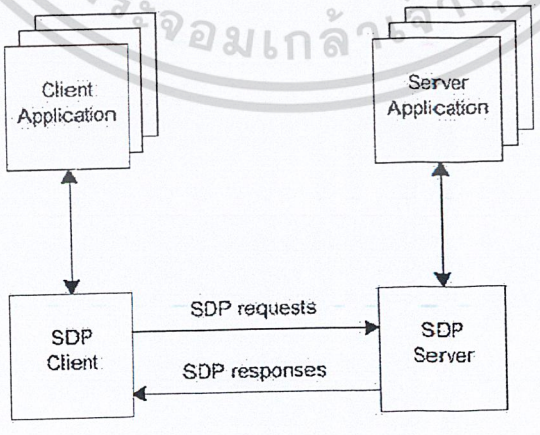
บลูทูธไม่ได้กำหนดรูปแบบที่ใช้ติดต่อกับผู้ใช้สำหรับการค้นหาการบริการนี้ เพียงแต่กำหนดถึงข้อมูลที่ใช้ในการแลกเปลี่ยนระหว่างเซิร์ฟเวอร์ที่เป็นผู้ให้บริการและไคลเอ็นต์ที่เป็นผู้ใช้บริการนั้น

4.3.1 โมเดลไคลเอ็นต์ – เซิร์ฟเวอร์ของ SDP

SDP เซิร์ฟเวอร์เป็นอุปกรณ์บลูทูธตัวไคๆก็ได้ที่ต้องการเสนอการบริการที่ตนมีอยู่ไปยังอุปกรณ์บลูทูธตัวอื่นๆ ข้อมูลเกี่ยวกับการบริการนั้นจะถูกเก็บไว้ในฐานข้อมูล SDP ซึ่งแต่ละ SDP เซิร์ฟเวอร์จะดูแลฐานข้อมูลของตนเอง ไม่มีฐานข้อมูลที่เป็นส่วนกลาง

SDP ไคลเอ็นต์ใช้การบริการที่จัดเตรียมให้โดยเซิร์ฟเวอร์ ในการที่จะทำให้ได้ดังนี้นั้น เซิร์ฟเวอร์และไคลเอ็นต์จำเป็นต้องมีการแลกเปลี่ยนข้อมูลข่าวสารที่เกี่ยวกับการบริการโดยใช้เรคอดของการบริการ

- อุปกรณ์ที่ต้องการค้นหาการบริการที่มีอยู่ในเครือข่าย เรียกว่าเป็น SDP ไคลเอ็นต์
- อุปกรณ์ที่ต้องการเสนอการบริการที่ตนเองมีอยู่ เรียกว่าเป็น SDP เซิร์ฟเวอร์ โดยที่อุปกรณ์ต่างสามารถเป็นได้ทั้งเซิร์ฟเวอร์และไคลเอ็นต์พร้อมกันได้



รูปที่ 4-7 การติดต่อระหว่างไคลเอ็นต์และเซิร์ฟเวอร์บน SDP

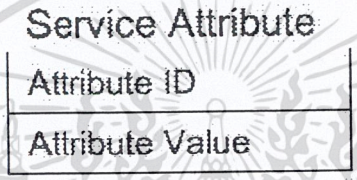
4.3.2 ฐานข้อมูล SDP

ฐานข้อมูล SDP ประกอบไปด้วยกลุ่มของเรคอดที่อธิบายถึงการบริการทั้งหมดที่อุปกรณ์บลูทูธตัวหนึ่งสามารถจะให้กับอุปกรณ์ตัวอื่นๆได้ SDP ได้จัดเตรียมวิธีที่อุปกรณ์ต่างๆจะมาหาเรคอดเหล่านี้ไว้ด้วย

4.3.2.1 แอททริบิวของการบริการ

การบริการ SDP ถูกอธิบายไว้โดยแอททริบิวของการบริการ ซึ่งได้เตรียมข้อมูลข่าวสารที่ไคลเอ็นต์จำเป็นต้องทราบในการใช้บริการต่างๆ

แต่ละแอททริบิวจะมีข้อมูล 16 บิตของ ID และข้อมูลการบริการ ดังรูปที่ 4-8 ข้อมูลในแอททริบิวเหล่านี้จะเป็นข้อความแบบสตริง บูลีน หรือเป็นเลขจำนวนเต็มก็ได้



รูปที่ 4-8 โครงสร้างแอทริบิวของการบริการ

แอททริบิวถูกใช้ในการส่งข้อมูลการบริการ แต่ละแอททริบิวในเรคอดนั้นจะอธิบายถึงการบริการชนิดต่างๆ ในเวอร์ชัน 1.0b ของมาตรฐานบลูทูธนั้นได้กำหนดชนิดของแอททริบิว ซึ่งมีอยู่ด้วยกัน 28 ชนิดดังนี้

- ServiceRecordHandle – เป็นเลขขนาด 32 บิตที่แตกต่างกัน เพื่อใช้เป็นแยกแยะเรคอดของการบริการต่างๆที่มีอยู่ในเซิร์ฟเวอร์
- ServiceClassIdList – ชนิดของการบริการในเรคอดของการบริการนี้
- ServiceRecorddState – เป็นเลข 32 บิตที่จะเปลี่ยนแปลงเมื่อแอททริบิวของการบริการในเรคอดมีการเปลี่ยนค่าไป
- ServiceID – เป็นเลขเฉพาะของแต่ละการบริการ การบริการชนิดเดียวกันจะมี ServiceID ที่ต่างกันถ้าอยู่บนคนละเซิร์ฟเวอร์
- ProtocolDescriptorList – เป็นโพรโตคอลที่จำเป็นในการใช้การบริการนี้
- BrowseGroupList – เป็นลิสต์ของกลุ่มเพื่อใช้ในการค้นหาการบริการทั้งหมดที่มีอยู่ในเครือข่าย
- LanguageBasedAttributeList – เป็นลิสต์ของภาษาที่เรคอดของการบริการรองรับ

- ServiceInfoTimeToLive – เป็นเลขจำนวนเต็มขนาด 32 บิตที่ใช้ประมาณเมื่อถึงเวลาที่เรคอดของการบริการจะมีการเปลี่ยนแปลง (เพื่อให้ไคลเอ็นต์สามารถที่จะเก็บข้อมูลในแคชและเชคเพื่อสอบถามถึงการเปลี่ยนแปลง)
- ServiceAvailability – เป็นเลขจำนวนเต็มขนาด 8 บิต ซึ่งจะบอกถึงจำนวนสูงสุดของไคลเอ็นต์ที่สามารถให้การบริการได้
- BluetoothProfileDescriptoList – เป็นลิสต์ของบลูทูธโพรไฟล์ที่ใช้ในการบริการนี้
- DocumentationURL – เป็น URL สำหรับเอกสารที่บอกรายละเอียดของการบริการนี้
- ClientExecutableURL – เป็น URL สำหรับการทำงานในการบริการนี้
- IconURL – เป็น URL สำหรับไอคอนที่เป็นส่วนแสดงผลต่อผู้ใช้ในการใช้การบริการนี้
- ServiceName – ชื่อของการบริการ
- ServiceDescription – รายละเอียดที่อธิบายถึงการบริการนี้
- ProviderName – ชื่อของผู้สร้างการบริการนี้
- VersionNumberList – ลิสต์ของเวอร์ชันที่รองรับ
- ServiceDatabaseSet – เป็นเลขจำนวนเต็มขนาด 32 บิต ซึ่งจะเปลี่ยนเมื่อเรคอดของการบริการเปลี่ยนไป
- GroupID – ค่าที่ใช้บอกกลุ่มของการบริการ ใช้ในการค้นหาการบริการทั้งหมดที่มีอยู่ในเครือข่าย
- RemoteAudioVolumeControl – ใช้กับหูฟังไร้สายเมื่อต้องการปรับความดังของเสียง
- External Network – ใช้กับโพรโตคอลโทรศัพท์ไร้สายเพื่อบ่งบอกถึงชนิดเครือข่ายของระบบโทรศัพท์ที่เชื่อมต่ออยู่
- Service Version – เวอร์ชันของการบริการ
- Supported Data Stores List – ใช้กับโพรไฟล์การเข้าจังหวะข้อมูล
- Supported Formats List – ใช้ในโพรไฟล์การส่งข้อมูลแบบออบเจ็กต์ ซึ่งเป็นรูปแบบข้อมูลของออบเจ็กต์ที่ใช้ในการส่ง
- Fax Class 1 Support – ใช้ในโพรไฟล์แฟกซ์ เป็นค่าบูลีนเมื่อรองรับการทำงานของแฟกซ์คลาส 1
- Fax Class 2.0 Support – ใช้ในโพรไฟล์แฟกซ์ เป็นค่าบูลีนเมื่อรองรับการทำงานของแฟกซ์คลาส 2.0
- Fax Class 2 Support – ใช้ในโพรไฟล์แฟกซ์ เป็นค่าบูลีนเมื่อรองรับการทำงานของแฟกซ์คลาส 2
- Audio Feedback Support – ใช้กับโพรไฟล์การเชื่อมต่อไปยังเครือข่าย เป็นค่าบูลีนเมื่อการย้อนกลับของข้อมูลเสียงถูกใช้ในแชนเนล SCO ในระหว่างการเริ่มต้นการเชื่อมต่อ

4.3.2.2 ชนิดและขนาดของข้อมูล

ภายในแอททริบิวจะมีค่าของข้อมูล ซึ่งค่าเหล่านี้มีหลายชนิดและหลายขนาด ดังนั้นอุปกรณ์ที่ได้รับแอททริบิวมาต้องทราบชนิดและขนาดที่รับมาด้วย เมื่อแอททริบิวถูกส่งออกไปจะมีส่วนที่บอกถึงชนิดและขนาดของแอททริบิวที่ส่งไป โดยข้อมูลไบต์แรกจะบอกถึงชนิดของข้อมูล และสามบิตต่อมาจะบอกถึงขนาดของข้อมูล

ส่วนที่บอกชนิดของข้อมูลนั้น ประกอบไปด้วยชนิดต่างๆดังนี้

- 0 – NULL
- 1 – เลขจำนวนเต็มไม่คิดเครื่องหมาย
- 2 – เลขจำนวนเต็มแบบ Signed 2's complement
- 3 – UUID (Universally Unique Identifier)
- 4 – ข้อความหรือสตริง
- 5 – บูลีน
- 6 – ลำดับของข้อมูลทั้งหมด
- 7 – ลำดับของข้อมูลที่ถูกเลือก
- 8 – URL (Uniform Resource Locator)

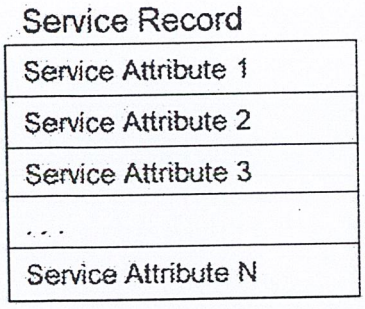
สำหรับส่วนที่บอกขนาดของข้อมูลนั้น ประกอบไปด้วยค่าต่างๆดังนี้

- 0 – 1 ไบต์หรือ 0 ไบต์ ในกรณีที่เป็น NULL
- 1 – 2 ไบต์
- 2 – 4 ไบต์
- 3 – 8 ไบต์
- 4 – 16 ไบต์
- 5 – ขนาดของข้อมูลอยู่ถัดไปอีก 1 ไบต์
- 6 – ขนาดของข้อมูลอยู่ถัดไปอีก 2 ไบต์
- 7 – ขนาดของข้อมูลอยู่ถัดไปอีก 4 ไบต์

4.3.2.3 เรคอดของการบริการ

แอททริบิวใช้ในการบอกถึงลักษณะต่างๆของการบริการที่เซิร์ฟเวอร์มีให้ใช้ เรคอดของการบริการจึงเป็นส่วนที่เก็บข้อมูลต่างๆเหล่านี้ไว้เพื่อใช้ในการอธิบายถึงรายละเอียดต่างๆของการบริการที่มีอยู่แต่ละชนิด

ตารางที่ 4-1 เป็นตัวอย่างเรคอดของการบริการหูฟังไร้สาย ซึ่งจะแสดงให้เห็นถึงวิธีที่เรคอดของการบริการถูกสร้างขึ้น



รูปที่ 4-9 เรคอดของการบริการ

ในส่วนแรกของเรคอดการบริการนั้นจะเกี่ยวข้องกับการบริการที่ถูกเสนอโดยอุปกรณ์นั้นๆ การบริการหูฟังไร้สายประกอบไปด้วยการบริการสองอย่าง คือ การบริการหูฟังไร้สายและการบริการข้อมูลเสียงแบบทั่วไป โดยที่การบริการหูฟังไร้สายจะเป็นตัวหลัก

ในส่วนต่อมาของเรคอดการบริการนี้จะเป็นลิสต์ของโพรโตคอลที่จำเป็นในการใช้การบริการนี้ การบริการหูฟังไร้สายจำเป็นต้องใช้เลเยอร์ L2CAP และ RFCOMM เลขแชนเนลของ RFCOMM เซิร์ฟเวอร์นี้จำเป็นต้องทราบก่อนที่อุปกรณ์จะใช้ติดต่อสื่อสารผ่าน RFCOMM ดังนั้นในส่วนที่อธิบายถึงโพรโตคอลที่ใช้ RFCOMM จะต้องระบุถึงพารามิเตอร์ที่บอกถึงหมายเลขของแชนเนล เพื่อให้ไคลเอ็นต์นำเลขนี้ไปใช้ในการติดต่อผ่าน RFCOMM

ส่วนต่อมาเป็นส่วนที่อธิบายถึงโพรไฟล์ที่รองรับ โพรไฟล์ที่อยู่ในลิสต์นั้นควรเป็นโพรไฟล์ที่อุปกรณ์นั้นรองรับการทำงานอย่างเต็มที่ ดังนั้นจึงเป็นไปได้ที่จะไม่มีโพรไฟล์ใดเลยในเรคอดของการบริการบางชนิด ดังนั้นในการบริการหูฟังไร้สายนี้จะรองรับโพรไฟล์ของหูฟังไร้สายเช่นเดียวกัน ตามมาด้วยหมายเลขของเวอร์ชันและชื่อของการบริการ

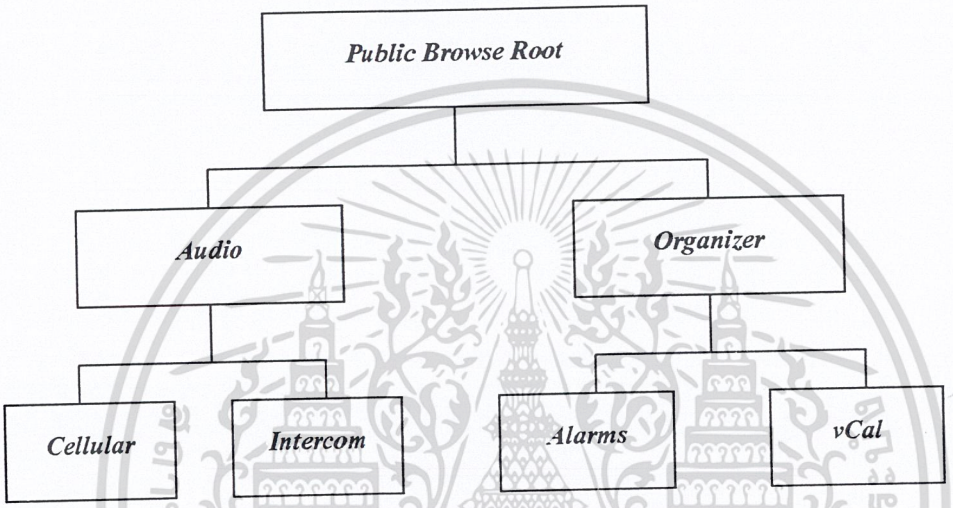
Item	Type	Value	Attribute ID
ServiceRecordHandle	Uint32	Assigned by Server	0x0000
ServiceClassIDList			0x0001
ServiceClass0	UUID	Headset	0x1108
ServiceClass1	UUID	Generic Audio	0x1203
ProtocolDescriptorList			0x0004
Protocol0	UUID	L2CAP	0x0100
Protocol1	UUID	RFCOMM	0x0003
ProtocolSpecificParameter0	Uint8	Server Channel	
BluetoothProfileDescriptorList			0x0009
Profile0	UUID	Headset	0x1108
Parameter0	Uint16	Version 1.0	0x0100
ServiceName	String	"Headset	0x0000+offset
Remote Audio Volume Control	Boolean	False	0x0302

ตารางที่ 4-1 เรคอดของการบริการหูฟังไร้สายบลูทูธ

4.3.3 การค้นหาเรคอด SDP

ในการที่จะทำให้การค้นหาการบริการเป็นเรื่องง่ายนั้น การบริการจะถูกจัดเรียงเป็นลักษณะของโครงสร้างลำดับแบบต้นไม้ ไคลเอ็นต์เริ่มตรวจสอบจากรากของโครงสร้างก่อน หลังจากนั้นจึงค้นหาต่อไปยังโหนดต่างๆ ซึ่งแต่ละการบริการจะถูกจัดเตรียมไว้

ในการจัดเตรียมในส่วนนี้ขึ้นอยู่กับผู้ให้บริการการบริการว่าโครงสร้างการบริการนั้นจะถูกสร้างแบบใด ตัวอย่างเช่น รูปที่ 4-10 แสดงถึงโครงสร้างการบริการของโทรศัพท์มือถือซึ่งได้จัดเตรียมการบริการไว้หลายอย่างด้วยกันกับอุปกรณ์บลูทูธตัวอื่นๆ ซึ่งสามารถแสดงเป็นโครงสร้างได้ดังรูป



รูปที่ 4-10 โครงสร้าง SDP สำหรับโทรศัพท์มือถือ

ผู้ให้บริการได้แบ่งการบริการออกเป็นสองกลุ่มด้วยกัน คือ กลุ่มออดิโอเป็นการบริการเกี่ยวกับข้อมูลที่เป็นเสียง ซึ่งอนุญาตให้หูฟังไร้สายบลูทูธทำการเชื่อมต่อกับโทรศัพท์มือถือ และเชื่อมต่อกับอินเตอร์คอม ส่วนกลุ่มของการจัดการนั้นได้รวมความสามารถของปฏิทิน ซึ่งรวมถึงความสามารถในการทำข้อมูลที่การนัดหมายในปฏิทินให้ตรงกันและการปลุก

4.3.4 การค้นหากลุ่ม

คลาสของการบริการถูกใช้เพื่อแยกแยะการบริการแต่ละชนิด แต่ละคลาสของการบริการนั้นได้รวมถึงแอททริบิว BrowseGroupList ข้อมูลใน BrowseGroupList จะเป็นลิสต์ UUID ของการค้นหาแบบกลุ่มที่เกี่ยวกับการบริการนั้น

ตารางที่ 4-2 แสดงถึงกลุ่มของเรคอดการบริการที่ใช้ในการค้นหาแบบกลุ่มของรูปที่ 4-10 ส่วนบนสุดของการค้นหาเรียกว่า PublicBrowseRoot ซึ่งมีค่า UUID เป็น 0x1002 (ซึ่งเป็นรูปแบบสั้น ส่วนรูปแบบยาว คือ 0x00001002-0000-1000-7007-00805F9B34FB) ในการค้นหาแบบกลุ่มจะมีส่วนนี้เป็นส่วนที่อยู่บนสุด เนื่องจากไม่มีข้อมูลใดๆที่อยู่บนข้อมูลนี้แล้ว

Service/Group Name	Service Class	Attribute Name	Attribute Value
Audio	BrowseGroupDescriptor	BrowseGroupList	PublicBrowseRoot
		GroupID	AudioID
Organizer	BrowseGroupDescriptor	BrowseGroupList	PublicBrowseRoot
		GroupID	OrganizerID
Cellular	BrowseGroupDescriptor	BrowseGroupList	AudioID
Intercom	BrowseGroupDescriptor	BrowseGroupList	AudioID
Alarms	BrowseGroupDescriptor	BrowseGroupList	OrganizerID
vCalendar	BrowseGroupDescriptor	BrowseGroupList	OrganizerID

ตารางที่ 4-2 เรคอดการบริการในการค้นหาแบบกลุ่มสำหรับโทรศัพท์มือถือ

4.3.5 UUID

UUID เป็นเลขขนาด 128 บิต แต่ละเรคอดจะมีแอททริบิวของ UUID เพื่อใช้ในการค้นหาเรคอด มาตรฐานบลูทูธได้กำหนดการใช้แอททริบิว UUID เพื่อใช้สำหรับโพรไฟล์บลูทูธ

4.3.5.1 UUID ที่ถูกกำหนดโดยมาตรฐานบลูทูธ

ในมาตรฐานบลูทูธนั้นมีการใช้งาน UUID ค่อนข้างมาก ดังนั้นเพื่อเป็นการลดแบนวิธที่ใช้ในการส่งและเพื่อเป็นการประหยัดพื้นที่ที่ใช้ในการเก็บข้อมูล จึงสามารถใช้งานในรูปสั้นเป็นขนาด 32 บิตหรือ 16 บิตได้

ในการเปรียบเทียบ UUID แต่ละคู่นั้นจะต้องเปลี่ยนให้อยู่ในรูปแบบเดียวกันก่อน โดยกฎการเปลี่ยนรูปแบบนี้ถูกกำหนดไว้ในมาตรฐานบลูทูธ

การเปลี่ยนจาก UUID ขนาด 16 บิตเป็นขนาด 32 บิต ทำได้โดยการเติม 0 ต่อท้าย ตัวอย่างเช่น UUID ขนาด 16 บิต 0x1265 จะเปลี่ยนเป็น UUID ขนาด 32 บิต 0x00001265 เป็นต้น

การเปลี่ยนจาก UUID ขนาด 16 บิตหรือขนาด 32 บิตไปเป็นขนาด 128 บิต ทำได้โดยการเติมคูณด้วย 2^{96} และบวกด้วยบลูทูธเบส UUID ตัวอย่างเช่น

$0x1265*2^{96}$+0x00000000-0000-1000-7007-00805F9B34FB = 0x00001265-0000-1000-7007-00805F9B34FB เป็นต้น

4.3.5.2 UUID ที่ถูกกำหนดโดยผู้ผลิต

ถ้าแหล่งที่ผลิตต้องการที่จะกำหนดการบริการขึ้นมาใหม่นั้น สามารถทำได้โดยการกำหนด UUID ของตนเองขึ้นมา SDP ใช้วิธีในการกำหนด UUID ซึ่งทำให้ไม่ซ้ำกัน ซึ่งใน UUID นั้นจะมีส่วนที่เป็นโหนด ในส่วนนี้จะเป็นส่วนที่ไม่มีทางซ้ำกันได้ โดยการใช้เลข IEEE ขนาด 48 บิต ทำให้สามารถรับประกันได้ว่า UUID ที่สร้างขึ้นมาไม่มีโอกาสที่จะซ้ำกัน

4.3.6 ข้อความใน SDP

ในการค้นหาคลาสของการบริการหรือการรับข้อมูลข่าวสารเกี่ยวกับการบริการที่ต้องการค้นหานั้น SDP ไคลเอ็นต์และเซิร์ฟเวอร์จะต้องทำการแลกเปลี่ยนข้อความกัน โดยข้อความเหล่านี้จะส่งไปโดยผ่าน SDP PDU (Protocol Data Units) ซึ่งมีอยู่ด้วยกัน 7 ประเภท แต่ละประเภทจะมี PDU ID ของตนเองดังนี้

- 0x01 = SDP_ErrorResponse
- 0x02 = SDP_ServiceSearchRequest
- 0x03 = SDP_ServiceSearch_Response
- 0x04 = SDP_ServiceAttributeRequest
- 0x05 = SDP_ServiceAttributeResponse
- 0x06 = SDP_ServiceSearchAttributeRequest
- 0x07 = SDP_ServiceSearchAttributeResponse

โดยไคลเอ็นต์จะเป็นตัวที่เริ่มส่งการร้องขอออกไป และเซิร์ฟเวอร์จะเป็นตัวที่ตอบกลับมา

4.3.6.1 การค้นหาการบริการ



รูปที่ 4-11 การค้นหาการบริการ

SDP ไคลเอ็นต์จะสร้าง SDP_ServiceSearchRequest เพื่อค้นหาเรคอดการบริการที่ตรงกับรูปแบบการบริการที่ส่งออกไปใช้ในการค้นหา ซึ่งเป็นพารามิเตอร์ตัวแรกของ PDU เมื่อฝั่งรับได้รับคำร้องขอแล้วจะตรวจสอบเรคอดการบริการของตนที่มีอยู่ในฐานข้อมูล และส่ง SDP_ServiceSearchResponse กลับ ซึ่งประกอบด้วยตัวควบคุมเรคอดการบริการที่ตรงกันกลับรูปแบบการบริการที่ส่งออกไป โดยพารามิเตอร์ของ SDP_ServiceSearchRequest มีดังนี้

- ServiceSearchPattern – เป็นลำดับของกลุ่มข้อมูล ซึ่งข้อมูลแต่ละตัวนั้นเป็นค่า UUID ค่าต่ำสุดคือหนึ่ง และค่ามากสุดที่เป็นไปได้เท่ากับสิบสอง
- MaximumServiceRecordCount – เป็นเลข 16 บิตที่ใช้กำหนดจำนวนมากสุดของตัวควบคุมเรคอดการบริการที่ตอบกลับมา

- ContinuationState – เป็นเลข 8 บิต ใช้เมื่อขนาดของคำสั่งที่ส่งไปมีขนาดใหญ่กว่าขนาดของหนึ่งแพ็กเกต ดังนั้นจึงต้องแบ่งคำสั่งออกเป็นหลายๆแพ็กเกตย่อย

ส่วนค่าพารามิเตอร์ของ SDP_ServiceSearchResponse มีดังนี้

- TotalServiceRecordCount – เป็นเลขจำนวนเต็มที่บอกถึงจำนวนเรคอดการบริการที่ตรงกับรูปแบบการบริการที่ร้องขอมา
- CurrentServiceRecordCount – เป็นเลขจำนวนเต็มที่บอกถึงจำนวนของตัวควบคุมเรคอดการบริการที่อยู่ในพารามิเตอร์ถัดไป
- ServiceRecordHandleList – เป็นลิสต์ของตัวควบคุมเรคอดการบริการ
- ContinuationState – เป็นเลข 8 บิต ใช้เมื่อขนาดของคำสั่งที่ส่งไปมีขนาดใหญ่กว่าขนาดของหนึ่งแพ็กเกต ดังนั้นจึงต้องแบ่งคำสั่งออกเป็นหลายๆแพ็กเกตย่อย

4.3.6.2 การค้นหาแอททริบิวของการบริการ



รูปที่ 4-12 การค้นหาแอททริบิวของการบริการ

SDP ไคลเอ็นต์จะสร้าง SDP_ServiceAttributeRequest ออกไปเพื่อรับค่าแอททริบิวที่ต้องการกลับมาจากเรคอดการบริการที่ระบุไป โดยตัวเรคอดการบริการที่ระบุไปนั้นจะเป็นการส่งค่าตัวควบคุมเรคอดการบริการที่ได้รับมาจากคำสั่งก่อนหน้านี้ โดยพารามิเตอร์ของ SDP_ServiceAttributeRequest มีดังนี้

- ServiceRecordHandle – เป็นตัวรับถึงเรคอดการบริการที่ต้องการจะรับข้อมูลในแอททริบิวที่ส่งออกไป
- MaximumAttributeByteCount – ระบุถึงจำนวนสูงสุดของข้อมูลที่ตอบกลับมา
- AttributeIDList – เป็นลำดับของข้อมูล ซึ่งแต่ละค่าในข้อมูลนั้นจะเป็นแอททริบิว ID
- ContinuationState – เป็นเลข 8 บิต ใช้เมื่อขนาดของคำสั่งที่ส่งไปมีขนาดใหญ่กว่าขนาดของหนึ่งแพ็กเกต ดังนั้นจึงต้องแบ่งคำสั่งออกเป็นหลายๆแพ็กเกตย่อย

ส่วนค่าพารามิเตอร์ของ SDP_ServiceAttributeResponse มีดังนี้

- AttributeListByteCount – เป็นจำนวนของข้อมูลที่ตอบกลับมา

- AttributeList – เป็นลำดับข้อมูลที่ตอบกลับมา ซึ่งประกอบด้วยแอททริบิว ID และข้อมูลในแอททริบิวนั้น
- ContinuationState – เป็นเลข 8 บิต ใช้เมื่อขนาดของคำสั่งที่ส่งไปมีขนาดใหญ่กว่าขนาดของหนึ่งแพ็กเกต ดังนั้นจึงต้องแบ่งคำสั่งออกเป็นหลายๆแพ็กเกตย่อย

4.3.6.3 การค้นหาการบริการและแอททริบิวของการบริการ

ในคำสั่ง SDP_ServiceSearchAttributeRequest นี้จะเป็นการรวมความสามารถของคำสั่งสองคำสั่งข้างต้น คือ SDP_ServiceSearchRequest และ SDP_ServiceAttributeRequest ไว้ด้วยกันภายในคำสั่งเดียว ซึ่งข้อดีของการใช้คำสั่งนี้แทนที่จะแยกเป็นสองคำสั่ง คือ ลดจำนวนของทรานแซคชันที่เกิดขึ้น เมื่อมีการรับ-ส่งหลายๆเรคอดของการบริการ ซึ่งมีพารามิเตอร์ต่างๆดังนี้



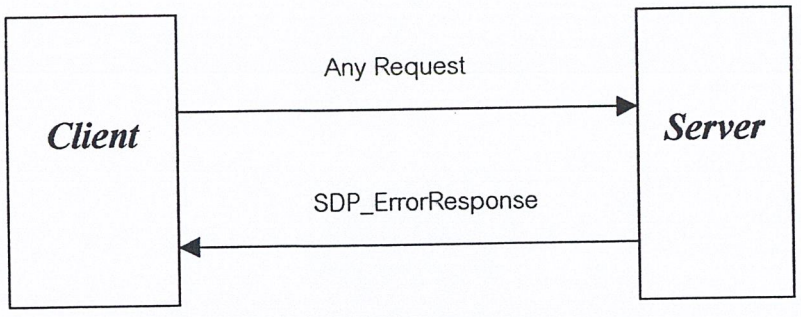
รูปที่ 4-13 การค้นหาการบริการและแอททริบิวของการบริการ

- ServiceSearchPattern – เป็นลำดับของกลุ่มข้อมูล ซึ่งข้อมูลแต่ละตัวนั้นเป็นค่า UUID ค่าต่ำสุดคือหนึ่ง และค่ามากสุดที่เป็นไปได้เท่ากับสิบสอง
- MaximumAttributeByteCount – ระบุถึงจำนวนสูงสุดของข้อมูลที่ตอบกลับมา
- AttributeIDList – เป็นลำดับของข้อมูล ซึ่งแต่ละค่าในข้อมูลนั้นจะเป็นแอททริบิว ID
- ContinuationState – เป็นเลข 8 บิต ใช้เมื่อขนาดของคำสั่งที่ส่งไปมีขนาดใหญ่กว่าขนาดของหนึ่งแพ็กเกต ดังนั้นจึงต้องแบ่งคำสั่งออกเป็นหลายๆแพ็กเกตย่อย

ส่วนค่าพารามิเตอร์ของ SDP_ServiceSearchAttributeResponse คือ

- AttributeListByteCount – เป็นจำนวนของข้อมูลที่ตอบกลับมา
- AttributeList – เป็นลำดับข้อมูลที่ตอบกลับมา ซึ่งประกอบด้วยแอททริบิว ID และข้อมูลในแอททริบิวนั้น
- ContinuationState – เป็นเลข 8 บิต ใช้เมื่อขนาดของคำสั่งที่ส่งไปมีขนาดใหญ่กว่าขนาดของหนึ่งแพ็กเกต ดังนั้นจึงต้องแบ่งคำสั่งออกเป็นหลายๆแพ็กเกตย่อย

4.3.6.4 ตัวควบคุมข้อผิดพลาด



รูปที่ 4-14 ตัวควบคุมข้อผิดพลาด

ถ้า SDP ในฝั่งของเซิร์ฟเวอร์ได้รับการร้องขอที่ไม่สามารถตอบกลับได้อย่างถูกต้อง เซิร์ฟเวอร์จะส่ง SDP_ErrorResponse กลับไป ซึ่งประกอบด้วยพารามิเตอร์สองตัวคือ โค้ดของความผิดพลาดและข้อมูลความผิดพลาดนั้น สำหรับโค้ดของความผิดพลาดสามารถเป็นได้หกค่าดังนี้ คือ

- 0x0001 – SDP ไคลเอ็นต์ส่งคำสั่งผิดพลาด
- 0x0002 – คำร้องขอประกอบด้วยตัวควบคุมเรคอดการบริการที่ผิดพลาด
- 0x0003 – คำร้องขอส่งมาผิดรูปแบบ
- 0x0004 – ขนาด PDU ของคำร้องขอผิดพลาด
- 0x0005 – ContinuationState ที่ส่งมาผิดพลาด
- 0x0006 – เซิร์ฟเวอร์ไม่มีทรัพยากรเพียงพอที่จะตอบกลับการร้องขอที่ส่งมา

4.3.7 โพรไฟล์การค้นหาการบริการ

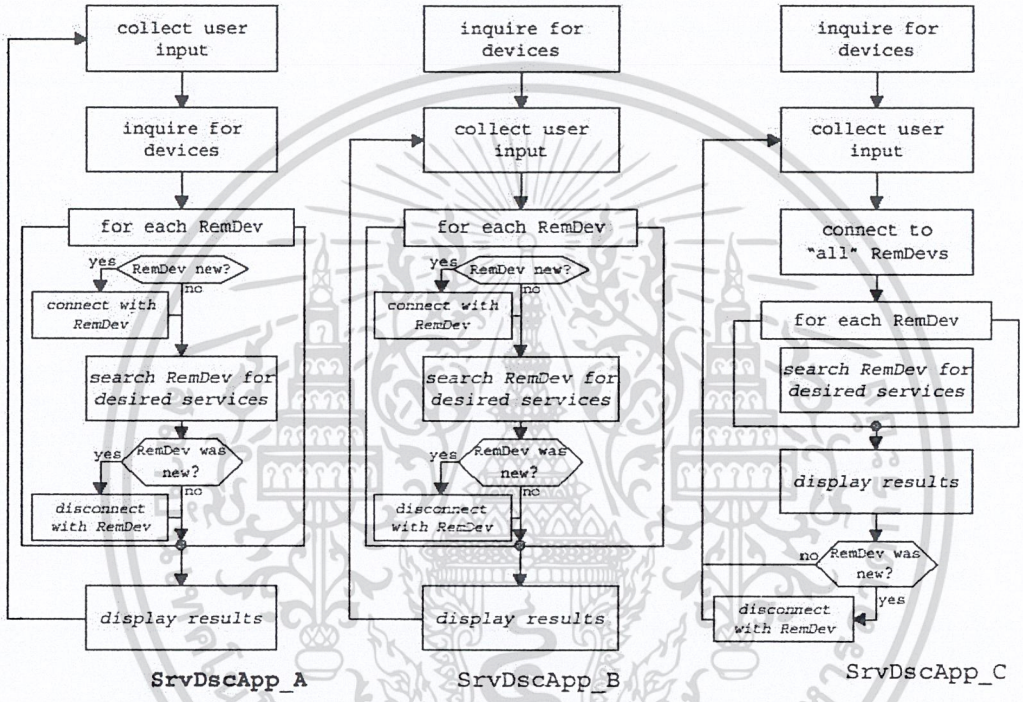
โพรไฟล์การค้นหาการบริการได้อธิบายถึงวิธีที่โปรแกรมประยุกต์ที่ทำงานอยู่บน SDP ไคลเอ็นต์จะใช้ SDP และคุณสมบัติอื่นของบลูทูธโพรโตคอลแสตกในการค้นหาการบริการที่มีอยู่ในบนอุปกรณ์ในเครือข่าย โพรไฟล์การค้นหาการบริการได้จัดเตรียมสิ่งต่างๆ ดังนี้

- กลุ่มคำสั่งที่โปรแกรมประยุกต์ใช้ในการค้นหาการบริการ
- ตัวอย่างที่เป็นโครงร่างของการทำงาน ซึ่งจะเป็นลำดับที่แสดงให้เห็นถึงขั้นตอนการทำงานในการที่โปรแกรมประยุกต์จะทำการค้นหาอุปกรณ์ เชื่อมต่อและค้นหาการบริการที่มีอยู่
- ลำดับของข้อความที่แสดงสถานะการเชื่อมต่อแบบ SDP
- แสดงรายชื่อของคุณสมบัติที่ต้องการใน SDP, L2CAP, LMP และส่วนควบคุมการเชื่อมต่อ

4.3.8 โปรแกรมประยุกต์การค้นหาการบริการ

โครงร่างการทำงานของโปรแกรมประยุกต์การค้นหาการบริการสามารถเป็นได้ดังรูปที่ 4-15 ในรูปจะเป็นโครงร่างการทำงานของโปรแกรมประยุกต์การค้นหาการบริการที่สามารถเป็นไปได้ แต่ละวิธีจะ

ให้ผลลัพธ์ที่เหมือนกัน เพียงแต่วิธีทำต่างกันเท่านั้น ในรูปแบบแรก (SrvDscApp_A) นั้นโปรแกรมประยุกต์จะทำการอินไควด้วยข้อมูลที่ผู้ใช้กำหนดมาในการค้นหาการบริการที่กำหนด จากนั้นโปรแกรมจะทำการค้นหาอุปกรณ์ เมื่อเจออุปกรณ์แล้วจะทำการเชื่อมต่อเพื่อค้นหาการบริการที่ต้องการ ในรูปแบบที่สอง (SrvDscApp_B) นั้นจะทำการอินไควเพื่อค้นหาอุปกรณ์ก่อนที่จะเก็บข้อมูลจากผู้ใช้เพื่อใช้ในการค้นหาการบริการ ในสองวิธีแรกนี้การเพจ, การสร้างการเชื่อมต่อและการค้นหาการบริการจำทำเป็นลำดับกับอุปกรณ์ทีละตัว ส่วนรูปแบบสุดท้าย (SrvDscApp_C) นั้น จะทำการเชื่อมต่อไปยังอุปกรณ์ทั้งหมดก่อน (เชื่อมต่อได้ครั้งละเจ็ดตัวในคราวเดียวกัน) หลังจากนั้นจึงค่อยทำการค้นหาการบริการที่ต้องการ



รูปที่ 4-15 โครงร่างของโปรแกรมประยุกต์การค้นหาการบริการที่เป็นไปได้

4.3.9 สรุปการทำงานของโพรโตคอลค้นหาการบริการ

โพรโตคอลค้นหาการบริการเป็นการที่ SDP ไคลเอ็นต์ทำการเข้าถึงข้อมูลการบริการที่อยู่ใน SDP เซิร์ฟเวอร์ เซิร์ฟเวอร์เป็นอุปกรณ์บลูทูธตัวใดก็ได้ที่เสนอการบริการที่ตนเองมีอยู่ให้กับอุปกรณ์บลูทูธตัวอื่นๆใช้ ส่วนไคลเอ็นต์เป็นอุปกรณ์บลูทูธใดๆก็ตามที่ต้องการใช้การบริการ ดังนั้นอุปกรณ์ต่างๆสามารถที่จะเป็นได้ทั้งไคลเอ็นต์และเซิร์ฟเวอร์ได้ในเวลาเดียวกัน

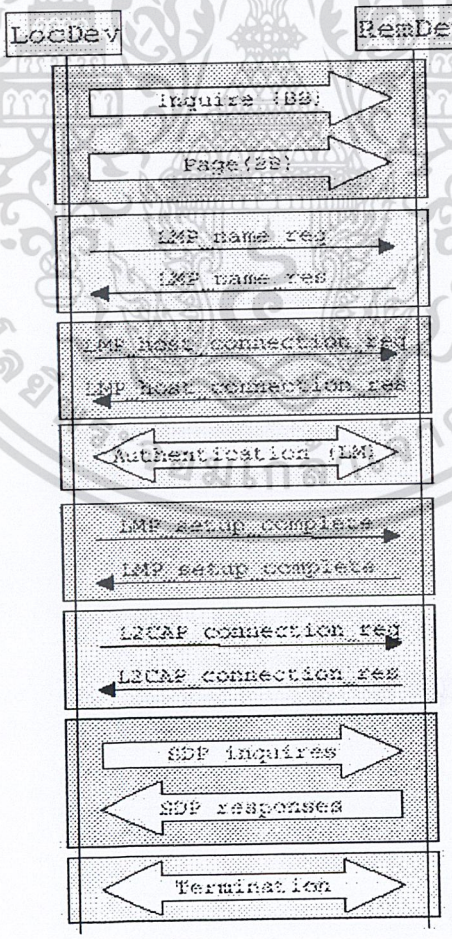
SDP เซิร์ฟเวอร์จะคอยดูแลฐานข้อมูลของเรคอดการบริการ แต่ละเรคอดการบริการจะจัดเตรียมข้อมูลต่างๆที่ไคลเอ็นต์จำเป็นต้องใช้ในการเข้าถึงการบริการนั้นๆ ซึ่งตัวอย่างข้อมูลเช่น URL ที่ใช้ในการเรียกการทำงาน, รายละเอียดของากรบริการและไอคอนของการบริการ ดังนั้นไคลเอ็นต์สามารถเรียกใช้งานบริการต่างๆได้ผ่านทาง URL ที่กำหนด เป็นต้น

ในการใช้ SDP นั้น แชนเนล L2CAP จะต้องถูกสร้างขึ้นมาก่อนระหว่าง SDP ไคลเอ็นต์และเซิร์ฟเวอร์ แชนเนลนี้จะถูกสงวนไว้ใช้สำหรับ SDP เท่านั้น ดังนั้นอุปกรณ์ต่างๆสามารถที่จะเชื่อมต่อมาใช้การบริการบนอุปกรณ์ตัวอื่นได้ หลังจากที่ข้อมูลข่าวสารที่ได้รับมาจากเซิร์ฟเวอร์แล้ว ไคลเอ็นต์จะสร้างช่องทางขึ้นมาใหม่เพื่อใช้การบริการนั้น

การบริการจะมีเลข UUID ที่ใช้ระบุถึง การบริการที่ถูกกำหนดโดยโพรไฟล์บลูทูธจะมีเลข UUID ที่ถูกกำหนดขึ้นมาเป็นมาตรฐาน แต่สำหรับผู้ที่ต้องการสร้างการบริการใหม่ขึ้นมาสามารถกำหนด UUID ของตนเองได้ UUID จะถูกกำหนดขึ้นมาภายใต้วิธีที่รับประกันว่าไม่มีการซ้ำ ดังนั้นจึงไม่จำเป็นต้องมีส่วนกลางในการกำหนดเลข UUID ขึ้นมา

UUID จะถูกส่งไปในข้อความเพื่อถามเซิร์ฟเวอร์ว่ารองรับการบริการที่ระบุมาใน UUID หรือไม่ นอกจากนี้แทนที่จะสอบถามถึงการบริการที่ระบุไปอย่างเดียวนั้น SDP ยังจัดเตรียมกระบวนการในการค้นหาการบริการโดยที่ไม่จำเป็นต้องทราบรายละเอียดมาก่อนด้วย

SDP ไม่ได้กำหนดถึงโปรแกรมประยุกต์ที่ใช้ในการค้นหาการบริการ และไม่ได้กำหนดถึงส่วนเชื่อมต่อกับโปรแกรมประยุกต์ต่างๆ ขึ้นอยู่กับผู้ที่ทำการพัฒนาต่อ ซึ่งอาจนำไปทำงานร่วมกับการค้นหาการบริการอื่นๆที่ได้จัดเตรียม API ไว้ให้ เช่น Salutation หรือ SLP(Service Location Protocol) เป็นต้น



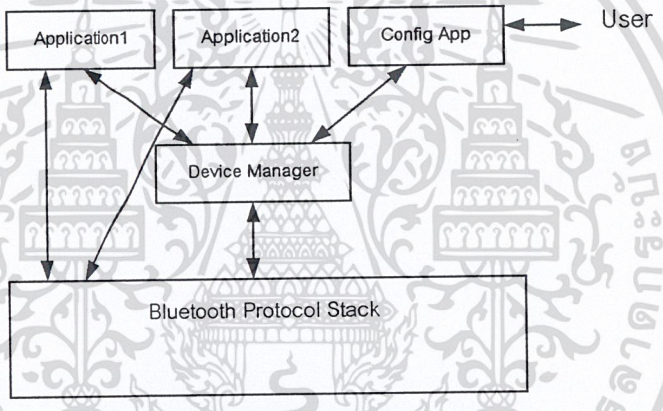
รูปที่ 4-16 ลำดับขั้นตอนในการเชื่อมต่อ SDP

บทที่ 5
การออกแบบและหลักการทำงาน

ในบทนี้เริ่มจากการกำหนดโครงสร้างหลักของระบบ กำหนดขอบเขตของตัวจัดการอุปกรณ์ คัดเลือกโพรโตคอลสแต็กที่เหมาะสมจะนำมาใช้ในโครงงาน จากนั้นจึงทำการศึกษาโพรโตคอลสแต็กที่หามาได้แล้วจึงออกแบบส่วนของซอฟต์แวร์ซึ่งเป็นส่วนที่กำหนดคุณสมบัติและฟังก์ชันการทำงานของตัวจัดการอุปกรณ์ (Device Manager) หลังจากนั้นจึงเป็นส่วนของการอิมพลีเมนต์ตามหลักการที่ได้ออกแบบมาข้างต้นนี้

5.1 โครงสร้างหลักของระบบ

โครงสร้างหลักของระบบแสดงใน รูปที่ 5-1



รูปที่ 5-1 โครงสร้างหลักของระบบ

จากรูปจะเห็นได้ว่าโครงสร้างหลักของระบบมีส่วนประกอบดังนี้

- โพรโตคอลสแต็ก เป็นส่วนโพรโตคอลของตัวโมดูลบูลทูธ ซึ่งทำให้แอพพลิเคชันและตัวจัดการอุปกรณ์ที่สร้างขึ้นสามารถติดต่อกับโมดูลได้
- Device Manager (ตัวจัดการอุปกรณ์) เป็นส่วนที่สร้างขึ้นมาเพื่อจัดการเกี่ยวกับอุปกรณ์และการบริการที่อยู่ในเครือข่าย
- Config App เป็นส่วนที่ติดต่อกับผู้ใช้ (User interface) เพื่อให้ผู้ใช้สามารถใช้งานตัวจัดการอุปกรณ์ได้
- Application 1 และ 2 เป็นแอพพลิเคชันที่สร้างขึ้นมา เพื่อทดสอบการใช้การบริการที่ผู้ใช้ทำการเรียกใช้ ซึ่งในที่นี้ได้ทำการสร้างแอพพลิเคชันมา 2 อย่างคือ Chat และ File transfer

5.2 ขอบเขตของตัวจัดการอุปกรณ์

ในโมเดลของ OSI นั้นได้พูดถึงเรื่องของการจัดการ (OSI Management) ไว้ 5 หัวข้อ ดังนี้

- การจัดการในเรื่องของการปรับแต่งค่าและการจัดการชื่อ (Configuration and Name Management)
- การจัดการในเรื่องของประสิทธิภาพในการทำงาน (Performance Management)
- การจัดการในเรื่องของความปลอดภัย (Security Management)
- การจัดการในเรื่องของความผิดพลาด (Fault Management)
- การจัดการในเรื่องของการเก็บเงิน (Accounting Management)

ในมาตรฐานบลูทูธนั้นไม่ได้มีการกำหนดมาตรฐานของตัวจัดการอุปกรณ์ที่ใช้ในบลูทูธว่าจะต้องมีรายละเอียดในการทำงานอย่างไร เพราะตัวจัดการสำหรับแต่ละอุปกรณ์นั้นจะมีรายละเอียดที่ต่างกัน เนื่องจากความหลากหลายของอุปกรณ์ที่ใช้งานในบลูทูธ เช่น หูฟังไร้สายนั้นจะจัดการการเชื่อมต่อเพียงการเชื่อมต่อเดียวเท่านั้น แต่สำหรับอุปกรณ์ที่เป็นจุดเข้าถึงระบบแลนนั้นจะต้องรองรับการเชื่อมต่อจากอุปกรณ์ต่างๆหลายอุปกรณ์ด้วยกัน ดังนั้นจึงไม่มีมาตรฐานกลางสำหรับตัวจัดการอุปกรณ์ขึ้นมา แต่ได้มีการพูดถึงประเด็นที่ตัวจัดการอุปกรณ์ของบลูทูธจะสามารถจัดการในเรื่องต่างๆดังต่อไปนี้ คือ

- การจัดการการเชื่อมต่อ
- การจัดการในเรื่องของความปลอดภัย
- การจัดการในเรื่องของการอินทิเกรต โปรแกรมประยุกต์
- การจัดการในเรื่องของการเก็บเงิน
- การจัดการในเรื่องของคุณภาพของการบริการ
- การจัดการในเรื่องของการออกแบบส่วนติดต่อกับผู้ใช้

ซึ่งตัวจัดการอุปกรณ์ที่พัฒนาขึ้นในโครงงานนี้นั้น จะดูแลในเรื่องของการจัดการการเชื่อมต่อในขอบเขตของการค้นหาอุปกรณ์และการค้นหาการบริการที่มีอยู่ในเครือข่าย เพื่อแสดงความสามารถในการทำงานเป็นเครือข่ายแบบแอดฮอคผ่านบลูทูธ โดยอุปกรณ์จะสามารถทำการค้นหาอุปกรณ์อื่นๆที่อยู่ในระยะเครือข่าย และสามารถค้นหาการบริการต่างๆที่อุปกรณ์อื่นๆมีให้ใช้ได้

5.3 การค้นหาโพรโตคอลสแต็ก

ในส่วนของการค้นหาโพรโตคอลสแต็กที่เหมาะสมจะนำมาใช้ในโครงงานนั้น ทางกลุ่มได้ค้นหาโพรโตคอลสแต็กจากแหล่งต่างๆในอินเทอร์เน็ต เพื่อจะหาโพรโตคอลสแต็กที่เหมาะสมจะนำมาใช้ในโครงงาน ซึ่งต้องมีคุณสมบัติขั้นพื้นฐานดังต่อไปนี้

- เป็นของฟรี
- มีเลเยอร์พื้นฐานที่สำคัญในการใช้งานบลทูธมาให้แล้ว เช่น HCI, L2CAP, RFCOMM เป็นต้น

จากการค้นหาโพรโตคอลสแต็กต่างๆที่มีอยู่ในอินเทอร์เน็ตนั้น ทางกลุ่มได้พบอยู่เป็นจำนวนมาก แต่โพรโตคอลสแต็กที่พบส่วนใหญ่นั้นไม่ใช่ของฟรี หากจะนำมาใช้งานต้องเสียเงินและโพรโตคอลสแต็กในกลุ่มนี้นั้นจะมีเลเยอร์ต่างๆพร้อมทั้งมีฟังก์ชันการใช้งานที่เพรียบพร้อมมาให้อยู่แล้ว ซึ่งเป็นโพรโตคอลสแต็กที่ไม่ตรงกับความต้องการของกลุ่ม ดังนั้นจึงทำการคัดเลือกหาโพรโตคอลสแต็กที่ตรงกับคุณสมบัติพื้นฐานข้างต้นที่กล่าวมาแล้วนั้นได้ 4 ตัวที่น่าสนใจด้วยกัน โดยที่โพรโตคอลสแต็กของบลูทูธทั้งสี่ตัวที่หามานี้เป็นของฟรี สามารถดาวน์โหลดมาใช้งานได้ทันที และมีเลเยอร์พื้นฐานที่จำเป็นในการใช้งานมาให้แล้ว (ในเลเยอร์บางตัวนั้นอาจยังไม่สมบูรณ์) โดยมีรายชื่อดังนี้ คือ

- OpenBT
- BlueZ
- BlueDrekar
- Cstack

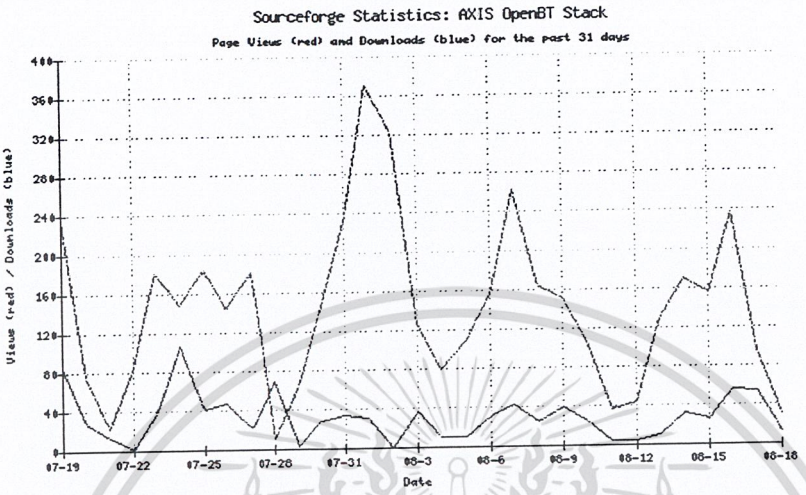
5.3.1 OpenBT

เดิมทีถูกพัฒนาโดย AXIS และต่อมาเป็นส่วนหนึ่งของโปรเจค SourceForge ซึ่งเป็นโปรเจคที่ทำการพัฒนาซอฟต์แวร์แบบเปิด (Open Source) โดยกลุ่มคนที่สนใจในซอฟต์แวร์ตัวใดที่อยู่ในโครงการนี้สามารถทำการดาวน์โหลดโปรแกรมไปใช้งานได้ทันที โดยจะมีส่วนของโค้ดโปรแกรมมาให้ด้วย โดยที่ไม่เสียค่าใช้จ่ายใดๆทั้งสิ้น ซึ่งในกรณีที่พบข้อผิดพลาดของโปรแกรมสามารถแจ้งกลับมายังโครงการเพื่อที่ทางโครงการจะได้นำไปแก้ไขปรับปรุงต่อได้ จะเห็นได้ว่าในการพัฒนาซอฟต์แวร์แบบเปิดนี้ ทุกคนสามารถมีส่วนร่วมในการพัฒนาได้ ดังนั้นโปรแกรมส่วนใหญ่ที่พัฒนาอยู่ในโครงการนี้ จะเป็นโปรแกรมที่นำมาใช้งานกับระบบปฏิบัติการ UNIX หรือ LINUX เป็นส่วนใหญ่ เนื่องจากระบบปฏิบัติการในกลุ่มนี้จะเป็นการพัฒนาซอฟต์แวร์แบบเปิด โดยที่รู้จักกันดีในชื่อของ GNU (General Public License)

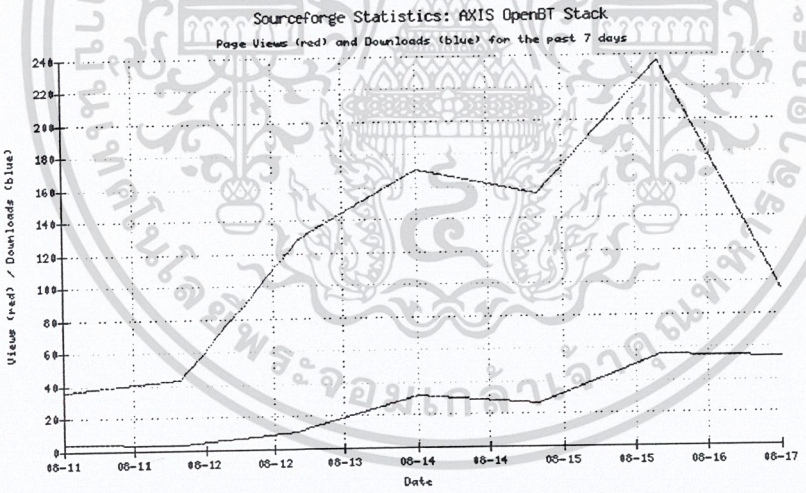
ในส่วนของ OpenBT โพรโตคอลสแต็กถูกเขียนขึ้นมาด้วยภาษาซี รองรับการทำงานกับ Glibc ซึ่งเป็นคอมไพเลอร์ภาษาซีบนระบบปฏิบัติการ LINUX ที่ทำงานบนเคอร์เนลเวอร์ชัน 2.4.x ขึ้นไป รองรับการทำงานของโพรโตคอลสแต็กบลูทูธในเลเยอร์ต่างๆ ดังนี้

- HCI
- L2CAP
- RFCOMM
- SDP

ซึ่งในรายละเอียดของแต่ละเลเยอร์นั้นยังไม่สมบูรณ์ รูปที่ 5-2 และ 5-3 เป็นการแสดงถึงสถิติจำนวนการเข้าชมเว็บเพจของโครงการและจำนวนผู้ใช้ที่ดาวน์โหลดโพรโตคอลสแต็กไปทดลองใช้งานในช่วงเวลาย้อนหลังไป 31 วัน และ 7 วัน ตามลำดับ (อ้างอิง ณ วันที่) 18 สิงหาคม 2544



รูปที่ 5-2 แสดงสถิติจำนวนการเข้าชมเว็บเพจของโครงการและจำนวนผู้ใช้ที่ดาวน์โหลด OpenBT ไปทดลองใช้งานในช่วงเวลาย้อนหลังไป 31 วัน



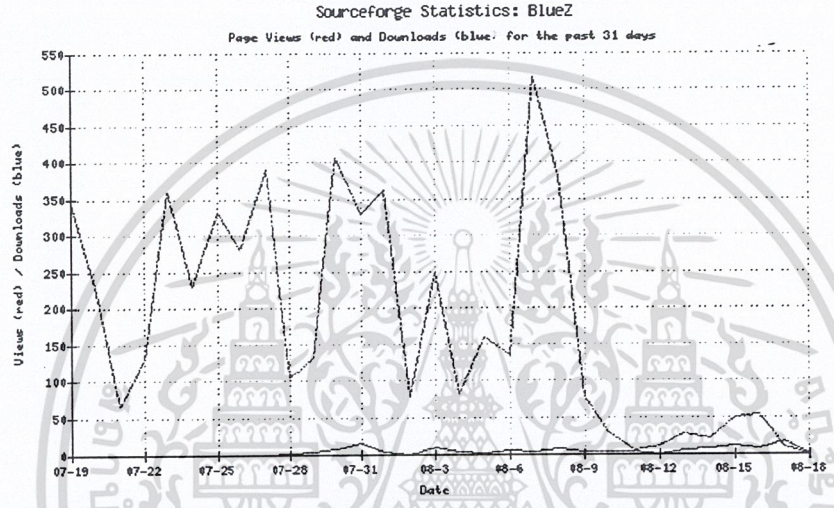
รูปที่ 5-3 แสดงสถิติจำนวนการเข้าชมเว็บเพจของโครงการและจำนวนผู้ใช้ที่ดาวน์โหลด OpenBT ไปทดลองใช้งานในช่วงเวลาย้อนหลังไป 7 วัน

5.3.2 BlueZ

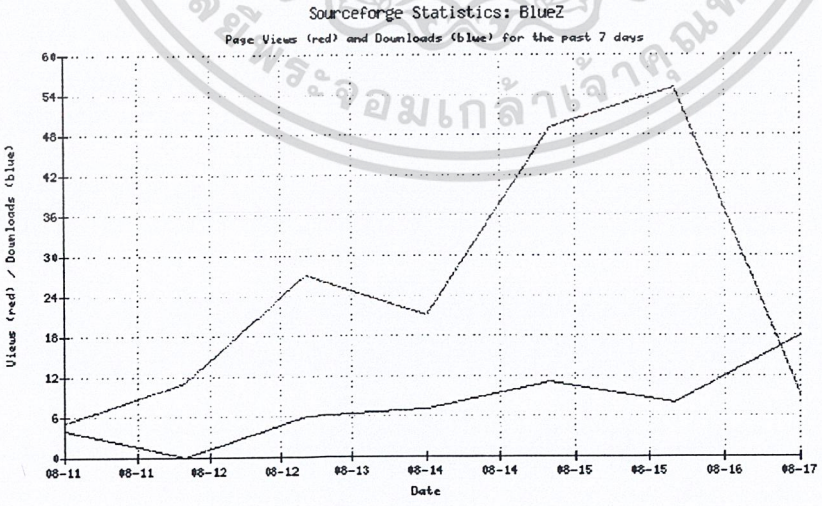
เป็นอีกโครงการหนึ่งในการพัฒนาโพรโตคอลสแต็กของบลูทูธที่อยู่ในโปรเจค SourceForge โพรโตคอลสแต็กถูกเขียนขึ้นมาด้วยภาษาซี รองรับการทำงานกับ Glibc ซึ่งเป็นคอมไพเลอร์ภาษาซีบนระบบปฏิบัติการ LINUX ที่ทำงานบนเคอร์เนลเวอร์ชัน 2.4.x ขึ้นไป รองรับการทำงานของโพรโตคอลสแต็กบลูทูธในส่วนต่างๆ ดังนี้

- HCI Core
- HCI UART, USB และ HCI Emulation Drivers
- L2CAP protocol Module
- ยูทิลิตีในการปรับแต่งและการทดสอบ

ซึ่งในรายละเอียดของแต่ละเลเยอร์นั้นยังไม่สมบูรณ์ รูปที่ 5-4 และ 5-5 เป็นการแสดงถึงสถิติจำนวนการเข้าชมเว็บเพจของโครงการและจำนวนผู้ใช้ที่ดาวน์โหลดโพรโตคอลสแต็กไปทดลองใช้งานในช่วงเวลาย้อนหลังไป 31 วัน และ 7 วัน ตามลำดับ



รูปที่ 5-4 แสดงสถิติจำนวนการเข้าชมเว็บเพจของโครงการและจำนวนผู้ใช้ที่ดาวน์โหลด BlueZ ไปทดลองใช้งานในช่วงเวลาย้อนหลังไป 31 วัน



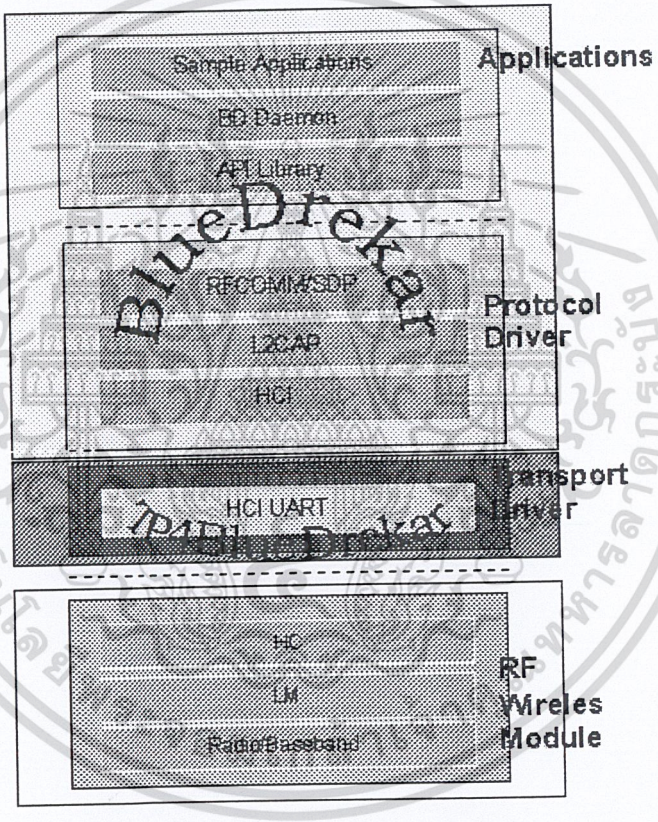
รูปที่ 5-5 แสดงสถิติจำนวนการเข้าชมเว็บเพจของโครงการและจำนวนผู้ใช้ที่ดาวน์โหลด BlueZ ไปทดลองใช้งานในช่วงเวลาย้อนหลังไป 7 วัน

5.3.3 Bluedrekar

เป็นโพรโตคอลสแต็กของบลูทูธที่ถูกพัฒนาโดยบริษัท IBM ซึ่งมีความต้องการที่จะพัฒนาให้เป็น โพรโตคอลที่เป็นมิดเดิลแวร์ ภายใต้มาตรฐานของบลูทูธ รองรับการทำงานบนระบบปฏิบัติการ LINUX ที่ทำงานอยู่บนเคอร์เนลเวอร์ชัน 2.2.14 และ 2.4.2 รองรับการทำงานของโพรโตคอลสแต็กบลูทูธในส่วนต่างๆ ดังนี้

- โมดูลจากในการติดต่อ HCI ไปยัง RFCOMM
- เลเยอร์ SDP

ซึ่งโค้ดการทำงานของโปรแกรมจะเผยแพร่ให้เฉพาะในกลุ่มผู้สนับสนุนและกลุ่มผู้พัฒนาบลูทูธภายใต้เงื่อนไขลิขสิทธิ์ของ IBM เท่านั้น



รูปที่ 5-6 โพรโตคอลสแต็กของ BlueDrekar

5.3.4 Cstack

เป็นโพรโตคอลสแต็กที่ถูกพัฒนาโดยภาษาซี แล้วนำมาห่อหุ้มข้อมูลภายในคอมโพเนนของ Active-X ซึ่งทำให้สามารถเข้าไปใช้งานได้ง่าย โดยสภาพแวดล้อมในการสร้างโปรแกรมประยุกต์ขึ้นมาใช้งานนั้นเป็นสภาพแวดล้อมภายใต้ระบบปฏิบัติการของวินโดวส์ (Windows) โพรโตคอลสแต็กเป็นของฟรีแต่ไม่ได้เป็นมาตรฐานเปิด

จากการที่ทางกลุ่มได้คัดเลือกโพรโตคอลสแต็กของบลูทูธที่จะนำใช้งานในโครงงานนี้จนได้มาสี่ตัวแล้วนั้น ทางกลุ่มได้ตัดสินใจเลือกใช้โพรโตคอลสแต็กของ OpenBT เนื่องจากมีข้อได้เปรียบที่ต่างจากอื่นๆ คือ ใช้ภาษาซีในการพัฒนา ข้อดีคือสามารถนำไปพอร์ตไปเป็นอุปกรณ์ในระบบฝังตัวได้ง่าย ซึ่งเหมาะกับหลักการของบลูทูธ เนื่องจากอุปกรณ์ที่ใช้งานบลูทูธนั้น ส่วนใหญ่เป็นอุปกรณ์ที่สามารถพกพาได้ ดังนั้นจึงมีการพัฒนาระบบฝังตัวต่างๆลงไปในอุปกรณ์ประเภทนี้, เป็นโพรโตคอลที่มีการพัฒนาอย่างต่อเนื่อง ทำให้มีการแก้ไขปรับปรุงข้อผิดพลาดที่เกิดขึ้นกับโปรแกรมในเวอร์ชันก่อนๆ และเป็นโพรโตคอลสแต็กที่มีผู้ให้ความสนใจเป็นจำนวนมากเมื่อเทียบกับ BlueZ (สังเกตได้จากสถิติจำนวนการเข้าชมเว็บเพจของโครงการและจำนวนผู้ใช้ที่ดาวน์โหลด OpenBT) ในกรณีของ BlueDrekar นั้นผู้ที่จะสามารถนำโค้ดไปพัฒนาต่อได้นั้น ต้องเป็นกลุ่มกลุ่มผู้สนับสนุนและกลุ่มผู้พัฒนาบลูทูธภายใต้เงื่อนไขลิขสิทธิ์ของ IBM เท่านั้น และสุดท้ายในกรณีของ Cstack นั้น โพรโตคอลที่มีให้นำไปใช้งานนั้นเป็นของฟรี แต่ไม่ได้เป็นมาตรฐานเปิดดังนั้นจึงไม่สามารถนำโค้ดของโปรแกรมไปทำการศึกษาและพัฒนาต่อได้

5.4 การออกแบบส่วนซอฟต์แวร์

ในการพัฒนาตัวจัดการอุปกรณ์สำหรับการทำงานในระบบเครือข่ายไร้สายแบบแอดฮอคผ่านบลูทูธได้มีการออกแบบซอฟต์แวร์ในส่วนของ Device manager , config application และแอพพลิเคชันโดยใช้ UML (Unified Modeling Language) ซึ่งเป็นการวิเคราะห์และออกแบบระบบโดยการสร้างโมเดลขึ้นมาเพื่อทำการจำลองระบบให้อยู่ในรูปของออบเจ็กต์และความสัมพันธ์ต่างๆระหว่างออบเจ็กต์

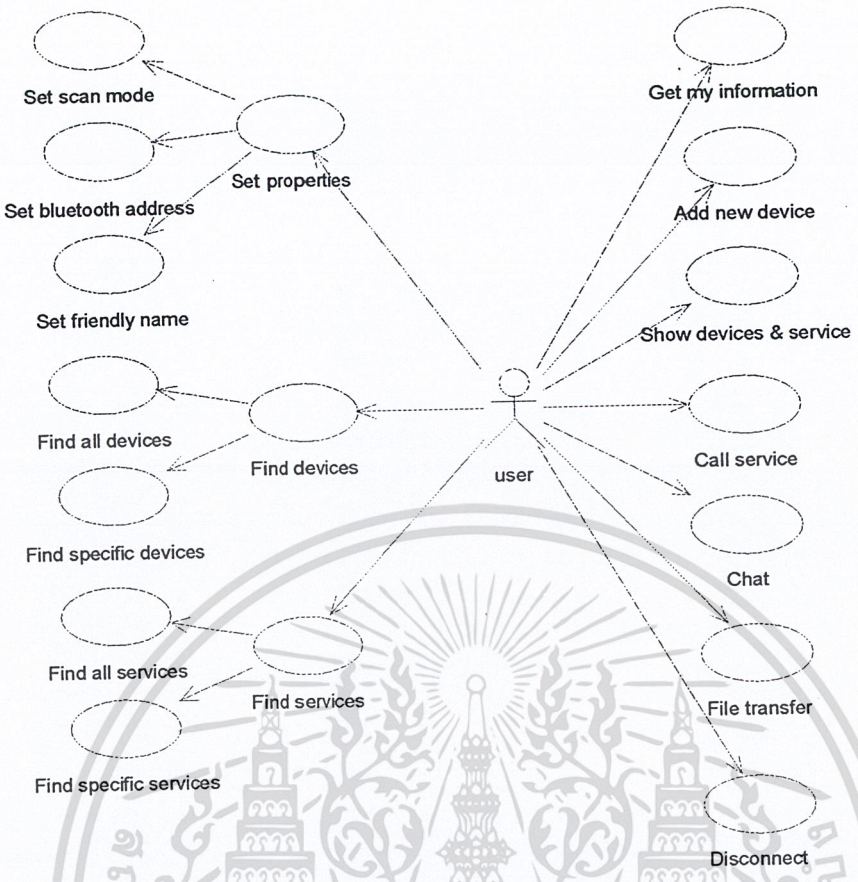
ในการวิเคราะห์และออกแบบระบบโดยใช้ UML จะประกอบไปด้วยไดอะแกรมต่างๆดังนี้

5.4.1 Use Case Diagram

Use Case Diagram เป็นไดอะแกรมที่แสดงถึงฟังก์ชันการใช้งานหลักๆ ที่ผู้ใช้สามารถใช้งานได้ แสดงในรูปที่ 5-7

จากรูปฟังก์ชันการทำงานหลักที่สำคัญประกอบไปด้วยฟังก์ชันต่างๆ ดังนี้

1. Set properties เป็นฟังก์ชันที่ให้ผู้ใช้ทำการกำหนดค่าคุณสมบัติต่างๆของอุปกรณ์ โดยภายในฟังก์ชันนี้ยังประกอบด้วยฟังก์ชันย่อยให้ทำการกำหนดค่าคุณสมบัติ ดังนี้
   - Set scan mode เป็นฟังก์ชันที่ให้ผู้ใช้ทำการกำหนดค่าสแกนโหมดว่าต้องการให้อุปกรณ์ตัวอื่นสามารถค้นหาหรือเชื่อมต่อกับอุปกรณ์ของเราหรือไม่
   - Set bluetooth address เป็นฟังก์ชันที่ให้ผู้ใช้ทำการกำหนดค่าแอดเดรสใหม่ให้กับอุปกรณ์
   - Set friendly name เป็นฟังก์ชันที่ให้ผู้ใช้ทำการกำหนดค่าเฟรนรี่เนมใหม่ให้กับอุปกรณ์
2. Find devices เป็นฟังก์ชันที่ให้ผู้ใช้ทำการค้นหาอุปกรณ์ที่อยู่ในระยะเครือข่ายและทำการเชื่อมต่อกับอุปกรณ์เหล่านั้น โดยภายในฟังก์ชันนี้ยังประกอบด้วยฟังก์ชันย่อยซึ่งมี 2 รูปแบบ คือ
   - Find all devices เป็นฟังก์ชันที่ให้ผู้ใช้ทำการค้นหาอุปกรณ์ที่อยู่ในระยะเครือข่ายทั้งหมด
   - Find specific devices เป็นฟังก์ชันที่ให้ผู้ใช้ทำการค้นหาอุปกรณ์ที่อยู่ในระยะเครือข่าย โดยระบุเฟรนรี่เนมหรือตัวอักษร



รูปที่ 5-7 Use Case Diagram ของระบบ

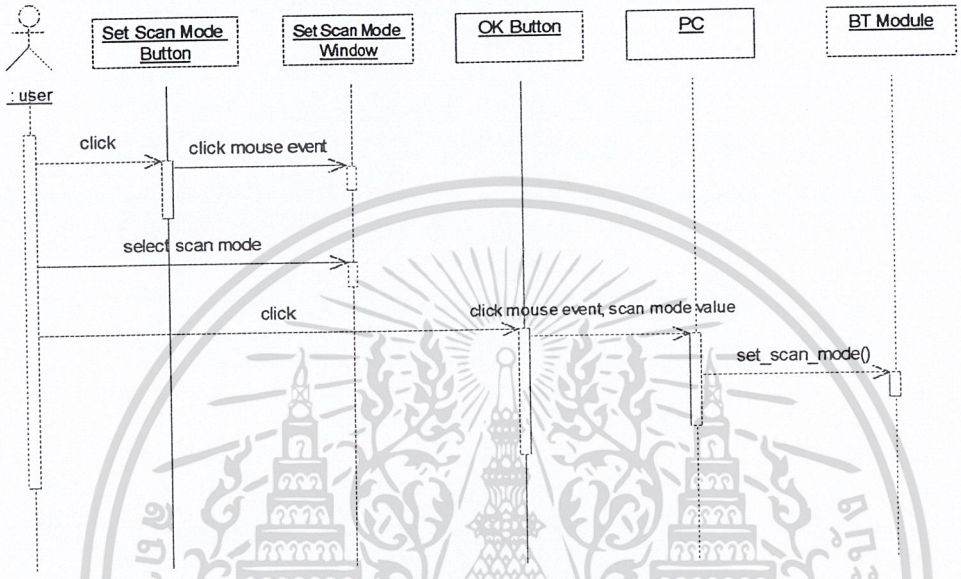
3. Find services เป็นฟังก์ชันที่ให้ผู้ใช้ทำการค้นหาการบริการของอุปกรณ์ต่างๆ ที่อยู่ในเครือข่าย โดยภายในฟังก์ชันนี้ยังประกอบด้วยฟังก์ชันย่อยซึ่งเป็นรูปแบบการค้นหาการบริการ 2 รูปแบบ คือ
   - Find all services เป็นฟังก์ชันที่ให้ผู้ใช้ทำการค้นหาการบริการของอุปกรณ์ที่เชื่อมต่ออยู่ทั้งหมด
   - Find specific services เป็นฟังก์ชันที่ให้ผู้ใช้ทำการค้นหาการบริการของอุปกรณ์ที่อยู่ในเครือข่าย โดยระบุชื่อการบริการหรือตัวอักษร
4. Get my information เป็นฟังก์ชันที่ให้ผู้ใช้ดูรายละเอียดของอุปกรณ์ ได้แก่ แอดเดรส เฟรนรี่เนมและค่าสแกนโหมด
5. Add new device เป็นฟังก์ชันที่ให้ผู้ใช้ทำการเพิ่มอุปกรณ์ตัวใหม่ลงไปในรายการ ซึ่งเก็บรายละเอียดอุปกรณ์และการบริการของอุปกรณ์ตัวนั้น
6. Show devices & services เป็นฟังก์ชันที่ทำการแสดงรายชื่ออุปกรณ์และการบริการให้แก่ผู้ใช้
7. Call service เป็นฟังก์ชันที่ให้ผู้ใช้ทำการขอใช้การบริการที่ต้องการกับอุปกรณ์ตัวอื่น
8. Chat เป็นแอพพลิเคชันการคุยระหว่างอุปกรณ์ 2 ตัว ซึ่งอุปกรณ์ทั้ง 2 ตัวต้องมีแอพพลิเคชันนี้
9. File transfer เป็นแอพพลิเคชันจำลองการโอนถ่ายข้อมูลโดยขอใช้บริการนี้จากอุปกรณ์ตัวอื่น นั่นคืออุปกรณ์ของเราไม่มีโปรแกรมการโอนถ่ายข้อมูลจึงต้องทำการขอให้อุปกรณ์ที่มีโปรแกรมนี้ทำให้โดยผู้ใช้จะทำการส่งรายละเอียดต่างๆไปให้ ได้แก่ โฮส ชื่อผู้ใช้ พาสเวิร์ดและชื่อไฟล์ที่ต้องการ
10. Disconnect เป็นฟังก์ชันที่ทำการยกเลิกการติดต่อกับอุปกรณ์ทั้งหมดและออกจากโปรแกรม

5.4.2 Scenario

Scenario เป็นไดอะแกรมที่ทำการจำลองสถานะการณ์การทำงานต่างๆ ของระบบ เพื่อแสดงให้เห็นถึงการทำงานที่เป็นลำดับขั้นตอนและการทำงานร่วมกันของออบเจ็กต์ต่างๆ ในระบบ

5.4.2.1 Sequence Diagram แสดง Set scan mode

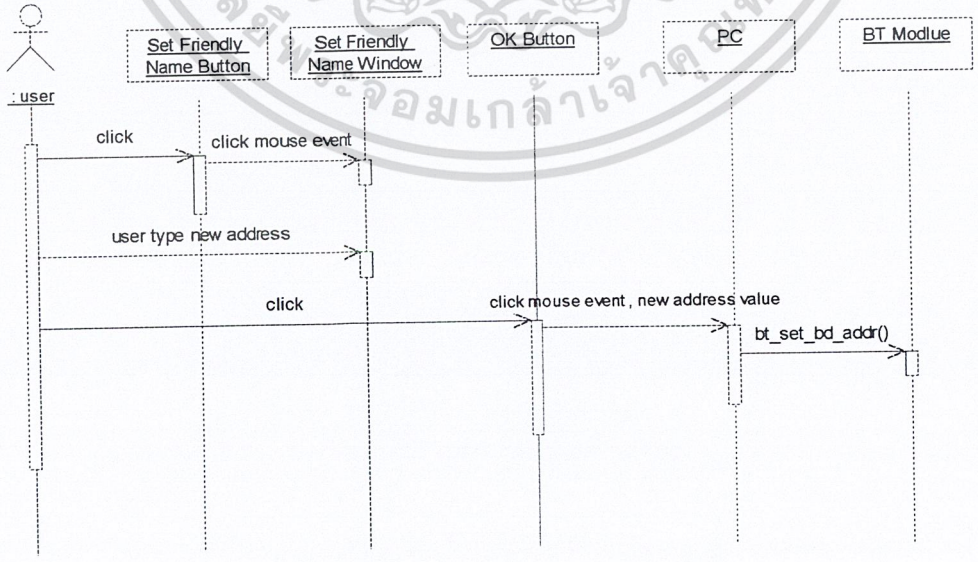
เป็นการจำลองสถานการณ์การกำหนดคุณสมบัติการสแกนโหมดของอุปกรณ์ แสดงในรูปที่ 5-8



รูปที่ 5-8 Sequence Diagram แสดง Set scan mode

5.4.2.2 Sequence Diagram แสดง Set bluetooth address

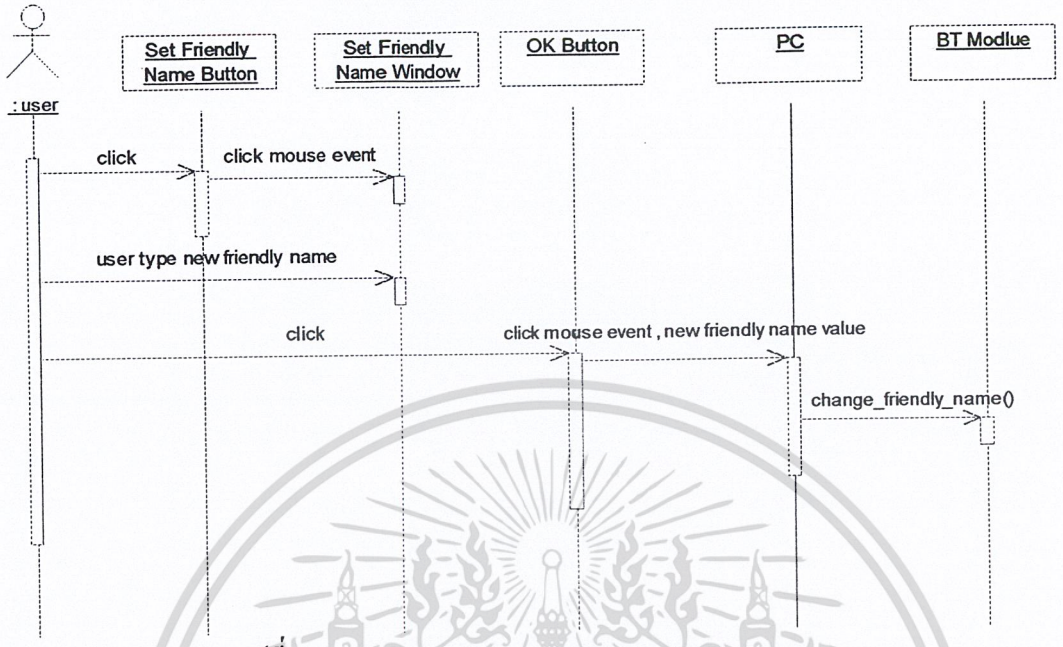
เป็นการจำลองสถานการณ์การกำหนดค่าแอดเดรสใหม่ให้กับอุปกรณ์ แสดงในรูปที่ 5-9



รูปที่ 5-9 Sequence Diagram แสดง Set bluetooth address

5.4.2.3 Sequence Diagram แสดง Set friendly name

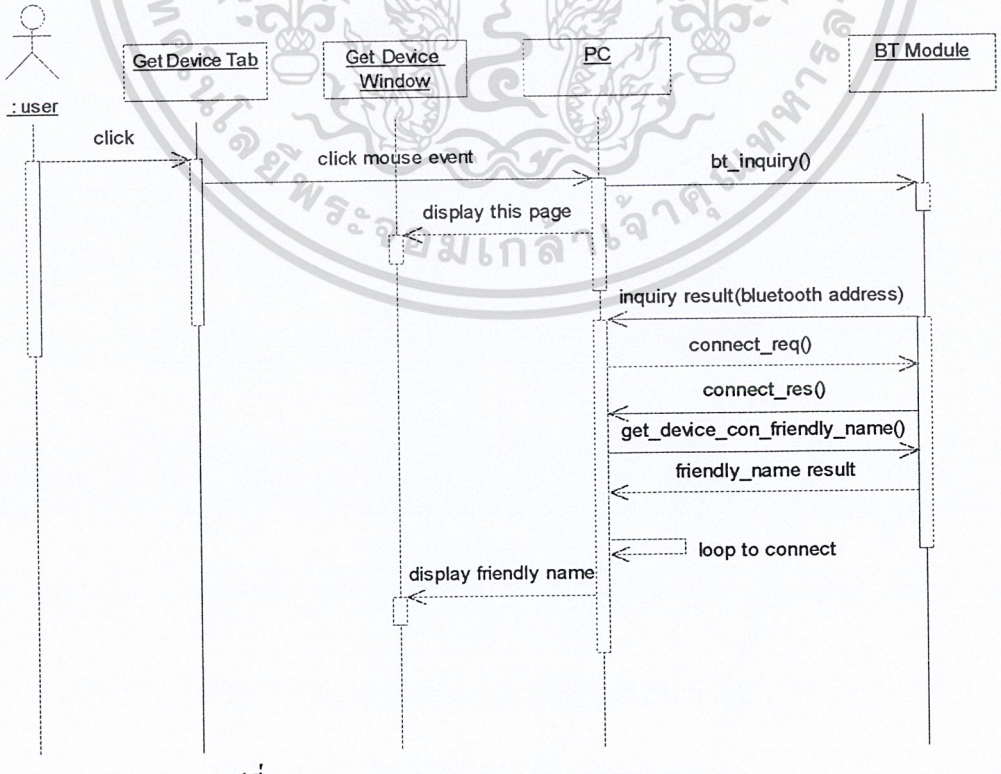
เป็นการจำลองสถานการณ์การกำหนดเฟรนรี่เนมใหม่ให้กับอุปกรณ์ แสดงในรูปที่ 5-10



รูปที่ 5-10 Sequence Diagram แสดง Set friendly name

5.4.2.4 Sequence Diagram แสดง File all devices

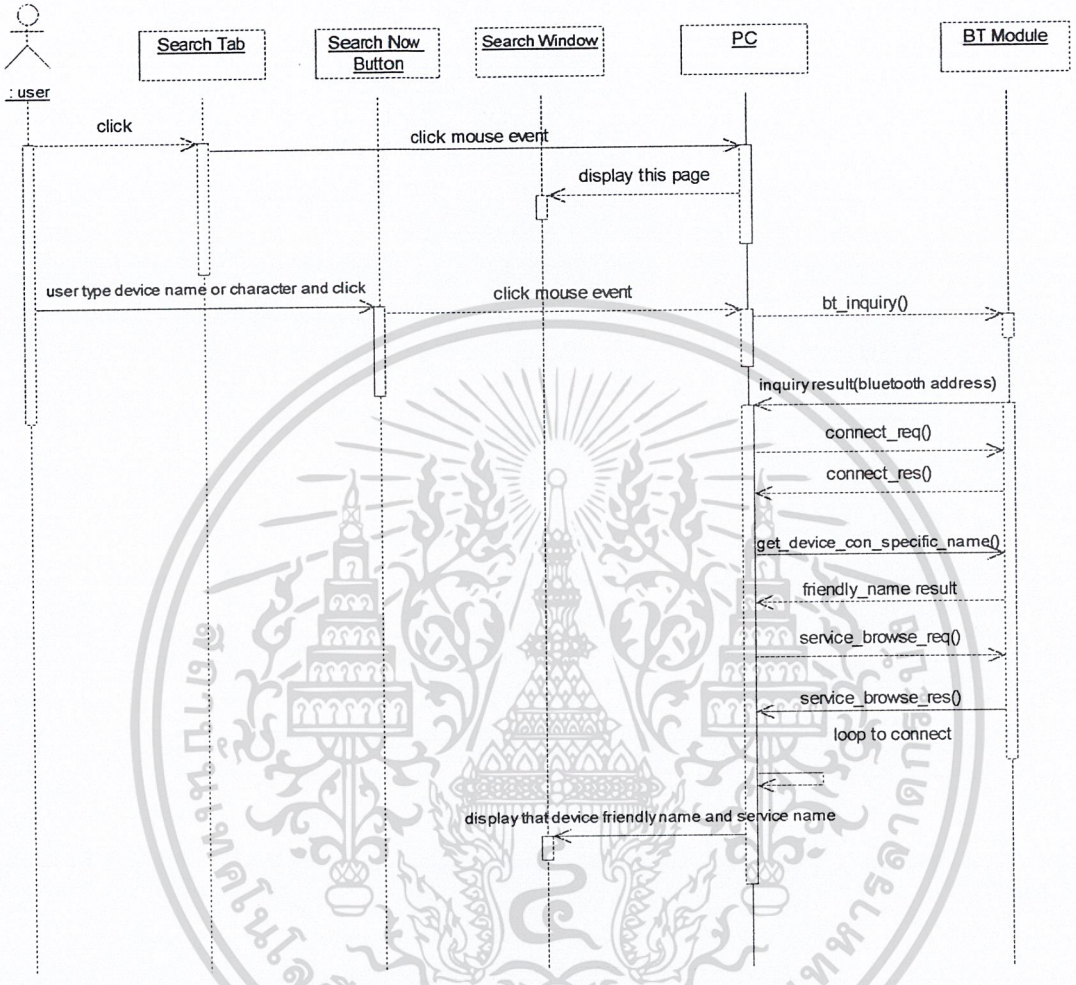
เป็นการจำลองสถานการณ์การค้นหาอุปกรณ์ที่อยู่ในเครือข่ายทั้งหมด แล้วทำการเชื่อมต่อกับอุปกรณ์ตัวนั้น แสดงในรูปที่ 5-11



รูปที่ 5-11 Sequence Diagram แสดง File all devices

5.4.2.5 Sequence Diagram แสดง File specific devices

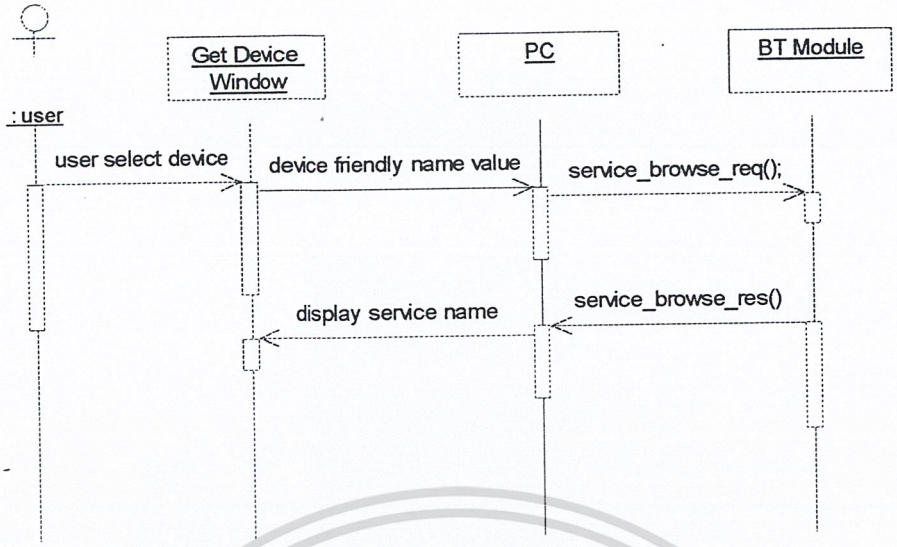
เป็นการจำลองสถานการณ์การค้นหาอุปกรณ์ที่อยู่ในเครือข่ายโดยระบุเฟรนรี่เนมหรือตัวอักษร แล้วทำการเชื่อมต่อกับอุปกรณ์ตัวนั้น แสดงในรูปที่ 5-12



รูปที่ 5-12 Sequence Diagram แสดง File specific devices

5.4.2.6 Sequence Diagram แสดง Find all services

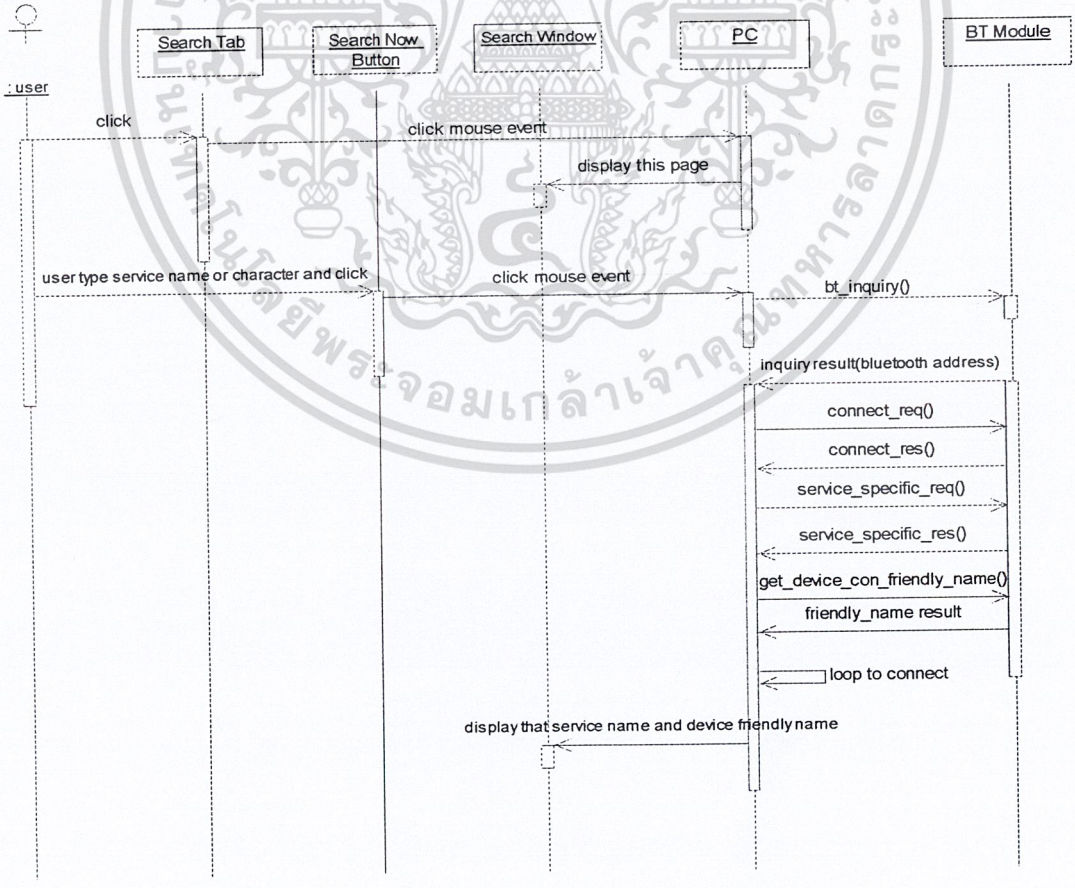
เป็นการจำลองสถานการณ์การค้นหาการบริการของอุปกรณ์ที่เชื่อมต่ออยู่ทั้งหมด ในรูปที่ 5-13



รูปที่ 5-13 Sequence Diagram แสดง Find all services

5.4.2.7 Sequence Diagram แสดง Find specific services

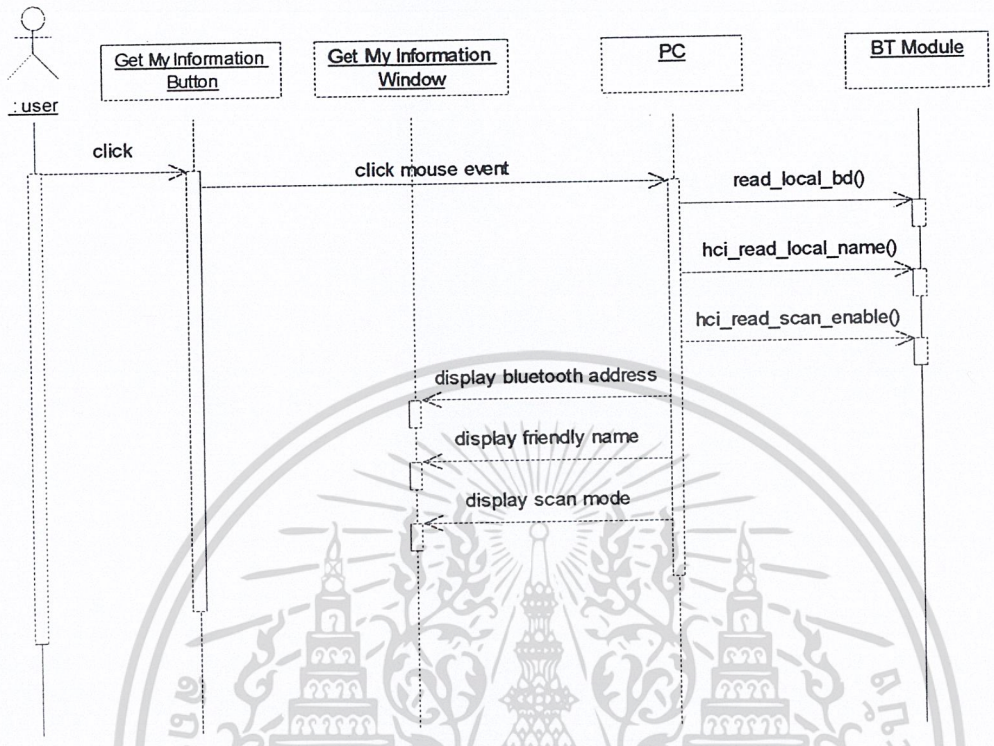
เป็นการจำลองสถานการณ์การค้นหาการบริการที่มีอยู่ในเครือข่าย โดยระบุชื่อการบริการหรือตัวอักษร แสดงในรูป 5-14



รูปที่ 5-14 Sequence Diagram แสดง Find specific services

5.4.2.8 Sequence Diagram แสดง Get my information

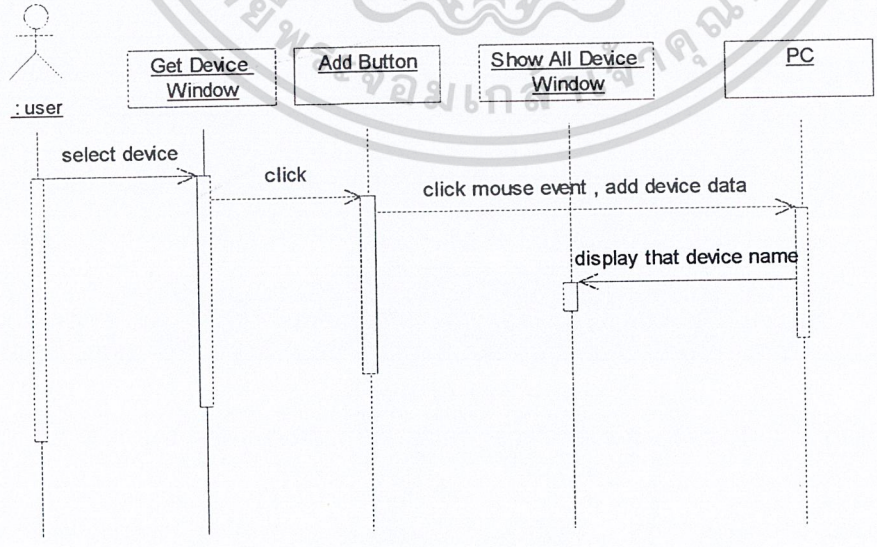
เป็นการจำลองสถานการณ์การให้ผู้ใช้ดูรายละเอียดของอุปกรณ์ ได้แก่ แอดเดรส เฟรนรี่เนมและค่าสแกนโหมด แสดงในรูปที่ 5-15



รูปที่ 5-15 Sequence Diagram แสดง Get my information

5.4.2.9 Sequence Diagram แสดง Add new device

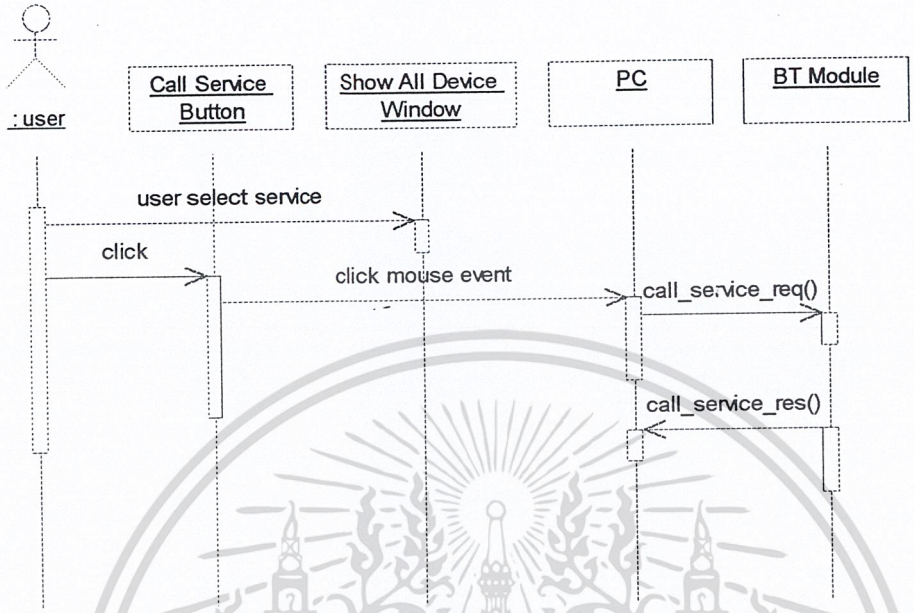
เป็นการจำลองสถานการณ์ให้ผู้ใช้ทำการเพิ่มอุปกรณ์ตัวใหม่ลงไปในรายการ ซึ่งเก็บรายละเอียดอุปกรณ์และการบริการของอุปกรณ์ตัวนั้น แสดงในรูป 5-16



รูปที่ 5-16 Sequence Diagram แสดง Add new device

5.4.2.10 Sequence Diagram แสดง Call service

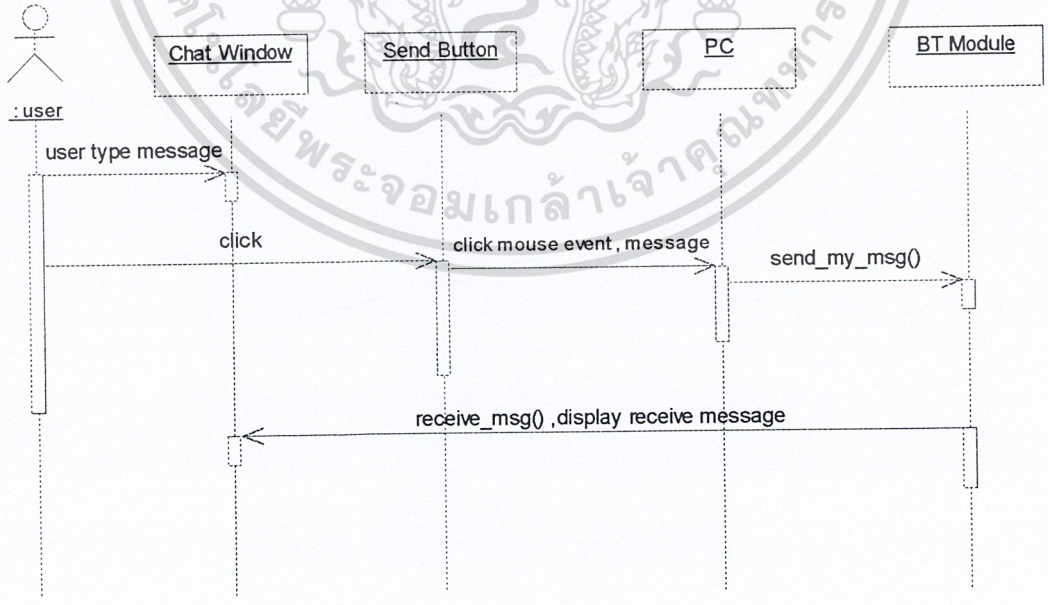
เป็นการจำลองสถานการณ์ให้ผู้ใช้ทำการขอใช้การบริการที่ต้องการกับอุปกรณ์ตัวอื่น แสดงในรูปที่ 5-17



รูปที่ 5-17 Sequence Diagram แสดง Call service

5.4.2.11 Sequence Diagram แสดง Chat

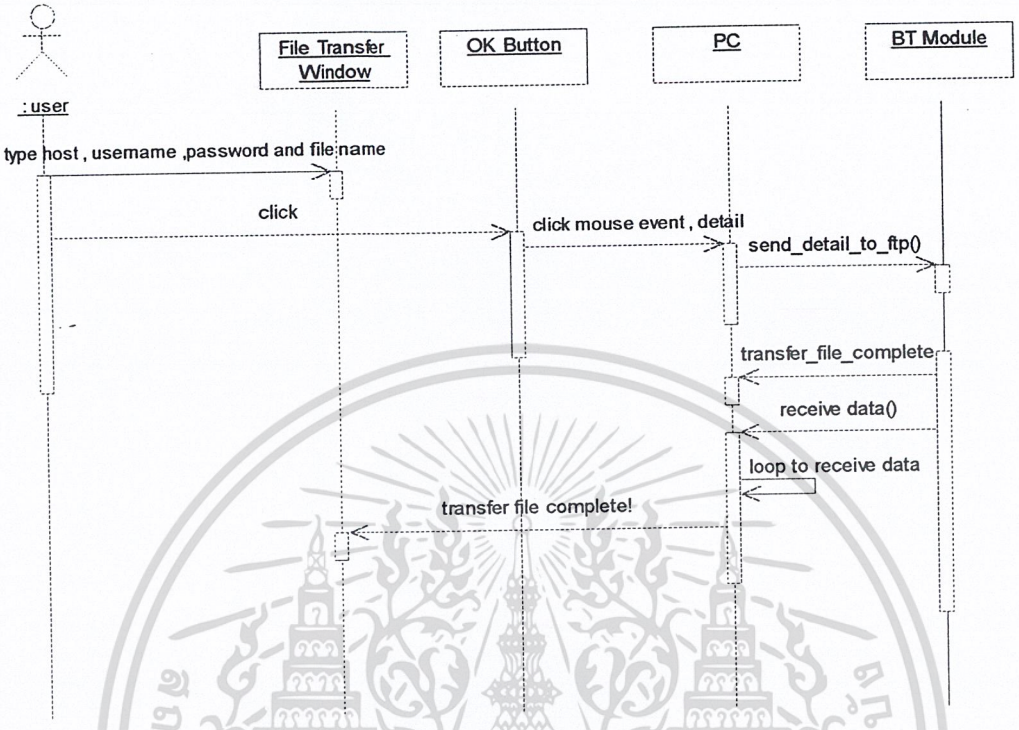
เป็นการจำลองสถานการณ์การใช้แอพพลิเคชันการคุยกันระหว่างอุปกรณ์ 2 ตัว แสดงในรูป 5-18



รูปที่ 5-18 Sequence Diagram แสดง Chat

5.4.2.12 Sequence Diagram แสดง File transfer

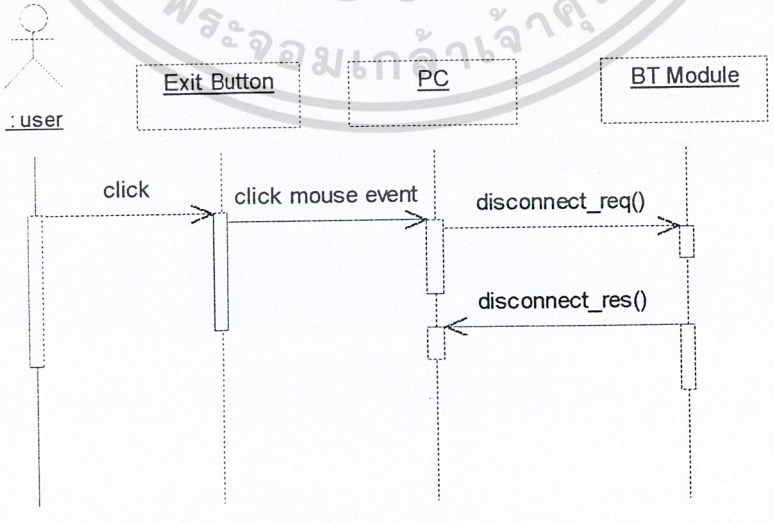
เป็นการจำลองสถานการณ์การขอใช้บริการการโอนถ่ายข้อมูลจากอุปกรณ์ตัวอื่น เนื่องจากอุปกรณ์ของเราไม่มีการบริการนี้ แสดงในรูปที่ 5-19



รูปที่ 5-19 Sequence Diagram แสดง File transfer

5.4.2.13 Sequence Diagram แสดง Disconnect

เป็นการจำลองสถานการณ์การยกเลิกการติดต่อกับอุปกรณ์ทั้งหมดและออกจากโปรแกรม แสดงในรูปที่ 5-20



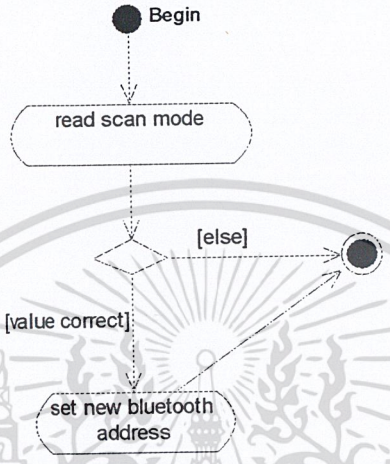
รูปที่ 5-20 Sequence Diagram แสดง Disconnect

5.4.3 Activity Diagram

Activity Diagram เป็นไดอะแกรมที่ใช้ในการช่วยออกแบบอัลกอริทึมหลักๆ ของระบบ ในที่นี้ได้ทำการออกแบบฟังก์ชันการทำงานหลักๆของระบบ ซึ่งได้มาจาก Use Case ได้ดังนี้

5.4.3.1 Activity Diagram แสดง Set scan mode

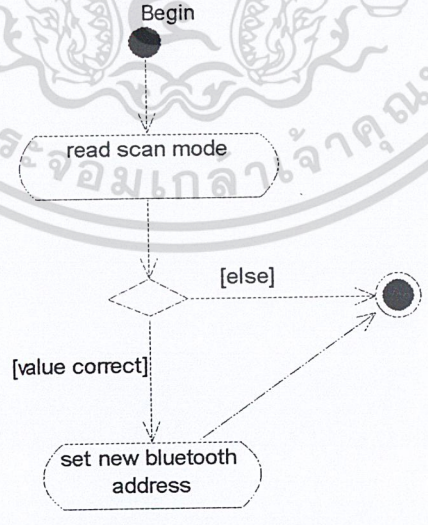
เป็นการออกแบบอัลกอริทึมของการกำหนดคุณสมบัติการสแกนโหมดของอุปกรณ์ แสดงในรูปที่ 5-21



รูปที่ 5-21 Activity Diagram แสดง Set scan mode

5.4.3.2 Activity Diagram แสดง Set bluetooth address

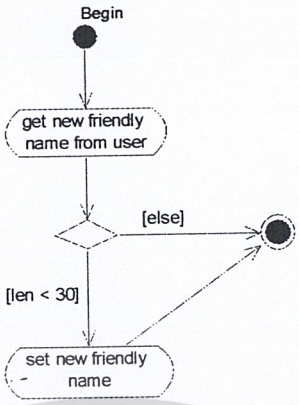
เป็นการออกแบบอัลกอริทึมของการกำหนดค่าแอดเดรสใหม่ให้กับอุปกรณ์ แสดงในรูปที่ 5-22



รูปที่ 5-22 Activity Diagram แสดง Set bluetooth address

5.4.3.3 Activity Diagram แสดง Set friendly name

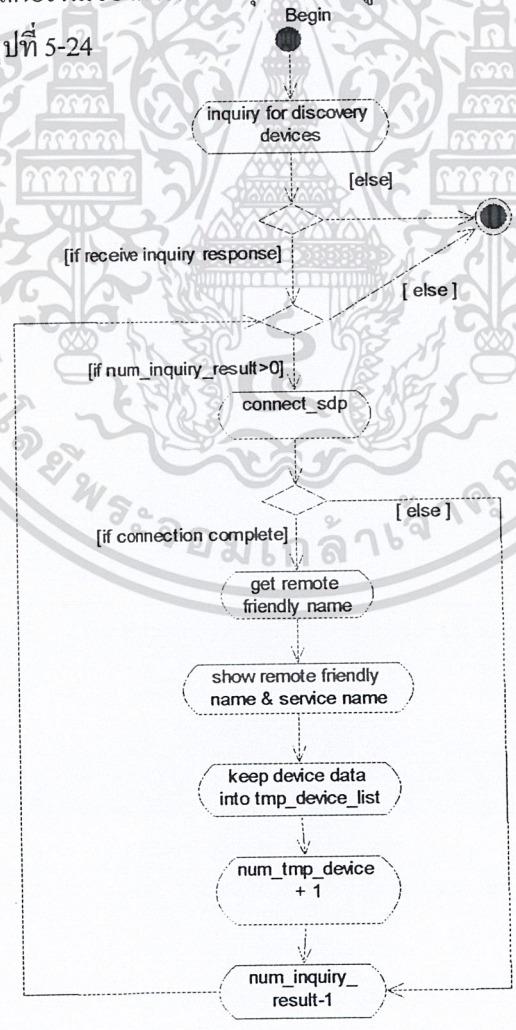
เป็นการออกแบบอัลกอริทึมของการกำหนดเฟรนรี่เนมใหม่ให้กับอุปกรณ์ แสดงในรูปที่ 5-23



รูปที่ 5-23 Activity Diagram แสดง Set friendly name

5.4.3.4 Activity Diagram แสดง Find all devices

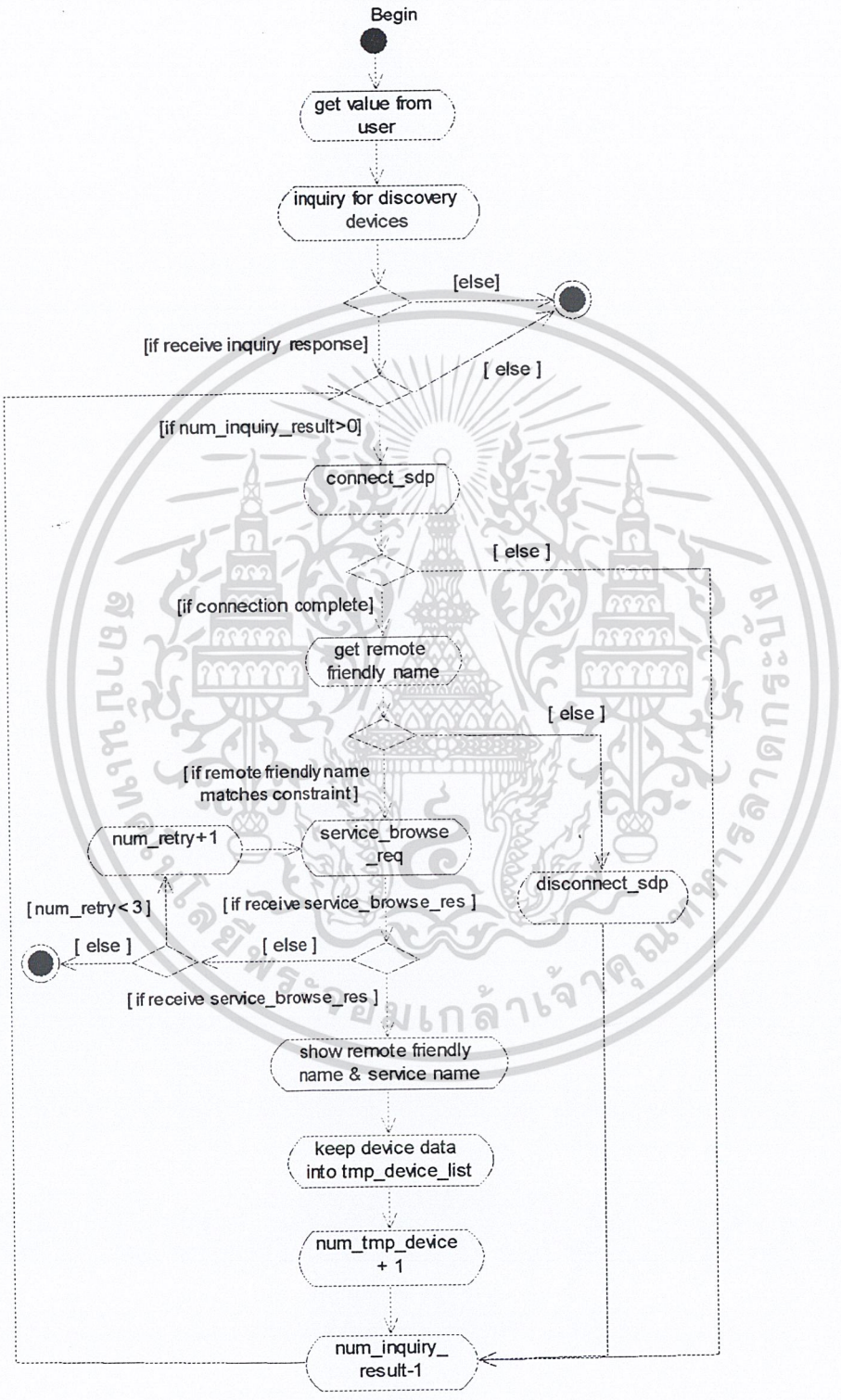
เป็นการออกแบบอัลกอริทึมของการค้นหาอุปกรณ์ที่อยู่ในเครือข่ายทั้งหมด แล้วทำการเชื่อมต่อกับอุปกรณ์ตัวนั้น แสดงในรูปที่ 5-24



รูปที่ 5-24 Activity Diagram แสดง Find all devices

5.4.3.5 Activity Diagram แสดง Find specific devices

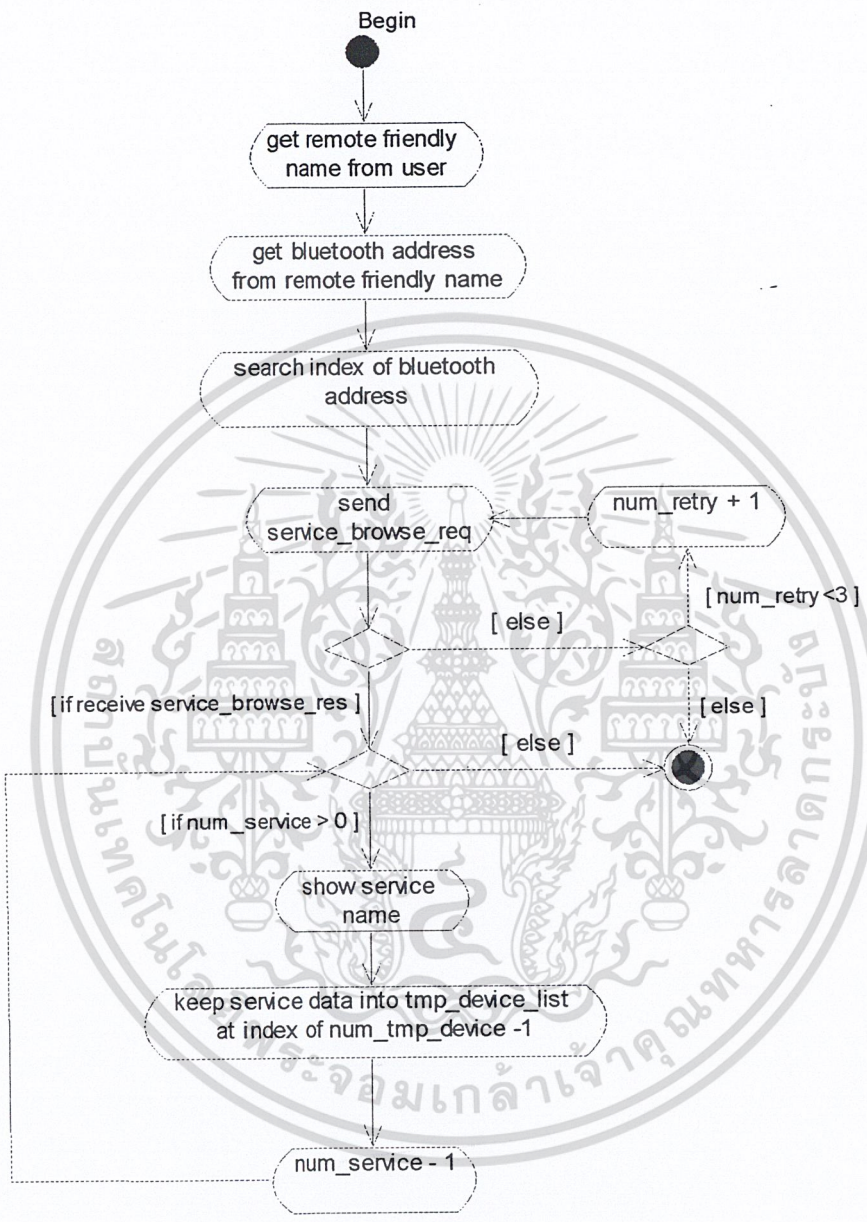
เป็นการออกแบบอัลกอริทึมของการค้นหาอุปกรณ์ที่อยู่ในเครือข่ายโดยระบุเฟรนรี่เนมหรือตัวอักษรแล้วทำการเชื่อมต่อกับอุปกรณ์ตัวนั้น แสดงในรูปที่ 5-25



รูปที่ 5-25 Activity Diagram แสดง Find specific devices

5.4.3.6 Activity Diagram แสดง Find all services

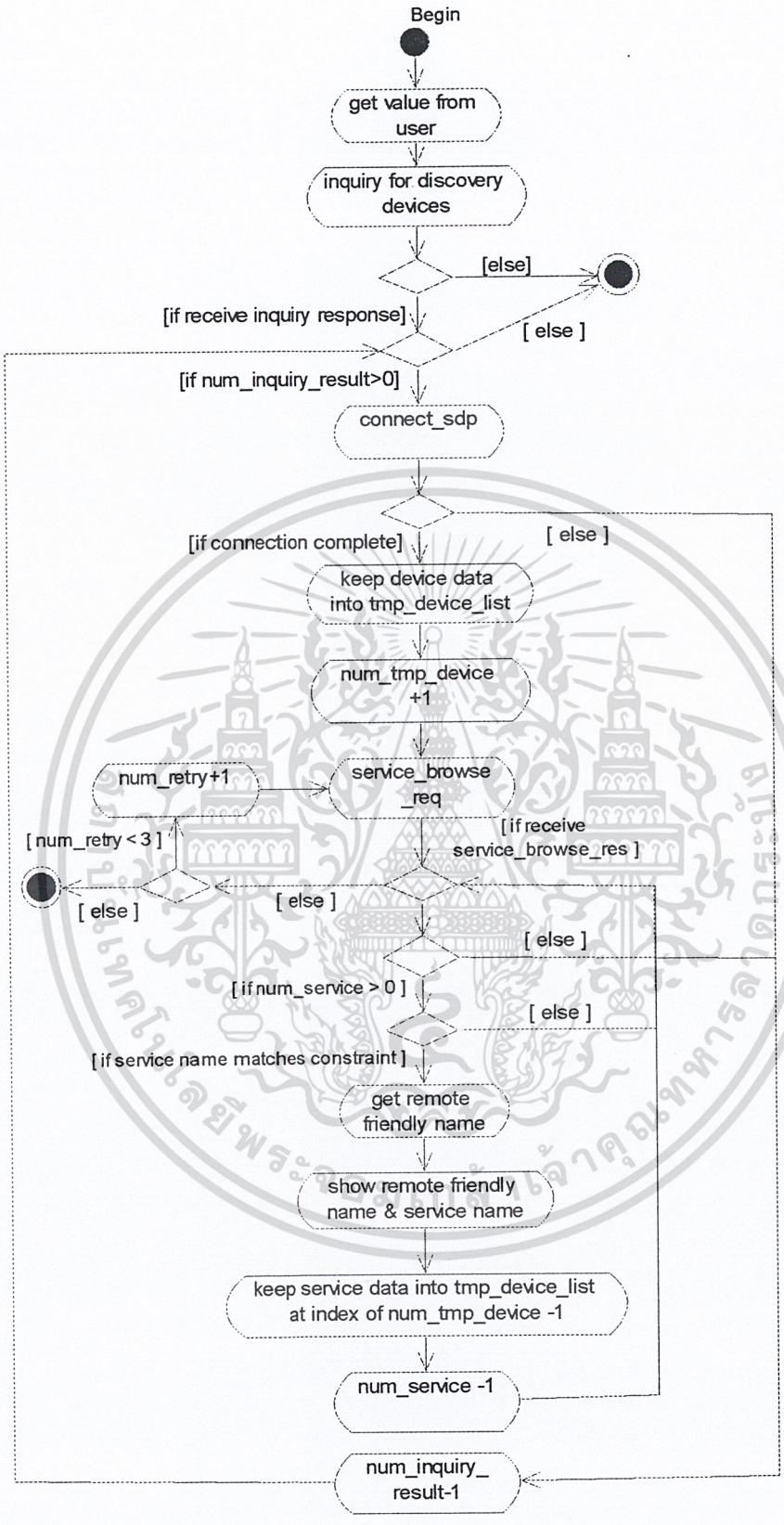
เป็นการออกแบบอัลกอริทึมของการค้นหาการบริการของอุปกรณ์ที่เชื่อมต่ออยู่ทั้งหมดแสดงในรูป 5-26



รูปที่ 5-26 Activity Diagram แสดง Find all services

5.4.3.7 Activity Diagram แสดง Find specific services

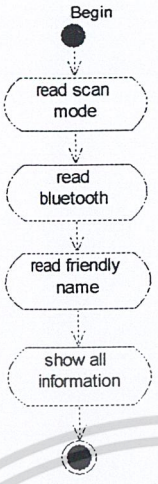
เป็นการออกแบบอัลกอริทึมของการค้นหาการบริการที่มีอยู่ในเครือข่าย โดยระบุชื่อการบริการหรือตัวอักษร แสดงในรูป 5-27



รูปที่ 5-27 Activity Diagram แสดง Find specific services

5.4.3.8 Activity Diagram แสดง Get my information

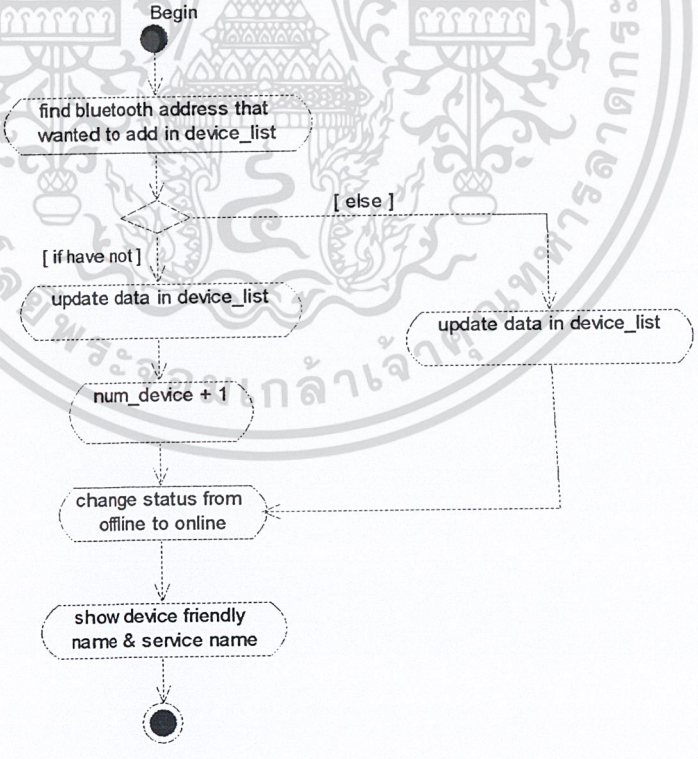
เป็นการออกแบบอัลกอริทึมของการให้ผู้ใช้ดูรายละเอียดของอุปกรณ์ ได้แก่ แอดเดรส เฟรนรี่เนมและค่าสแกนโหมด แสดงในรูปที่ 5-28



รูปที่ 5-28 Activity Diagram แสดง Get my information

5.4.3.9 Activity Diagram แสดง Add new device

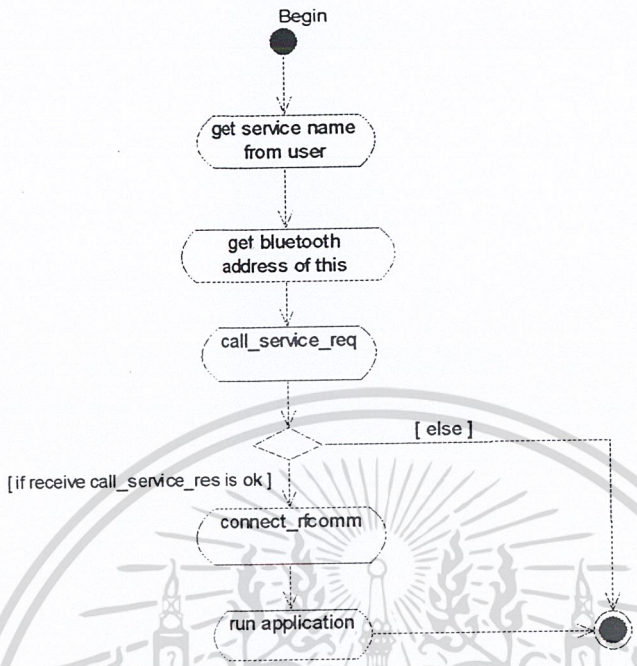
เป็นการออกแบบอัลกอริทึมของการให้ผู้ใช้ทำการเพิ่มอุปกรณ์ตัวใหม่ลงไปในรายการ ซึ่งเก็บรายละเอียดอุปกรณ์และการบริการของอุปกรณ์ตัวนั้น แสดงในรูป 5-29



รูปที่ 5-29 Activity Diagram แสดง Add new device

5.4.3.10 Activity Diagram แสดง Call service

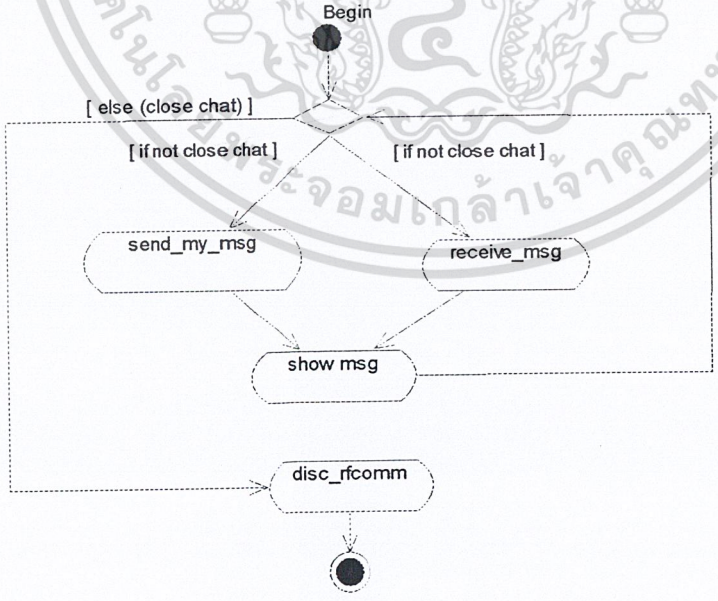
เป็นการออกแบบอัลกอริทึมในการให้ผู้ใช้ทำการขอใช้การบริการที่ต้องการกับอุปกรณ์ตัวอื่น แสดงในรูปที่ 5-30



รูปที่ 5-30 Activity Diagram แสดง Call service

5.4.3.11 Activity Diagram แสดง Chat

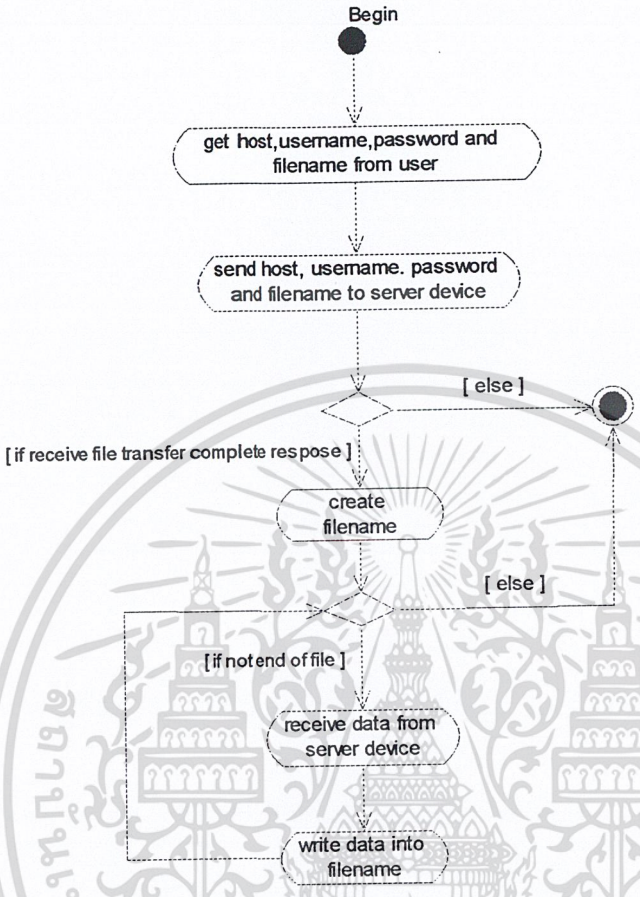
เป็นการออกแบบอัลกอริทึมของการใช้แอพพลิเคชันการคุยกันระหว่างอุปกรณ์ 2 ตัว แสดงในรูป 5-31



รูปที่ 5-31 Activity Diagram แสดง Chat

5.4.3.12 Activity Diagram แสดง File transfer

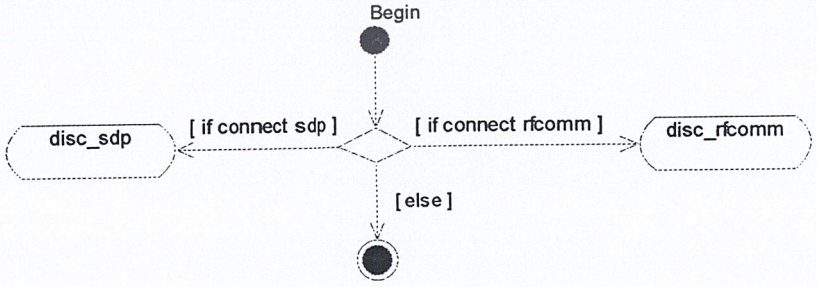
เป็นการออกแบบอัลกอริทึมของการขอใช้บริการการโอนถ่ายข้อมูลจากอุปกรณ์ตัวอื่น เนื่องจากอุปกรณ์ของเราไม่มีการบริการนี้ แสดงในรูปที่ 5-32



รูปที่ 5-32 Activity Diagram แสดง File transfer

5.4.3.13 Activity Diagram แสดง Disconnect

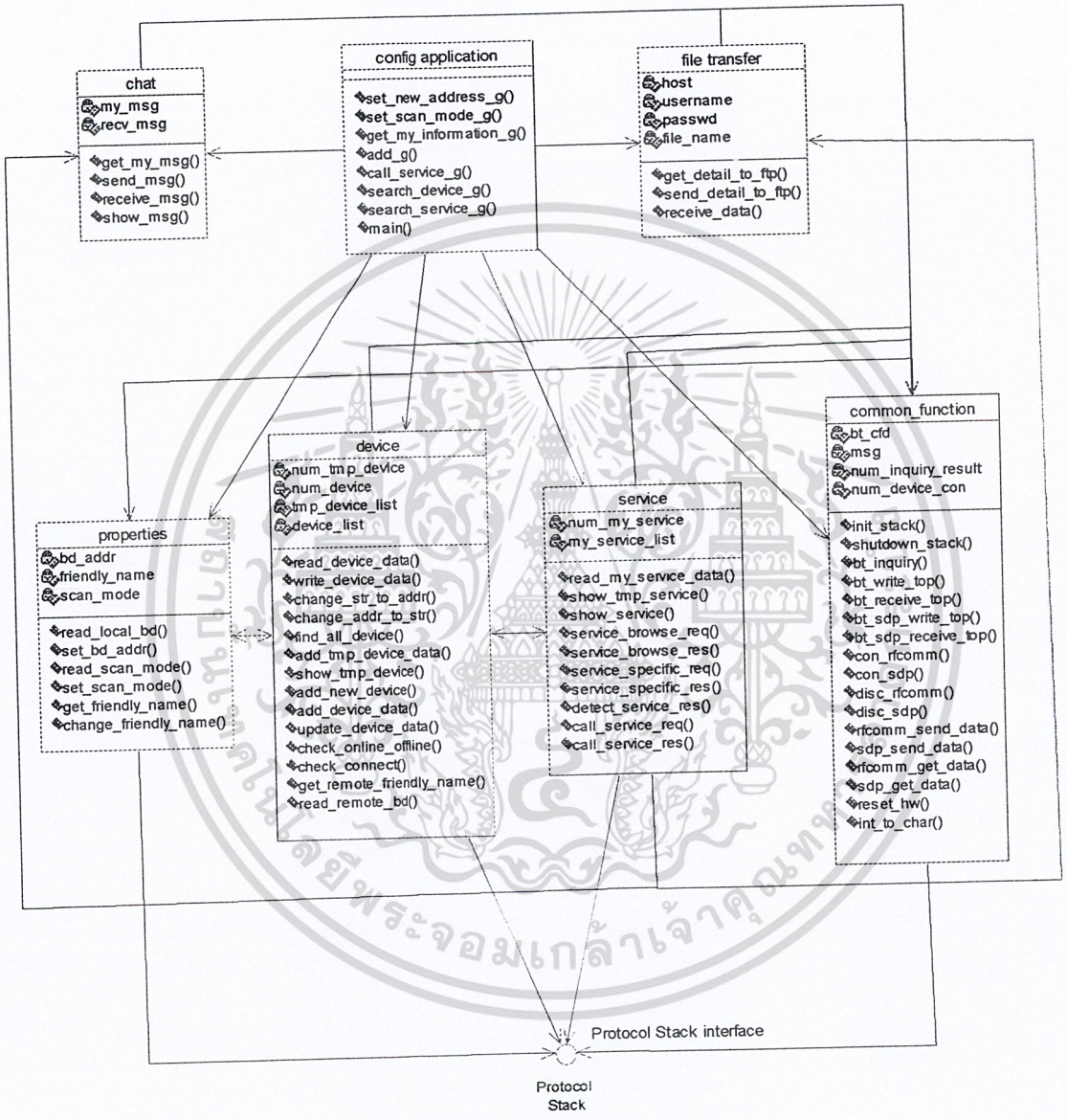
เป็นการออกแบบอัลกอริทึมของการยกเลิกการติดต่อกับอุปกรณ์ทั้งหมดและออกจากโปรแกรม แสดงในรูปที่ 5-33



รูปที่ 5-33 Activity Diagram แสดง Disconnect

5.4.4 Class Diagram

Class Diagram เป็นไดอะแกรมที่แสดงคลาสต่างๆที่มีในระบบ ทั้งในส่วนของ Device Manager , Config application และแอพพลิเคชัน (Chat , File transfer) แสดงในรูปที่ 5-34 ซึ่งคลาสต่างๆเหล่านี้จะมีความสัมพันธ์กับคลาสโพรโตคอลสแต็กด้วย



รูปที่ 5-34 คลาสไดอะแกรมของระบบ

จากรูประบบประกอบด้วยคลาสต่างๆ ซึ่งสามารถแบ่งเป็น 3 เลเยอร์ ดังนี้

1. โพรโตคอลสแต็ก เป็นเลเยอร์ล่างสุด ซึ่งในที่นี้ได้สร้างเป็นอินเทอร์เฟสเพื่อให้คลาสต่างๆ สามารถเรียกใช้งานได้
2. Device Manager เป็นเลเยอร์ตรงกลาง ประกอบด้วยคลาส 4 คลาส ดังนี้
   - คลาส properties เป็นคลาสที่ประกอบด้วยแอตทริบิวต์และฟังก์ชันการทำงานที่เกี่ยวกับคุณสมบัติต่างๆของตัวอุปกรณ์ ทั้งการอ่านค่าคุณสมบัติขึ้นมาดูและการกำหนดค่าคุณสมบัติใหม่
   - คลาส device เป็นคลาสที่ประกอบด้วยแอตทริบิวต์และฟังก์ชันการทำงานที่เกี่ยวกับอุปกรณ์ทั้งหมด ทั้งการค้นหาอุปกรณ์ การเชื่อมต่อกับอุปกรณ์ และการเพิ่มและอัพเดทข้อมูลของอุปกรณ์เหล่านั้น
   - คลาส service เป็นคลาสที่ประกอบด้วยแอตทริบิวต์และฟังก์ชันการทำงานที่เกี่ยวกับการบริการทั้งหมด ทั้งการค้นหาการบริการและการเรียกใช้การบริการที่ต้องการ
   - คลาส common_function เป็นคลาสที่ประกอบด้วยแอตทริบิวต์และฟังก์ชันการทำงานพื้นฐานที่คลาสต่างๆสามารถมาเรียกใช้ได้ เช่น ฟังก์ชัน init_stack() , shutdown_stack() เป็นต้น
3. แอพพลิเคชัน เป็นเลเยอร์บนสุดประกอบด้วยคลาส 3 คลาส ดังนี้
   - คลาส chat เป็นคลาสที่จำลองให้เป็นแอพพลิเคชันซึ่งประกอบด้วยแอตทริบิวต์และฟังก์ชันการทำงานที่เกี่ยวกับการคุยกันระหว่างอุปกรณ์ 2 ตัว
   - คลาส file_transfer เป็นคลาสที่จำลองให้เป็นแอพพลิเคชันประกอบด้วยแอตทริบิวต์และฟังก์ชันการทำงานที่เกี่ยวกับการส่งรายละเอียดสิ่งที่จำเป็นในการโอนถ่ายข้อมูลไปให้กับอุปกรณ์ตัวที่เราไปขอใช้บริการ ได้แก่ ชื่อโฮส ชื่อผู้ใช้ พาสเวิรด์และชื่อไฟล์ที่ต้องการ รวมทั้งฟังก์ชั่นการรับข้อมูล
   - คลาส config_application เป็นคลาสที่ประกอบด้วยแอตทริบิวต์และฟังก์ชันการทำงานที่เป็นอินเทอร์เฟซที่ติดต่อกับผู้ใช้ ซึ่งมีลักษณะเป็นกราฟฟิก

บทที่ 6
การทดสอบและผลการทดสอบ

6.1 วัตถุประสงค์ในการทดสอบ

ในส่วนของการทดสอบนี้จะเป็นส่วนของการทดสอบความสามารถของอุปกรณ์บลูทูธในการค้นหาอุปกรณ์และค้นหาการบริการที่มีอยู่ในเครือข่าย เพื่อเป็นการแสดงความสามารถการทำงานในเครือข่ายแอดฮอคผ่านบลูทูธ ซึ่งเป็นเครือข่ายที่มีความยืดหยุ่น เกิดขึ้นในทันทีทันใด ไม่มีการเตรียมตัวมาก่อนและไม่มีการจัดการจากส่วนกลาง ดังนั้นอุปกรณ์ที่ทำงานอยู่ในลักษณะของเครือข่ายแบบนี้นั้น ต้องเป็นอุปกรณ์ที่มีความสามารถในการจัดการการเชื่อมต่อด้วยตนเองได้ ซึ่งเป็นผลมาจากการจัดการของตัวจัดการอุปกรณ์ โดยมีวัตถุประสงค์ในการทดสอบดังนี้

1. แสดงความสามารถของอุปกรณ์บลูทูธในการค้นหาอุปกรณ์บลูทูธตัวอื่นๆที่อยู่ในเครือข่าย
2. แสดงความสามารถของอุปกรณ์บลูทูธในการค้นหาการบริการที่มีอยู่ในเครือข่าย

6.2 ขั้นตอนการทดสอบ

ในขั้นตอนของการทดสอบนี้ สามารถแบ่งการทดสอบออกเป็น 4 ส่วนด้วยกัน คือ

1. การทดสอบความสามารถในการค้นหาอุปกรณ์
2. การทดสอบความสามารถในการค้นหาการบริการ
3. การทดสอบความสามารถในการปรับแต่งคุณสมบัติต่างๆ
4. การทดสอบการเรียกใช้การบริการ

ซึ่งในการทดสอบแต่ละส่วนนั้น จะประกอบไปด้วยการทดสอบต่างๆ ดังนี้ คือ

6.2.1 การทดสอบความสามารถในการค้นหาอุปกรณ์

- ทดสอบการค้นหาอุปกรณ์

ในส่วนนี้เป็นการทดสอบความสามารถในการค้นหาอุปกรณ์ โดยอุปกรณ์จะต้องมีความสามารถในการค้นหาอุปกรณ์อื่นๆที่อยู่ในระยะเครือข่าย และแสดงรายชื่อของอุปกรณ์ที่ค้นพบขึ้นมาได้

- ทดสอบการแสดงสถานะของอุปกรณ์

ในส่วนนี้เป็นการทดสอบว่าอุปกรณ์อยู่ในลิสต์ของเราว่ามีสถานะการทำงานเป็นอย่างไรบ้าง โดยอุปกรณ์ที่อยู่ในลิสต์จะเป็นอุปกรณ์ที่ถูกค้นพบมาจากการค้นหาอุปกรณ์ก่อน แล้วจึงเพิ่มรายชื่อของอุปกรณ์นี้เข้าไปในลิสต์ เมื่อเปิดโปรแกรมขึ้นมาทำงานอีกครั้งจึงทำการตรวจสอบสถานะการทำงานของอุปกรณ์ที่อยู่ในลิสต์ ซึ่งสถานะการทำงานมีอยู่ด้วยกันสองอย่าง คือ เปิดการทำงาน (Online) และปิดการทำงาน (Offline)

- ทดสอบการค้นหาอุปกรณ์ที่มีชื่อตามตัวอักษรที่ต้องการ

ในส่วนนี้เป็นการค้นหาอุปกรณ์ในกรณีที่เราทราบตัวอักษรบางตัวในเฟรนรี่เนมของอุปกรณ์นั้น โดยการระบุตัวอักษรที่ต้องการลงไป หลังจากนั้นจะทำการค้นหาอุปกรณ์และแสดงรายชื่ออุปกรณ์ที่มีตัวอักษรนั้นปรากฏอยู่ในเฟรนรี่เนมขึ้นมา

6.2.2 การทดสอบความสามารถในการค้นหาการบริการ

- ทดสอบการค้นหาการบริการ

ในส่วนนี้เป็นการทดสอบความสามารถในการค้นหาการบริการ โดยอุปกรณ์จะต้องมีความสามารถในการค้นหาการบริการที่มีอยู่ในอุปกรณ์ต่างๆภายในระยะเครือข่าย และแสดงรายชื่อของการบริการที่ค้นพบขึ้นมาได้

- ทดสอบการค้นหาการบริการในอุปกรณ์ที่เปิดการทำงาน

สำหรับอุปกรณ์ที่มีรายชื่ออยู่ในลิสต์ที่เปิดการทำงานอยู่นั้น ในส่วนนี้จะเป็นการทดสอบเพื่อแสดงความสามารถในการค้นหาการบริการที่มีอยู่ในตัวอุปกรณ์ต่างๆ โดยจะแสดงเป็นรายชื่อของการบริการที่แต่ละอุปกรณ์มีให้ใช้

- ทดสอบการค้นหาการบริการที่มีชื่อตามตัวอักษรที่ต้องการได้

ในส่วนนี้เป็นการค้นหาการบริการในกรณีที่เราทราบชื่อของการบริการนั้น โดยการระบุตัวอักษรที่ต้องการลงไป หลังจากนั้นจะทำการค้นหาการบริการและแสดงรายชื่อการบริการที่มีตัวอักษรนั้นปรากฏอยู่ขึ้นมา

6.2.3 การทดสอบความสามารถในการปรับแต่งคุณสมบัติต่างๆ

- ทดสอบการกำหนดเฟรนรี่เนม (Friendly Name) ของอุปกรณ์

โดยทั่วไปแอดเดรสของอุปกรณ์บลูทูธจะเป็นตัวที่ระบุถึงอุปกรณ์บลูทูธแต่ละตัว แต่ในกรณีที่นำแอดเดรสของอุปกรณ์มาติดต่อกับผู้ใช้นั้นอาจทำให้เกิดความยุ่งยากในการใช้งานได้ ดังนั้นจึงได้นำเฟรนรี่เนม ซึ่งเปรียบเสมือนเป็นชื่อเล่นของอุปกรณ์ต่างๆมาใช้เป็นตัวระบุถึงอุปกรณ์แต่ละตัว การทดสอบในส่วนนี้นั้นเป็นการทดสอบการกำหนดเฟรนรี่เนมให้กับอุปกรณ์ ซึ่งจะใช้ชื่อนี้ในการอ้างอิงถึงอุปกรณ์ต่างๆ

- ทดสอบโหมดการสแกนของอุปกรณ์

ในบลูทูธนั้น ได้กำหนดโหมดการสแกนอยู่ 2 ชนิดคือ อินไควรี่สแกนและเพจสแกน

- การอินไควรี่สแกนนั้นเป็นการตอบรับกับอุปกรณ์ที่ทำการอินไควอยู่ อุปกรณ์ที่ทำการอินไควเป็นอุปกรณ์ที่ต้องการค้นหาอุปกรณ์อื่นๆที่อยู่ในระยะเครือข่าย เมื่อเปิดการทำงานของการอินไควรี่สแกนแล้ว อุปกรณ์ที่ทำการอินไควรี่อยู่จะสามารถพบอุปกรณ์ที่เปิดการอินไควรี่สแกนได้ ดังนั้นการเปิดการทำงานของการอินไควรี่สแกนนั้นเปรียบเสมือนเป็นการอนุญาตให้อุปกรณ์อื่นๆเห็นอุปกรณ์ของเราได้นั่นเอง
- การเพจสแกนเป็นการตอบรับอุปกรณ์ที่ทำการเพจอยู่ อุปกรณ์ที่ทำการเพจจะเป็นอุปกรณ์ที่ต้องการที่จะเชื่อมต่อด้วยกับอุปกรณ์ตัวใดตัวหนึ่งที่อยู่ในเครือข่าย เช่นเดียวกับการทำอิน

ไควรี่ เมื่อเปิดการทำงานของการเพจสแกนแล้ว อุปกรณ์ที่ทำการเพจอยู่จะสามารถเชื่อมต่อกับอุปกรณ์ที่ทำการเพจสแกนได้ ดังนั้นการเปิดการทำงานของการเพจสแกนจึงเปรียบเสมือนเป็นการอนุญาตให้มีการเชื่อมต่อกับอุปกรณ์ที่ร้องขอมาได้

ดังนั้นในส่วนนี้จะเป็นการทดสอบความสามารถของการทำงานทั้งสองโหมดนี้ โดยที่เมื่อเปิดการทำงานของการอินไควรี่สแกนอุปกรณ์จะสามารถถูกค้นพบโดยอุปกรณ์อื่นที่ทำการอินไควรี่อยู่ได้ และไม่สามารถถูกค้นพบเมื่ออุปกรณ์ทำการปิดการทำงานของการอินไควรี่ และเมื่อเปิดการทำงานของการเพจสแกนนั้น อุปกรณ์จะสามารถทำการเชื่อมต่อกับอุปกรณ์ที่ทำการร้องขอมาได้และไม่อนุญาตให้มีการเชื่อมต่อเมื่ออุปกรณ์ปิดการทำงานของเพจสแกน

- ทดสอบการเรียกดูข้อมูลของอุปกรณ์

ในส่วนนี้จะเป็นการเรียกดูข้อมูลต่างๆที่ผู้ใช้กำหนดให้กับอุปกรณ์ เช่น เฟรนรี่เนมของอุปกรณ์, โหมดการสแกนของอุปกรณ์ เป็นต้น

- ทดสอบการแสดงสถานะของอุปกรณ์

ในส่วนนี้เป็นการทดสอบว่าอุปกรณ์อยู่ในลิสต์ของเรานั้นมีสถานะการทำงานเป็นอย่างไรบ้าง ซึ่งสถานะการทำงานมีอยู่ด้วยกันสองอย่าง คือ เปิดการทำงาน (Online) และปิดการทำงาน (Offline)

- ทดสอบการเพิ่มรายชื่อของอุปกรณ์เข้าไปยังลิสต์

ลิสต์เป็นรายชื่อของอุปกรณ์ที่เราต้องการเก็บไว้ เพื่อใช้ในการติดต่อครั้งต่อไปเมื่อเปิดการทำงานของโปรแกรม ดังนั้นในส่วนนี้จึงเป็นการทดสอบการเพิ่มรายชื่อของอุปกรณ์เข้าไปยังลิสต์ โดยรายชื่อที่ได้มานั้นมาจากการค้นหาอุปกรณ์นั่นเอง

6.2.4 การทดสอบการเรียกใช้การบริการ

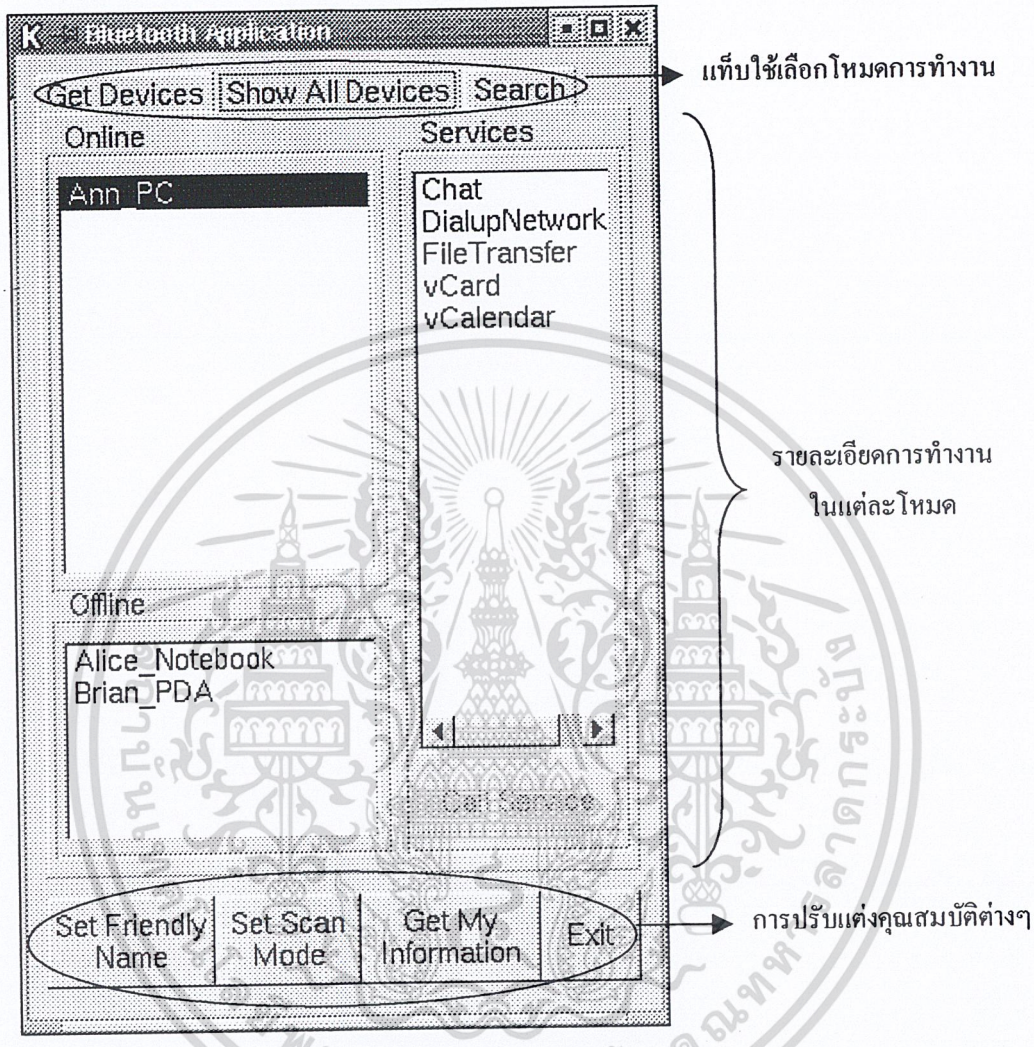
ในส่วนนี้จะเป็นการทดสอบการเรียกใช้บริการระหว่างอุปกรณ์ ซึ่งรายชื่อของการบริการที่เรียกใช้นี้มาจากการค้นหาการบริการของอุปกรณ์ที่มีชื่ออยู่ในลิสต์ โดยการบริการตัวอย่างที่ทำขึ้นมาเพื่อทดสอบ คือ การบริการส่งข้อความ และการบริการการรับส่งไฟล์ ซึ่งการบริการทั้งสองนี้เป็นตัวอย่างการบริการมีอยู่ในอุปกรณ์ทั้งสองฝั่ง และตัวอย่างการบริการที่อุปกรณ์ตัวหนึ่งขอใช้การบริการอีกฝั่งหนึ่ง ตามลำดับ

6.3 อุปกรณ์ที่ใช้ในการทดสอบ

1. โมดูลบลูทูธ 2 ตัว
2. สาย USB 2 เส้น
3. สายเคเบิลที่เป็นพอรต์อนุกรม 2 ด้าน (Null Modem) 2 เส้น
4. เครื่องคอมพิวเตอร์ที่รันระบบปฏิบัติการลีนุกส์และเปิดโปรแกรมของตัวจัดการอุปกรณ์บลูทูธ 2 เครื่อง

6.4 โปรแกรมที่ใช้ในการทดสอบ

โปรแกรมที่ใช้ในการทดสอบ เป็นส่วนที่ดึงความสามารถของตัวจัดการอุปกรณ์ขึ้นมาแสดงต่อผู้ใช้และเป็นตัวติดต่อกับผู้ใช้ ซึ่งมีส่วนต่างๆดังนี้



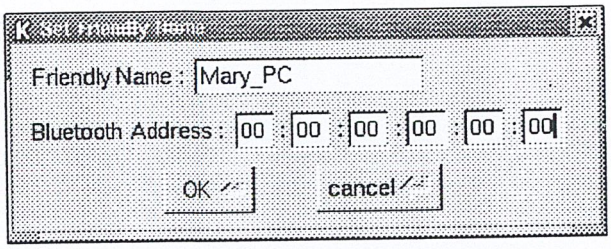
รูปที่ 6-1 ส่วนประกอบของโปรแกรม

6.5 ผลการทดสอบ

หลังจากที่ออกแบบการทดสอบในส่วนของขั้นตอนการทดสอบเรียบร้อยแล้ว ในหัวข้อนี้เป็นผลการทดสอบตามหัวข้อขั้นตอนการทดสอบ โดยจะเรียงลำดับตามการใช้งานจริง ซึ่งมีขั้นตอนดังนี้

- การกำหนดเฟรนรี่เนมของอุปกรณ์

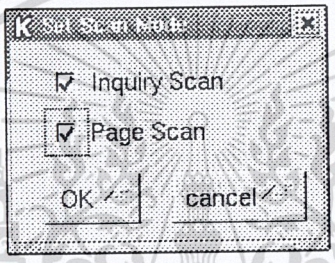
หลังจากที่เปิดโปรแกรมขึ้นมาเพื่อทำการทดสอบแล้ว ในตอนแรกนั้นจะเป็นการกำหนดค่าต่างๆให้กับอุปกรณ์ก่อนที่จะเริ่มใช้งาน โดยในส่วนแรกนี้จะเป็นการกำหนดเฟรนรี่เนมของอุปกรณ์โดยการกดปุ่ม  จะปรากฏหน้าต่างสำหรับให้ใส่เฟรนรี่เนมและกำหนดแอดเดรส ดังรูป 6-2



รูปที่ 6-2 หน้าต่างการกำหนดค่าแอดเดรส

- การกำหนดโหมดการสแกนของอุปกรณ์

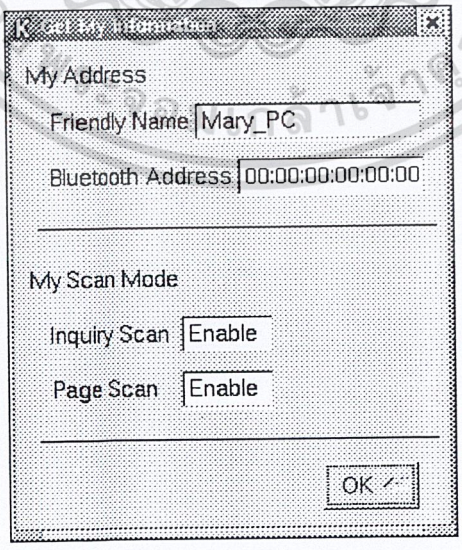
ในส่วนนี้เป็นการกำหนดโหมดการทำงานของการสแกน โดยกดที่ปุ่ม Set Scan Mode จะปรากฏหน้าต่างสำหรับเลือกโหมดดังรูป 6-3



รูปที่ 6-3 หน้าต่างการกำหนดโหมดการสแกน

- การเรียกดูข้อมูลของอุปกรณ์

หลังจากที่ทำการปรับแต่งคุณสมบัติต่างๆแล้ว สามารถเรียกดูข้อมูลต่างๆนั้นได้ โดยกดปุ่ม Get My Information จะปรากฏหน้าต่างที่แสดงข้อมูลของตัวอุปกรณ์ดังรูป 6-4



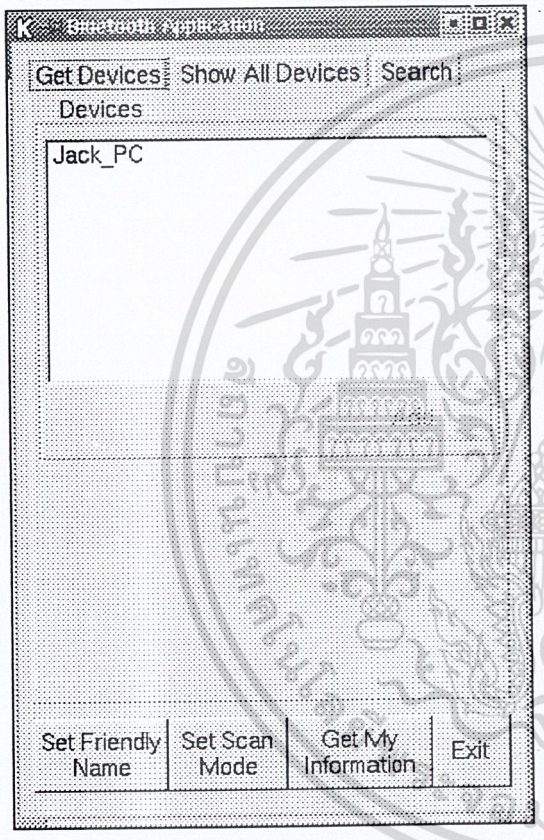
รูปที่ 6-4 หน้าต่างแสดงข้อมูลของอุปกรณ์

- การค้นหาอุปกรณ์

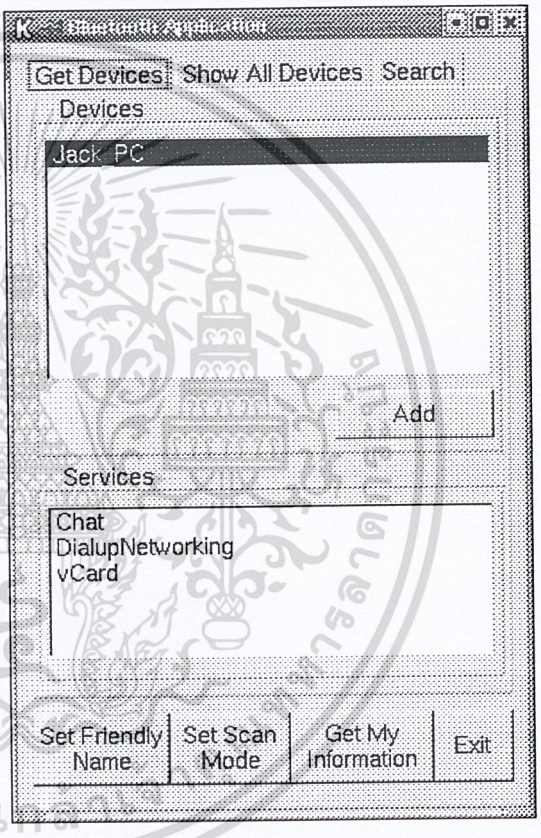
เมื่อเริ่มใช้งานครั้งแรกนั้น ยังไม่มีรายชื่ออุปกรณ์ที่อยู่ในลิสต์ ดังนั้นจึงต้องทำการค้นหาอุปกรณ์ที่อยู่ในเครือข่ายก่อน โดยการคลิกที่แท็บ Get Device จะทำการค้นหาอุปกรณ์ทั้งหมดที่อยู่ในบริเวณเครือข่าย และจะแสดงออกมาดังรูป 6-5

- การค้นหาการบริการ

หลังจากนั้นเมื่อคลิกที่รายชื่ออุปกรณ์ก็จะแสดงการบริการของอุปกรณ์ตัวนั้นทางด้านล่าง ดังรูป 6-6



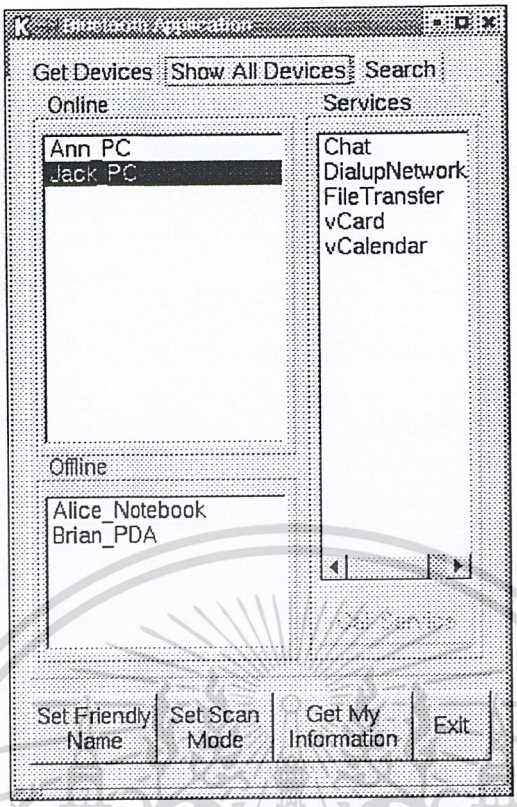
รูปที่ 6-5
แสดงรายชื่ออุปกรณ์ที่ค้นพบ



รูปที่ 6-6
แสดงรายชื่อการบริการในอุปกรณ์ที่ค้นพบ

- การเพิ่มรายชื่อของอุปกรณ์เข้าไปยังลิสต์

เมื่อดูรายละเอียดการบริการที่มีในอุปกรณ์ต่างๆแล้ว ถ้าต้องการเก็บชื่อของอุปกรณ์ตัวใดให้กดปุ่ม Add ชื่อของอุปกรณ์ตัวนั้นจะถูกเก็บลงลิสต์ ดังรูปที่ 6-7



รูปที่ 6-7 แสดงรายชื่ออุปกรณ์ที่ถูกเพิ่มชื่อไปในลิสต์

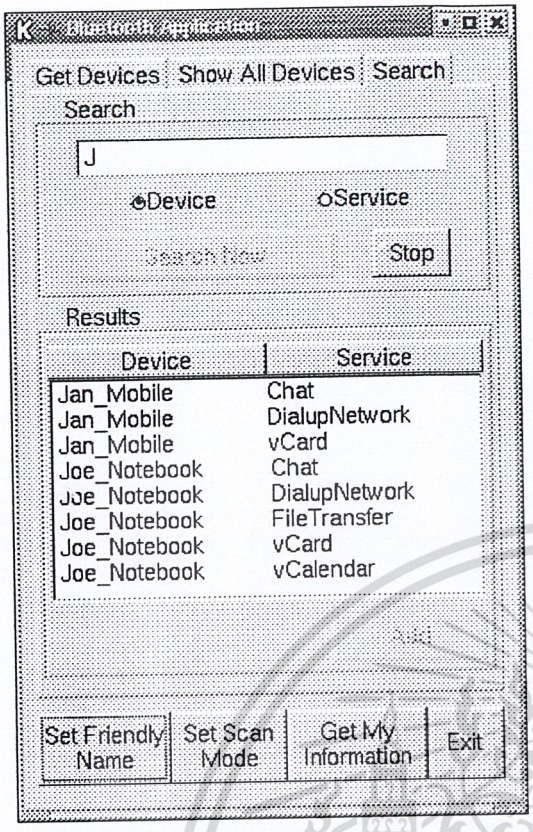
นอกจากนั้นในรูปที่ 6-7 นั้นยังแสดงถึงความสามารถในการแสดงสถานะของอุปกรณ์ โดยอุปกรณ์ที่อยู่ในลิสต์จะถูกตรวจสอบสถานะการทำงานว่าเปิดการทำงานหรือปิดการทำงานอยู่ด้วย โดยตัวที่เปิดการทำงานจะอยู่ในโหมด Online และตัวที่ปิดการทำงานจะอยู่ในโหมด Offline

- การค้นหาอุปกรณ์ที่มีชื่อตามตัวอักษรที่ต้องการ

ในกรณีที่ทราบเฟรนรี่เนมของอุปกรณ์ หรือทราบคำที่ปรากฏในเฟรนรี่เนมของอุปกรณ์นั้น สามารถทำการค้นหาได้ โดยการใส่คำที่ต้องการลงในช่องว่าง แล้วจึงกดปุ่ม Search Now ในช่องของ Result นั้นจะแสดงผลลัพธ์ของการค้นหาโดยจะแสดงรายชื่อของอุปกรณ์ที่ค้นพบพร้อมทั้งการบริการที่อุปกรณ์นั้นมีให้ใช้ ดังรูปที่ 6-8

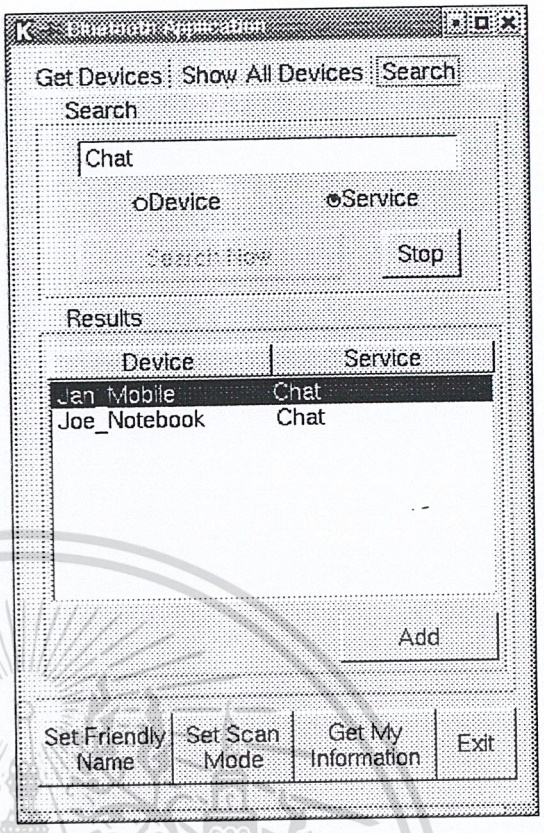
- การค้นหาการบริการที่มีชื่อตามตัวอักษรที่ต้องการ

ในกรณีที่ทราบชื่อของการบริการ หรือทราบคำที่ปรากฏอยู่ในการบริการนั้น สามารถทำการค้นหาได้โดยการใส่คำที่ต้องการลงในช่องว่าง แล้วจึงกดปุ่ม Search Now ในช่องของ Result นั้นจะแสดงผลลัพธ์ของการค้นหาโดยจะแสดงรายชื่อของการบริการที่ค้นพบพร้อมทั้งชื่อของอุปกรณ์ที่ให้บริการนั้น ดังรูปที่ 6-9



รูปที่ 6-8

แสดงการค้นหาอุปกรณ์โดยใส่ตัวอักษรที่ทราบ



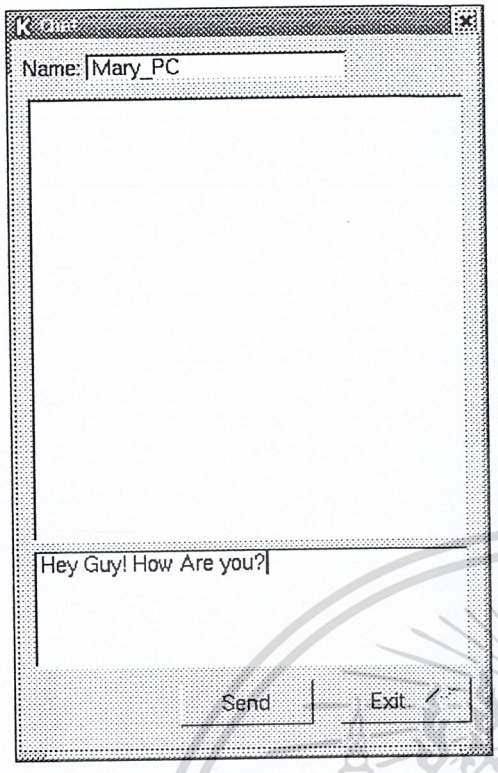
รูปที่ 6-9

แสดงการค้นหาการบริการโดยใส่ตัวอักษรที่ทราบ

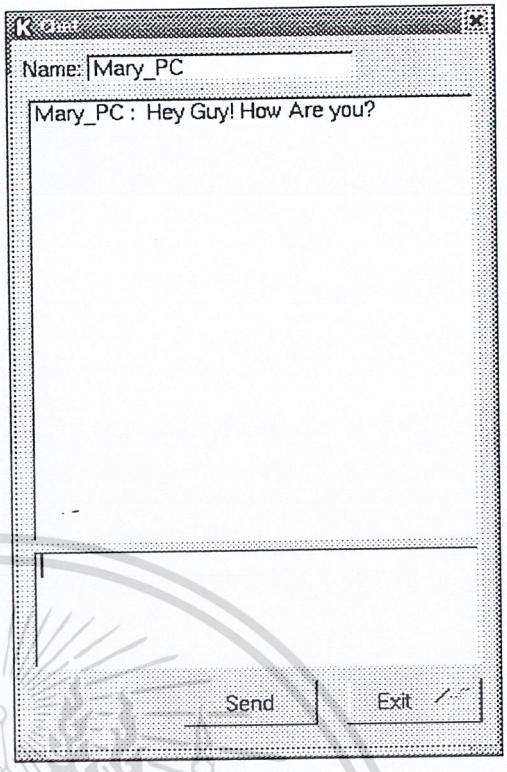
รูปที่ 6-8 และ 6-9 นี้เมื่อเราค้นหาการบริการหรืออุปกรณ์ที่ต้องการพบแล้ว สามารถกดปุ่ม Add เพื่อนำรายชื่อของอุปกรณ์นั้นเข้าไปในลิสต์ได้ทันที

- ทดสอบการเรียกใช้การบริการ

การบริการที่สร้างขึ้นมาเป็นตัวอย่างในการทดสอบการเรียกใช้นั้นมีอยู่ด้วยกันสองตัว คือ การรับส่งข้อความและการรับส่งไฟล์ โดยการคลิกแท็ปไปยังหน้า Show All Devices ซึ่งเป็นหน้าที่จะแสดงสถานะของอุปกรณ์ สำหรับอุปกรณ์ที่เปิดทำงานอยู่นั้น จะอยู่ในสถานะเปิดทำงาน เมื่อคลิกเลือกที่ชื่ออุปกรณ์จะขึ้นรายชื่อของการบริการที่อุปกรณ์นั้นมีให้ใช้ จากที่กล่าวมาแล้วว่าตัวอย่างการบริการที่สร้างขึ้นเพื่อทดสอบ คือ การรับส่งข้อความและการรับส่งไฟล์ เมื่อผู้ใช้คลิกเลือกชื่อของการบริการรับส่งข้อความและกดปุ่ม Call Service แล้วจะไปเริ่มต้นการทำงานของการบริการรับส่งข้อความ โดยขึ้นหน้าต่างดังรูปที่ 6-10 และเมื่อพิมพ์ข้อความและทำการส่งข้อความโดยกดปุ่ม Send แล้ว ทางอุปกรณ์อีกฝั่งหนึ่งจะขึ้นข้อความที่อุปกรณ์อีกตัวส่งข้อความมา และทำให้สามารถรับส่งข้อความติดต่อกันได้ ดังแสดงในรูปที่ 6-10 และ 6-11

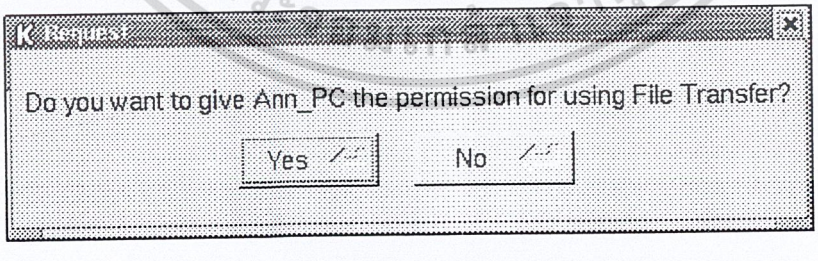


รูปที่ 6-10
แสดงหน้าต่างการบริการรับส่งข้อความ



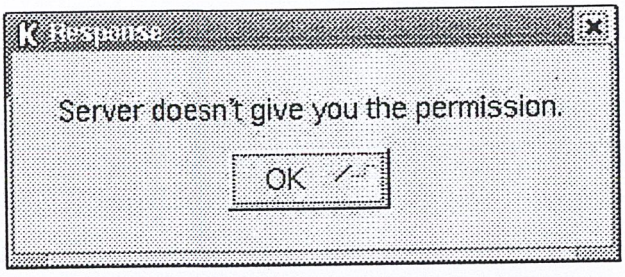
รูปที่ 6-11
แสดงหน้าต่างการบริการรับส่งข้อความ
ของอุปกรณ์อีกฝั่งหนึ่ง

ส่วนการบริการอีกตัวหนึ่งนั้น คือ การรับส่งไฟล์ การรับส่งไฟล์ในกรณีนี้เป็นการรับส่งไฟล์ที่ผู้ใช้ไม่สามารถดึงข้อมูลเองได้ ต้องทำการดึงข้อมูลผ่านอุปกรณ์ของผู้อื่นที่สามารถติดต่อไปยังอินเตอร์เนตได้ เมื่อผู้ใช้คลิกเลือกชื่อของการบริการรับส่งไฟล์และกดปุ่ม Call Service แล้วจะไปเริ่มต้นการทำงานของการบริการรับส่งไฟล์ โดยการร้องขอนี้จะไปปรากฏที่ฝั่งของเซิร์ฟเวอร์ ดังรูปที่ 6-12

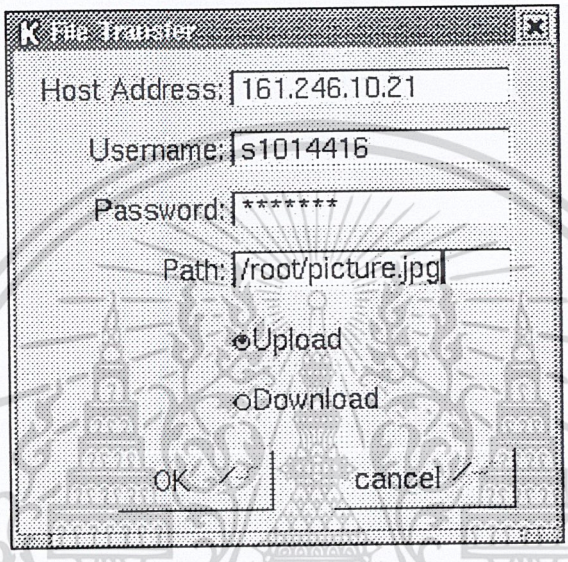


รูปที่ 6-12 หน้าต่างการร้องขอที่ไปปรากฏฝั่งเซิร์ฟเวอร์

ในกรณีที่ฝั่งเซิร์ฟเวอร์ไม่อนุญาตให้มีการใช้บริการรับส่งไฟล์นี้ ก็จะกดปุ่ม No ซึ่งจะปรากฏหน้าต่างในรูปที่ 6-13 แต่ถ้าเซิร์ฟเวอร์อนุญาตให้ใช้บริการรับส่งไฟล์นี้ ก็จะกดปุ่ม Yes ซึ่งจะปรากฏหน้าต่างในรูปที่ 6-14 ให้ผู้ใช้กรอกรายละเอียดเพื่อทำการรับส่งไฟล์จากข้อมูลของผู้ใช้เอง



รูปที่ 6-13 หน้าต่างแสดงข้อความเมื่อเซิร์ฟเวอร์ไม่อนุญาตให้ใช้การบริการรับส่งไฟล์



รูปที่ 6-14 หน้าต่างกำหนดค่าการรับส่งไฟล์เมื่อเซิร์ฟเวอร์อนุญาตให้ใช้การบริการรับส่งไฟล์

บทที่ 7
บทวิจารณ์และสรุป

7.1 บทวิจารณ์

ในส่วนนี้จะเป็นบทวิจารณ์ถึงสิ่งที่ได้ทำการทดสอบมา ซึ่งขอกล่าวเป็นหัวข้อตามลำดับที่ได้ทดสอบมาดังนี้

1. ความสามารถในการค้นหาอุปกรณ์
2. ความสามารถในการค้นหาการบริการ
3. ความสามารถในการปรับแต่งคุณสมบัติต่างๆ
4. การเรียกใช้การบริการ

7.1.1 ความสามารถในการค้นหาอุปกรณ์

อุปกรณ์ที่ทำการทดสอบสามารถค้นหาอุปกรณ์อื่นๆที่อยู่ในระยะเครือข่าย และแสดงรายชื่อของอุปกรณ์ที่ค้นพบขึ้นมาได้ ในกรณีของการทดสอบการแสดงสถานะของอุปกรณ์อุปกรณ์ที่ทำการทดสอบสามารถแสดงสถานะของอุปกรณ์ที่อยู่ในลิสต์ได้อย่างถูกต้อง โดยอุปกรณ์ที่เปิดอยู่จะอยู่ในสถานะเปิดการทำงานและอุปกรณ์ที่ปิดอยู่จะอยู่ในสถานะปิดการทำงาน และการทดสอบการค้นหาอุปกรณ์โดยการใส่ตัวอักษรที่ต้องการลงไปนั้น อุปกรณ์ที่ทำการทดสอบสามารถค้นหาอุปกรณ์ที่มีชื่อตามตัวอักษรที่กำหนดได้ แต่อย่างไรก็ตาม เนื่องจากข้อจำกัดของอุปกรณ์ที่มีอยู่ ทำให้สามารถทดสอบกับอุปกรณ์สูงสุดได้เพียงสามตัวเท่านั้น

7.1.2 การทดสอบความสามารถในการค้นหาการบริการ

อุปกรณ์ที่ทำการทดสอบสามารถค้นหาการบริการที่มีอยู่ในอุปกรณ์ต่างๆภายในระยะเครือข่ายและแสดงรายชื่อการบริการที่พบขึ้นมาได้ ในกรณีของการทดสอบการค้นหาการบริการในอุปกรณ์ที่เปิดการทำงานนั้นสามารถแสดงรายชื่อของการบริการได้อย่างถูกต้องเช่นกัน และการทดสอบการค้นหาการบริการโดยการใส่ตัวอักษรที่ต้องการลงไปนั้น อุปกรณ์ที่ทำการทดสอบสามารถค้นหาการบริการที่มีชื่อตามตัวอักษรที่กำหนดขึ้นมาได้

7.1.3 การทดสอบความสามารถในการปรับแต่งคุณสมบัติต่างๆ

อุปกรณ์ที่ทำการทดสอบสามารถปรับแต่งค่าต่างๆได้อย่างถูกต้อง โดยมีรายละเอียดในแต่ละหัวข้อ ดังนี้

- ทดสอบการกำหนดเฟรนรี่เนม (Friendly Name) ของอุปกรณ์

อุปกรณ์ที่ทำการทดสอบสามารถเปลี่ยนเฟรนรี่เนมของอุปกรณ์ได้ โดยเฟรนรี่เนมสามารถตั้งได้ไม่เกิน 30 ตัวอักษร สามารถเป็นตัวอักษรหรือตัวเลขก็ได้

- ทดสอบโหมดการสแกนของอุปกรณ์

ในโปรแกรมนั้น ได้กำหนดโหมดการสแกนอยู่ 2 ชนิดคือ อินไควรี่สแกนและเพจสแกน ในกรณีที่ปิดการทำงานทั้งอินไควรี่สแกนและเพจสแกน อุปกรณ์จะไม่สามารถถูกค้นพบและไม่สามารถเชื่อมต่อกับอุปกรณ์อื่นๆที่ทำการร้องขอมาได้ ถ้าปิดการทำงานของอินไควรี่สแกนแต่เปิดการทำงานของเพจสแกน อุปกรณ์จะสามารถไม่สามารถถูกค้นพบโดยอุปกรณ์อื่น แต่สามารถทำการเชื่อมต่อกับอุปกรณ์ที่ทำการร้องขอได้จะใช้ในกรณีที่อุปกรณ์นั้นเคยติดต่อกันมาแล้ว ถ้าเปิดการทำงานของการอินไควรี่สแกนและการเพจสแกน อุปกรณ์จะสามารถถูกค้นพบโดยอุปกรณ์อื่นและสามารถเชื่อมต่อกับอุปกรณ์ที่ร้องขอมาได้ ส่วนในกรณีที่เปิดการทำงานของอินไควรี่สแกนแต่ปิดการทำงานของเพจสแกนจะไม่สามารถทำได้

- ทดสอบการเรียกดูข้อมูลของอุปกรณ์

อุปกรณ์ที่ทดสอบสามารถเรียกดูข้อมูลของอุปกรณ์ได้ โดยค่าที่แสดงขึ้นมานั้นเป็นค่าที่ผู้ใช้กำหนดให้กับอุปกรณ์นั้น เช่น โหมดการทำงานของการสแกน หรือ เฟรนรี่เนม เป็นต้น

- ทดสอบการแสดงสถานะของอุปกรณ์

อุปกรณ์ที่ทำการทดสอบสามารถตรวจสอบสถานะการทำงานของอุปกรณ์ที่มีชื่ออยู่ในลิสต์ โดยอุปกรณ์ที่กำลังทำงานอยู่จะแสดงสถานะเปิดการทำงาน และอุปกรณ์ที่ปิดอยู่จะแสดงสถานะปิดการทำงาน

- ทดสอบการเพิ่มรายชื่อของอุปกรณ์เข้าไปยังลิสต์

อุปกรณ์ที่ทำการทดสอบสามารถเพิ่มรายชื่อของอุปกรณ์เข้าไปยังลิสต์ได้ โดยอุปกรณ์ที่เพิ่งจะถูกเพิ่มชื่อไปในลิสต์นั้นจะอยู่ในสถานะเปิดการทำงาน และจะถูกตรวจสอบสถานะทุกครั้งเมื่อเปิดโปรแกรมขึ้นมาใช้งานใหม่

7.1.4 การทดสอบการเรียกใช้การบริการ

อุปกรณ์ที่ทำการทดสอบสามารถเรียกใช้การบริการตัวอย่างที่ทำขึ้นมาเพื่อทดสอบได้ โดยการบริการตัวอย่างที่ทำขึ้นมาเพื่อทดสอบ คือ การบริการส่งข้อความ และการบริการการรับส่งไฟล์ ซึ่งการบริการทั้งสองนี้เป็นตัวอย่างการบริการมีอยู่ในอุปกรณ์ทั้งสองฝั่ง และตัวอย่างการบริการที่อุปกรณ์ตัวหนึ่งขอใช้การบริการอีกฝั่งหนึ่ง ตามลำดับ

7.2 แนวทางการพัฒนาต่อ

เนื่องจากตัวจัดการอุปกรณ์ที่ได้ทำการพัฒนาขึ้นมานั้นอยู่ในขอบเขตของการค้นหาอุปกรณ์และการค้นหาการบริการที่มีอยู่ในเครือข่าย และจากประสบการณ์ที่ได้ทำการศึกษาในเรื่องนี้มาแล้วนั้น ทำให้มองแนวทางการพัฒนาต่อออกเป็นสองหัวข้อหลักด้วยกัน คือ

- การพัฒนาในส่วนของตัวจัดการอุปกรณ์
- การพัฒนาในส่วนของการค้นหาการบริการ

7.2.1 การพัฒนาในส่วนของตัวจัดการอุปกรณ์

บลูทูธไม่มีมาตรฐานกลางสำหรับตัวจัดการอุปกรณ์ แต่ได้มีการพูดถึงประเด็นที่ตัวจัดการอุปกรณ์ของบลูทูธจะสามารถจัดการในเรื่องต่างๆดังต่อไปนี้ คือ

- การจัดการการเชื่อมต่อ
- การจัดการในเรื่องของความปลอดภัย
- การจัดการในเรื่องของการอินทิเกรตโปรแกรมประยุกต์
- การจัดการในเรื่องของการเก็บเงิน
- การจัดการในเรื่องของคุณภาพของการบริการ
- การจัดการในเรื่องของการออกแบบส่วนติดต่อกับผู้ใช้

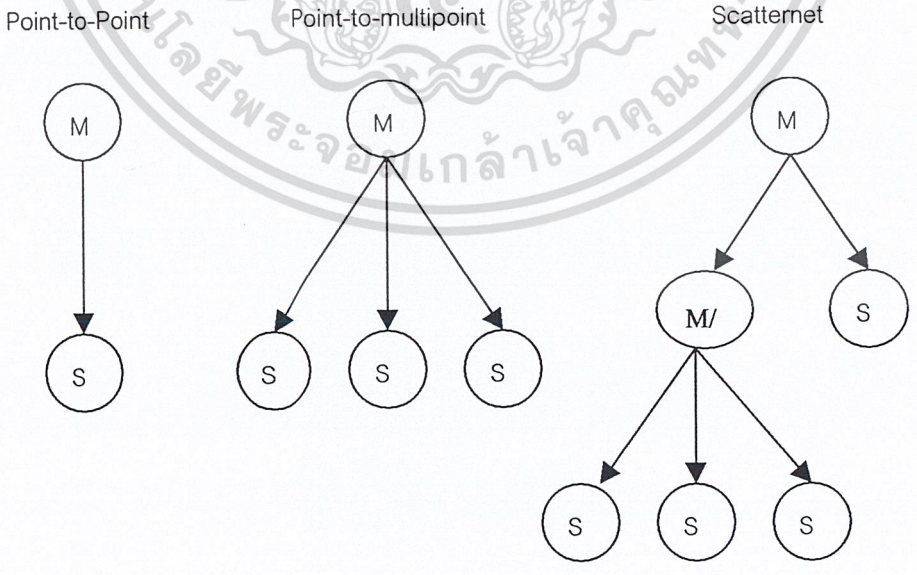
ดังนั้นแนวทางการพัฒนาต่อในส่วนนี้ คือ เป็นการพัฒนาตัวจัดการอุปกรณ์ในด้านต่างๆที่กล่าวมา ซึ่งมีรายละเอียดดังต่อไปนี้

- **การจัดการการเชื่อมต่อ**

การเชื่อมต่อของบลูทูธนั้นเกิดขึ้นได้หลายลักษณะ ดังรูปที่ 7.1 ซึ่งสามารถแบ่งการเชื่อมต่อออกเป็น 3 ลักษณะ คือ

- การเชื่อมต่อแบบจุดต่อจุด (Point-to-Point)
- การเชื่อมต่อแบบจุดต่อหลายจุด (Point-to-Multipoint)
- การเชื่อมต่อแบบสแคตเตอร์เนต (Scatternet)

ซึ่งการเชื่อมต่อแบบต่างๆนั้นจะมีความซับซ้อนในการจัดการที่แตกต่างกันออกไป ดังนั้นตัวจัดการอุปกรณ์สามารถเข้ามามีบทบาทในการจัดการการเชื่อมต่อนี้ได้ โดยซ่อนความยุ่งยากในกระบวนการการเชื่อมต่อแบบต่างๆ ทำให้ผู้ใช้ไม่รู้สึกถึงความยุ่งยากในการใช้งาน



รูปที่ 7-1 การเชื่อมต่อในรูปแบบต่างๆ

- **การจัดการในเรื่องของความปลอดภัย**

เนื่องจากบลูทูธเป็นการสื่อสารไร้สายโดยใช้คลื่นวิทยุ ดังนั้นประเด็นในเรื่องของความปลอดภัยจึงเป็นสิ่งสำคัญ ที่จริงแล้วการรับส่งข้อมูลในบลูทูธนั้นใช้หลักการของการกระโดดของความถี่ ซึ่งเป็นการรับส่งข้อมูลที่มีความปลอดภัยสูงอยู่แล้ว แต่ตัวจัดการอุปกรณ์สามารถเพิ่มระดับของความปลอดภัยในการทำงานของอุปกรณ์ โดยเพิ่มความปลอดภัยในด้านต่างๆดังนี้ได้ เช่น เมื่อมีการร้องขอในกรณีที่อุปกรณ์ระยะไกลส่งการร้องขอการใช้บริการมายังอุปกรณ์ของเรา หรือ การพิสูจน์สิทธิ์ และการพิสูจน์ตน เป็นต้น

- **การจัดการในเรื่องของการอินทิเกรตโปรแกรมประยุกต์**

โปรแกรมประยุกต์เป็นความสามารถในด้านต่างๆที่กำหนดไว้ในบลูทูธโพรไฟล์ ซึ่งโปรแกรมประยุกต์นั้นอาจเป็นโปรแกรมที่ถูกออกแบบมาเพื่อใช้ในอุปกรณ์นั้นๆโดยเฉพาะ เช่น ในกรณีของหูฟังไร้สาย เป็นต้น หรืออาจเป็นโปรแกรมที่สามารถจะนำไปอินทิเกรตเข้ากับอุปกรณ์ต่างๆเพื่อให้สามารถใช้งานในด้านนั้นได้ เช่น โปรแกรมการเชื่อมต่อไปยังเครือข่าย (Dial Up Networking) ซึ่งเหมาะที่จะนำไปใช้กับอุปกรณ์ต่างๆได้ เป็นต้น ดังนั้นในกรณีที่โปรแกรมประยุกต์มีลักษณะเพื่อจะนำไปอินทิเกรตเข้ากับบลูทูธโพรโตคอลแสตกที่อยู่ในอุปกรณ์อื่นๆนั้น ตัวจัดการอุปกรณ์จะเป็นตัวที่ช่วยให้โปรแกรมประยุกต์เหล่านี้สามารถนำมาอินทิเกรตเข้ากับโพรโตคอลแสตกได้ง่าย ช่วยลดความยุ่งยากให้กับผู้พัฒนาโปรแกรมประยุกต์เพราะไม่ต้องไปเขียนโปรแกรมติดต่อกับโพรโตคอลแสตกโดยตรง ซึ่งจะมีความซับซ้อนและยุ่งยากกว่า

- **การจัดการในเรื่องของการเก็บเงิน**

การเก็บเงินนั้นคิดจากการใช้บริการ ถึงแม้ว่าบลูทูธจะใช้ความถี่ในช่วงของ ISM Band ซึ่งเป็นความถี่อิสระแต่ไม่ได้หมายความว่าการบริการที่ให้ผ่านบลูทูธนั้นจะไม่คิดค่าใช้จ่ายเลย ยกตัวอย่างเช่น การใช้โทรศัพท์มือถือ หรือ จุดที่ให้บริการการเข้าถึงระบบแลน เป็นการบริการที่ผู้ให้สามารถคิดค่าใช้จ่ายกับผู้ที่รับบริการได้ ซึ่งตัวจัดการอุปกรณ์จะเข้ามาดูแลในเรื่องนี้ อาจจะคิดค่าใช้จ่ายตามเวลาที่ใช้บริการหรืออาจจะคิดค่าใช้จ่ายตามจำนวนข้อมูลที่ใช้บริการก็เป็นได้ ขึ้นกับความเหมาะสมกับการนำมาใช้กับโปรแกรมประยุกต์

- **การจัดการในเรื่องของคุณภาพของการบริการ**

ตัวจัดการอุปกรณ์สามารถเข้ามาช่วยดูแลคุณภาพของการให้บริการ โดยจัดการในเรื่องของความจุในการเชื่อมต่อ ซึ่งจะดูแลในเรื่องชนิดของแพ็กเกตที่ใช้ในการรับ-ส่งข้อมูล เพื่อให้เป็นไปตามคุณภาพของการบริการในงานนั้นๆ เนื่องจากการบริการที่ต่างกันย่อมต้องการแบนวิธที่ใช้ในการรับ-ส่งข้อมูลที่ต่างกันด้วยและคุณภาพของการบริการที่ต่างกันด้วย เช่น การบริการการรับส่งไฟล์จะเป็นการบริการที่ต้องการแบนวิธที่สูง แต่สามารถทนต่อการรับส่งข้อมูลที่ช้าได้ ซึ่งต่างกับการใช้งานโทรศัพท์ที่ต้องการแบนวิธที่ต่ำกว่าแต่ต้องการความต่อเนื่องของข้อมูล

- **การจัดการในเรื่องของการออกแบบส่วนติดต่อกับผู้ใช้**

เนื่องจากกระบวนการในการติดต่อกับอุปกรณ์บลูทูธตัวอื่นๆ และกระบวนการในการใช้การบริการของอุปกรณ์ตัวอื่นนั้นค่อนข้างที่จะยุ่งยากและซับซ้อน ดังนั้นตัวจัดการอุปกรณ์จึงมาช่วยในเรื่องของการติดต่อกับผู้ใช้ โดยที่จะต้องซ่อนความยุ่งยากในการติดต่อกับโพรโตคอลแสตกของตนเอง และการติดต่อกับโพรโตคอลแสตกของอุปกรณ์อื่นๆที่ติดต่อด้วย ซึ่งผลที่ได้จะทำให้ผู้ใช้งานรู้สึกง่ายและสะดวกสบายในการใช้งานอุปกรณ์บลูทูธซึ่งเป็นจุดประสงค์หลักอย่างหนึ่งของบลูทูธ

7.2.2 การพัฒนาในส่วนของการค้นหาการบริการ

เนื่องจากโพรโตคอลการค้นหาการบริการที่มีอยู่ในโพรโตคอลแสตกของบลูทูธเป็นเพียงกระบวนการในการค้นหาการบริการที่มีอยู่ในเครือข่ายขณะนั้น เพื่อให้อุปกรณ์ต่างๆสามารถค้นหาการบริการที่มีอยู่ได้ แต่ไม่ได้กำหนดหรือระบุถึงวิธีในการเข้าใช้การบริการที่สามารถค้นหามาได้เลย ดังนั้นเพื่อให้ระบบเครือข่ายแบบแอดฮอคแบบมีความสมบูรณ์แบบมากยิ่งขึ้น จึงควรนำโพรโตคอลที่ใช้ในการค้นหาการบริการตัวอื่นๆที่มีอยู่มาใช้งานร่วมกับบลูทูธ โดยโพรโตคอลการค้นหาการบริการเดิมที่มีอยู่นั้นจะใช้ในการค้นหาการบริการที่มีอยู่ในเครือข่าย และเมื่อได้รายชื่อพร้อมทั้งข้อมูลในการเชื่อมต่อเพื่อใช้งานบริการนั้นๆแล้ว จึงนำโพรโตคอลการค้นหาบริการตัวอื่นที่นำมาใช้ร่วมกับบลูทูธนี้เพื่อใช้ในการเข้าถึงการใช้การบริการ หรือเรียกใช้การบริการที่มีอยู่ในเครือข่ายต่อไป

สืบเนื่องมาจากปัจจุบันมีการบริการที่มีให้ใช้ในระบบเครือข่ายได้เพิ่มขึ้นเป็นจำนวนมาก ทั้งการบริการเดิมที่มีให้ใช้ เช่น การให้บริการจากพรินเตอร์, สแกนเนอร์ และเครื่องโทรสาร และการบริการใหม่ๆที่เพิ่มขึ้นมา เช่น การให้บริการข่าวสารผ่านอินเตอร์เน็ต, บริการดาวน์โหลดเพลง ทำให้ความสามารถในการค้นหาและการใช้บริการที่มีอยู่ในเครือข่ายมีความจำเป็นมากยิ่งขึ้น ซึ่งก่อให้เกิดโพรโตคอลการค้นหาการบริการต่างๆขึ้นมากมาย เช่น SLP (Service Location Protocol), Jini, UpnP (Universal Plug and Play), และ Salutaion ภายใต้สภาพแวดล้อมของการค้นหาการบริการนี้ การบริการจำเป็นต้องประกาศตัวเอง ซึ่งจะบอกรายละเอียดเกี่ยวกับความสามารถของการบริการและข้อมูลข่าวสารที่จำเป็นต่อการใช้บริการนั้นๆ โพรโตคอลการค้นหาการบริการที่มีอยู่ในเครือข่ายต้องทำให้กระบวนการการค้นหาการบริการที่มีอยู่ในเครือข่ายเป็นไปอย่างหยืดหยุ่น และต้องให้ข้อมูลข่าวสารดังต่อไปนี้

- การค้นหา (Search) และการดู (Browse) การบริการ
- เลือกการบริการที่มีอยู่ได้อย่างเหมาะสม
- ใช้การบริการที่มีอยู่ให้เป็นประโยชน์

ในมุมมองของผู้ใช้บริการ การค้นหาการบริการเป็นการช่วยให้การค้นหาและการใช้บริการง่ายยิ่งขึ้น ส่วนมุมมองของผู้ดูแลระบบนั้น การค้นหาการบริการเป็นการช่วยให้การสร้างและการดูแลการบริการง่ายยิ่งขึ้น โดยเฉพาะเมื่อมีการบริการใหม่ๆและอุปกรณ์ใหม่ๆมาอยู่ในเครือข่าย

โพรโตคอลการบริการที่มีอยู่ในเครือข่ายที่เป็นที่รู้จักกันในปัจจุบัน ได้แก่

- Service Location Protocol (SLP) พัฒนาโดย IETF
- Jini พัฒนาโดย Sun

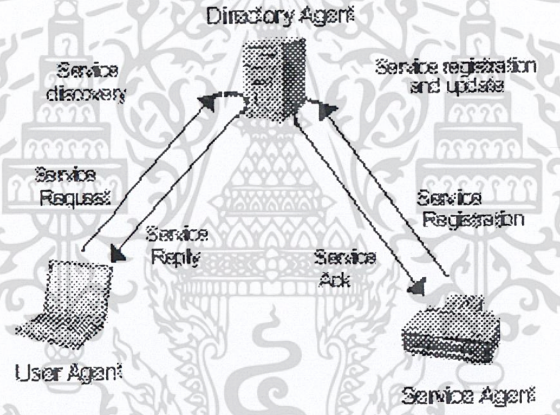
- Salutaion
- Microsoft's Universal Plug and Play (UPnP)

Service Location Protocol (SLP)

SLP ถูกพัฒนาขึ้นโดยกลุ่ม IETF โดยมีจุดประสงค์เพื่อให้เป็นมาตรฐานกลางที่ไม่ผูกมัดกับใคร โดยที่ออกแบบมาเพื่อใช้กับเครือข่าย TCP/IP และสามารถรองรับการใช้งานกับเครือข่ายขนาดใหญ่ได้

สถาปัตยกรรมของ SLP นั้นประกอบไปด้วย 3 องค์ประกอบหลักๆ คือ

- User Agents (UA) เป็นส่วนค้นหาการบริการเพื่อประโยชน์ของผู้เรียกใช้ (User or Application)
- Service Agents (SA) เป็นส่วนที่ประกาศลักษณะของการบริการ
- Directory Agents (DA) เป็นส่วนที่รวบรวมแอดเดรสของการบริการ, ข้อมูลข่าวสารที่ได้รับมาจาก SAs และตอบการร้องขอการบริการของ UAs



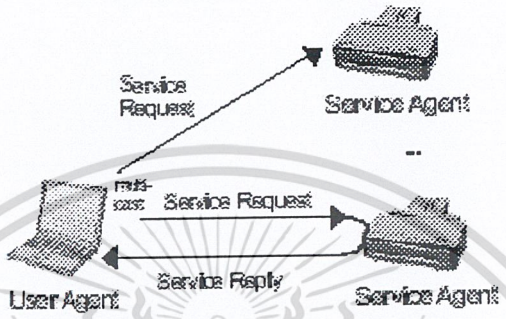
รูปที่ 7-1 SLP เอเจนต์ และทรานแซคชันสำหรับการค้นหาการบริการและการลงทะเบียน

รูปที่ 7-1 แสดงถึงการตอบโต้ระหว่างเอเจนต์ 3 ตัว เมื่อมีการบริการใหม่เชื่อมต่อเข้ามายังเครือข่าย SA จะติดต่อไปยัง DA เพื่อประกาศให้ทราบถึงการบริการของตน (Service Registration) เมื่อผู้ใช้ต้องการที่จะใช้การบริการใด UA จะสอบถามการบริการดังกล่าวไปยัง DA (Service Request) หลังจากที่ได้รับแอดเดรสและลักษณะของการบริการที่ต้องการแล้วผู้ใช้จึงสามารถใช้ประโยชน์จากการบริการนั้นได้

ก่อนที่ไคลเอนต์ (UA or SA) จะสามารถติดต่อกับ DA นั้น จะต้องทำการค้นหา DA เสียก่อน ซึ่งมีด้วยกัน 3 รูปแบบ คือ Static, Active and Passive

- Static : SLP เอเจนต์ได้รับแอดเดรสของ DA ผ่านทาง DHCP (Dynamic Host Configuration Protocol) DHCP เซิร์ฟเวอร์จะคอยแจกจ่ายแอดเดรสของ DAs ไปยังโฮสต์ที่ต้องการมัน

- Active : UAs และ SAs ส่งคำร้องขอการบริการไปยัง SLP multicast group address DA ที่คอยฟังอยู่บนแอดเดรสนี้จะได้รับคำร้องขอการบริการและการตอบรับโดยตรง (ผ่าน unicast) จากเอเจนต์ที่ทำการร้องขอ
- Passive : DAs จะคอยส่งการประกาศแบบ multicast สำหรับการบริการของตนอยู่เป็นระยะๆ UAs และ SAs จะเรียนรู้แอดเดรสของ DA เมื่อได้รับมาและสามารถที่จะติดต่อกับ DA ด้วยตนเองผ่าน unicast



รูปที่ 7-2 การค้นหาการบริการที่ไม่มี DA

สำหรับตัว DA นั้นอาจจะไม่จำเป็นต้องมีก็ได้ โดยทั่วไปแล้วมีไว้สำหรับใช้ในเครือข่ายขนาดใหญ่ซึ่งมีการบริการอยู่เป็นจำนวนมาก เนื่องจากมันสามารถที่จะจำแนกชนิดของการบริการออกเป็นหลายๆกลุ่มได้ สำหรับเครือข่ายขนาดเล็ก (เช่น ในบ้านหรือในรถ) จะมีประสิทธิภาพมากกว่าถ้าใช้ SLP แบบที่ไม่มี DA SLP ในลักษณะนี้จะมีการทำงานอยู่ 2 รูปแบบขึ้นอยู่กับมี DA หรือไม่ ในกรณีที่มี DA มันจะทำหน้าที่รวบรวมข้อมูลข่าวสารทั้งหมดของการบริการที่ถูกประกาศโดย SA UAs จะส่งคำร้องขอการบริการไปยัง DA และรับข้อมูลข่าวสารการบริการที่ต้องการ สำหรับในกรณีที่ไม่มี SA UAs จะคอยส่งคำร้องขอการบริการไปยัง SLP multicast address SAs ทุกตัวจะคอยฟังการร้องขอแบบมัลติแคสเหล่านี้ เมื่อมันต้องการที่จะประกาศคำร้องขอการบริการมันจะส่งการตอบกลับแบบยูนิแคสไปยัง UA นอกจากนี้ SAs จะมัลติแคสการประกาศการมีอยู่ของมันอยู่เป็นระยะๆ ดังนั้น UAs จึงสามารถเรียนรู้เกี่ยวกับการมาของการบริการใหม่ๆได้

การบริการจะถูกประกาศโดยใช้ URL ของการบริการและเทมเพลตของการบริการ URL ของการบริการ ประกอบด้วยไอพีแอดเดรสของการบริการ, หมายเลขพอร์ตและเส้นทาง ส่วนเทมเพลตของการบริการจะระบุแอททริบิวที่บอกลักษณะของการบริการและค่าพื้นฐานของการบริการนั้น

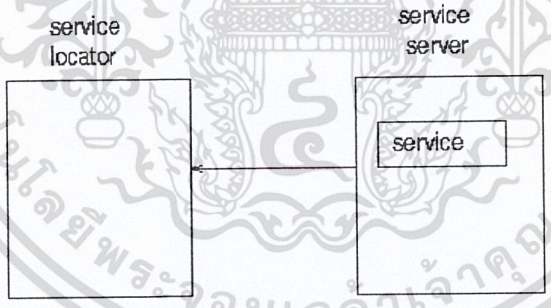
Jini

จินีถูกพัฒนาโดยซันไมโครซิสเทมส์ จินีจะบอกรายละเอียดเกี่ยวกับวิธีที่อุปกรณ์จะเชื่อมต่อกับอุปกรณ์อื่นๆและวิธีที่อุปกรณ์เหล่านี้จะให้การบริการกับอุปกรณ์อื่นๆที่อยู่ในเครือข่าย จินีประกอบด้วยสถาปัตยกรรมของเครือข่ายและรูปแบบการโปรแกรมมิ่ง

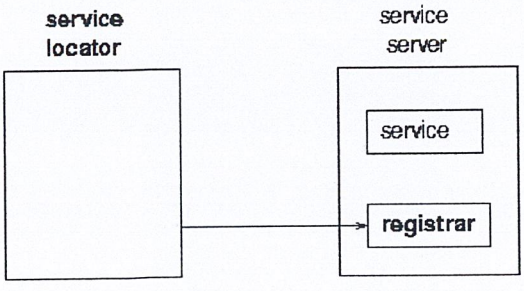
อุปกรณ์จินีแต่ละตัวจะมี JVM (Java Virtual Machine) รันอยู่บนตัวมัน อุปกรณ์ต่างๆและโปรแกรมประยุกต์จะรีจิสเตอร์เครือข่ายจินีโดยใช้โพรเซสที่เรียกว่า การค้นหา (Discovery) และการเชื่อมต่อ (Join) ในการเชื่อมต่อเข้ากับเครือข่ายจินี อุปกรณ์หรือโปรแกรมประยุกต์จะวางตัวเองไปอยู่ในตารางที่ใช้ในการค้นหา (Lookup Table) บนเซิร์ฟเวอร์ของการค้นหา (Lookup Server) ที่ซึ่งเป็นฐานข้อมูลสำหรับการบริการทุกอย่างที่มีอยู่ในเครือข่าย ตารางที่ใช้ในการค้นหาในจินีนี้จะโปรแกรมโดยใช้ภาษาจาวา นั่นหมายความว่าการบริการจะสามารถอัพโหลดตัวขับอุปกรณ์ (Device Driver), การเชื่อมต่อ (Interface) และโปรแกรมอื่นๆที่ช่วยให้ผู้ใช้เข้ามาใช้การบริการ เมื่อผู้ใช้ต้องการที่จะใช้การบริการโค้ดที่เป็นออบเจ็กต์จะถูกโหลดจากตารางที่ใช้ในการค้นหาไปยัง JVM ของผู้ใช้ ในขณะที่คำร้องขอการบริการใน SLP จะส่งค่า URL ของการบริการกลับมา โค้ดที่เป็นออบเจ็กต์นี้สามารถติดต่อโดยตรงผ่านการเชื่อมต่อที่ผู้ใช้รู้จัก ซึ่งโคดที่สามารถเคลื่อนย้ายนี้ช่วยลดความจำเป็นในการติดตั้งไดรเวอร์ก่อนได้



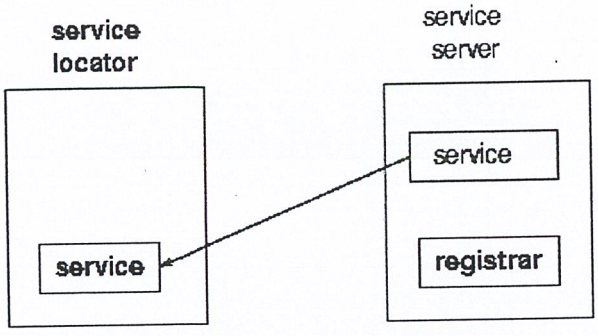
รูปที่ 7-3 องค์ประกอบของเครือข่ายจินี



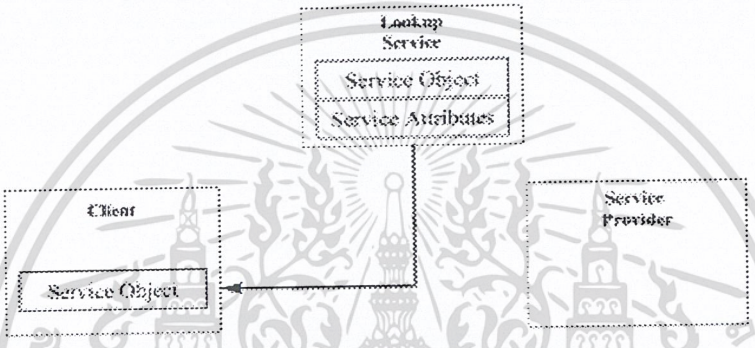
รูปที่ 7-4 การบริการทำการค้นหาตารางในการค้นหาการบริการ



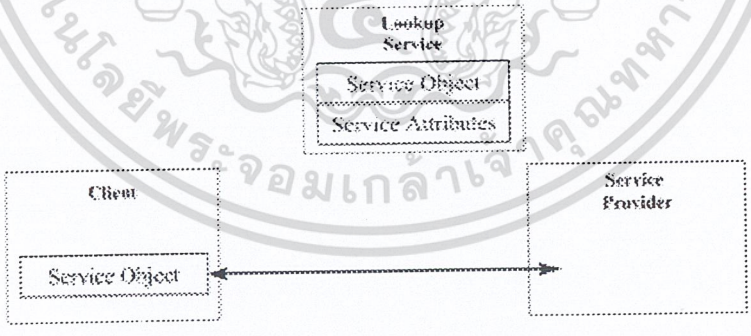
รูปที่ 7-5 ตารางในการค้นหาการบริการตอบรับด้วยออบเจ็กต์เซอร์วิส



รูปที่ 7-6 การบริการส่งออบเจ็กต์ของการบริการไปลงทะเบียนที่ลูกอัพรีจิสทรา



รูปที่ 7-7 ไคลเอนต์ได้รับออบเจ็กต์ของการบริการ

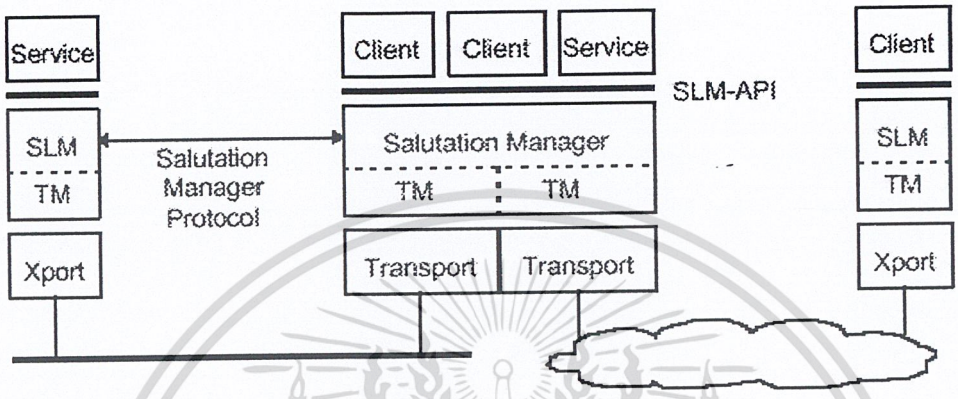


รูปที่ 7-8 ไคลเอนต์ติดต่อกับการบริการผ่านออบเจ็กต์ของการบริการ

เนื่องมาจากลักษณะที่ยืดหยุ่นของเครือข่ายแบบแอดฮอคทำให้เกิดหลักการของลีสสิ่ง (leasing) ทุกๆครั้งที่อุปกรณ์ต่อเข้ามาในเครือข่ายและมีการให้การบริการใช้นั้น มันจะรีจิสเตอร์ตัวเองกับเครือข่ายในระยะเวลาหนึ่งที่แน่นอน ซึ่งวิธีแบบนี้เรียกว่าลีส (lease)

Salutation

Salutation ถูกพัฒนาโดยกลุ่มอุตสาหกรรมเปิด Salutation Consortium สถาปัตยกรรม Salutation นี้ประกอบด้วย Salutation Managers (SLMs) ซึ่งมีหน้าที่เป็น Service Brokers การบริการจะรีจิสเตอร์ตัวเองเข้ากับ SLM ไคลเอนท์จะสอบถามไปยัง SLM เมื่อต้องการใช้การบริการ หลังจากที่ค้นพบการบริการที่ต้องการแล้ว ไคลเอนท์จะส่งคำร้องไปเพื่อใช้บริการผ่าน SLM



รูปที่ 7-9 โมเดลของ Salutation Manager

ในการที่จะทำหน้าที่เป็น Service Broker นั้น ตัว Salutation Manager จะต้องทำหน้าที่พื้นฐาน 4 อย่าง คือ

- Service Registry
- Service Discovery
- Service Availability
- Service Session Management

Universal Plug and Play (UPnP)

Universal Plug and Play (UPnP) ถูกพัฒนาโดยกลุ่มอุตสาหกรรม ซึ่งมีไมโครซอฟท์เป็นผู้ก่อตั้งและเป็นแกนนำสำคัญ อาจกล่าวได้ว่า UpnP เป็นส่วนเพิ่มเติมจาก Microsoft's Plug and Play ในกรณีที่อุปกรณ์จะติดต่อสื่อสารผ่านเครือข่าย TCP/IP เหมาะที่จะใช้ในสำนักงานขนาดเล็กหรือเครือข่ายคอมพิวเตอร์ภายในบ้าน ซึ่งเป็นลักษณะการเชื่อมต่อแบบ peer-to-peer รองรับการปรับแต่งอุปกรณ์อย่างอัตโนมัติ, การค้นหาการบริการ และการควบคุมการบริการต่างๆ ใน UpnP จะไม่มีส่วนกลางที่ทำหน้าที่เป็น Service register เหมือน SLP หรือ Jini Upnp ใช้ SSDP (Simple Service Discovery Protocol) ในการค้นหาการบริการ SSDP ใช้ HTTP บน UDP ดังนั้นมันจึงออกแบบมาเพื่อใช้บนเครือข่ายที่เป็น IP

Feature	SLP	Jini	Salutation	UPnP
Developer	IETF	Sun Microsystems	Salutation Consortium	Microsoft
License	Open Source	Open license, fee for commercial use	Open Source	Open (only for members)
Version	2	1.0	2.1	0.91
Network Transport	TCP/IP	Independent	Independent	TCP/IP
Programming Lang	Independent	Java	Independent	Independent
OS and Platform	dependant	Independent	dependent	dependent
Code mobility	No	Yes (Java RMI)	No	No
Srv Attrib Search	Yes	Yes	Yes	No
Leasing concept	Yes	Yes	No	Yes
Security	IP dependant	Java based	Authentication	IP dependant

ตารางที่ 7-1 เปรียบเทียบโพรโตคอลค้นหาการบริการต่างๆ

7.3 สรุปสิ่งที่ได้จากการพัฒนาโครงงาน

จากการที่ได้ศึกษา ออกแบบ และพัฒนาตัวจัดการอุปกรณ์สำหรับเครือข่ายแบบแอดฮอคผ่านบลูทูธ ทำให้ทราบถึงแนวทางในการเจริญเติบโตอย่างรวดเร็วของเทคโนโลยีไร้สายที่เข้ามามีบทบาทในยุคปัจจุบันอย่างมาก โดยในแต่ละเทคโนโลยีก็มีคุณสมบัติและลักษณะเด่นที่แตกต่างกันและเหมาะสำหรับการใช้งานที่แตกต่างกันไป แต่จะมีอัตราการเจริญเติบโตและการแข่งขันเพื่อการแย่งชิงส่วนแบ่งตลาดกันอย่างสูง จากที่บลูทูธนั้นมีลักษณะเด่นในด้านการทำงานเป็นระบบเครือข่ายแบบแอดฮอคโดยเฉพาะ ดังนั้นทางกลุ่มจึงได้พัฒนาความสามารถในการจัดการต่างๆลงไปในตัวจัดการอุปกรณ์เพื่อให้อุปกรณ์บลูทูธสามารถทำงานในระบบเครือข่ายแบบแอดฮอคได้ โดยพัฒนาในเรื่องของการค้นหาอุปกรณ์และการค้นหาการบริการที่มีอยู่ในเครือข่าย

กล่าวโดยสรุป สำหรับโครงงานที่ได้จัดทำขึ้นนี้ สามารถที่จะทำงานได้ตามความต้องการขั้นพื้นฐานของการจัดการเกี่ยบกับเครือข่ายแบบแอดฮอค และสามารถที่จะทำการปรับแต่งค่าต่างๆให้กับโมดูลของบลูทูธได้ ดังนั้นสำหรับโครงงานที่ได้พัฒนาขึ้นนี้นั้น ทำให้ได้เรียนรู้เกี่ยวกับเทคโนโลยีไร้สายต่างๆ จากนั้นก็ต้องทำการศึกษาหลักการทำงานของโพรโตคอลแสตกของบลูทูธ รวมทั้งค้นหาโพรโตคอลแสตกที่อนุญาตให้สามารถนำไปพัฒนาต่อได้ นำมาศึกษาส่วนการทำงานของโปรแกรมและฟังก์ชันต่างๆที่มีอยู่ภายใน เพื่อที่จะนำไปแก้ไขปรับปรุงและเขียนฟังก์ชั่นต่างๆเพิ่มเติมให้สามารถที่จะทำงานเชื่อมต่อกับตัวจัดการอุปกรณ์และสามารถที่จะปรับแต่งค่าต่างๆได้ตามต้องการ โดยทำการพัฒนาภาษาซีบนลีนุกซ์ เนื่องจากโพรโตคอลแสตกที่ได้มานั้นทำงานบนลีนุกซ์ และเพื่อให้ง่ายต่อการเข้าใจและนำไปใช้งานของผู้ใช้ จึงทำการเขียนส่วนการติดต่อกับผู้ใช้โดยการนำ FLTK เข้ามาช่วย พร้อมทั้งเขียนการบริการเพิ่มเติมเพื่อให้สามารถทดสอบการทำงานได้อย่างชัดเจนยิ่งขึ้น นอกจากนี้ในส่วนของการพัฒนาต่อนั้นได้กำหนดแนวทางไว้สองแนวทาง คือ การพัฒนาในส่วนของตัวจัดการอุปกรณ์และการพัฒนาในส่วนของการค้นหาการบริการ ซึ่งเป็นส่วนที่ทำให้อุปกรณ์บลูทูธสามารถทำงานในระบบเครือข่ายแบบแอดฮอคได้ดียิ่งขึ้น

ภาคผนวก



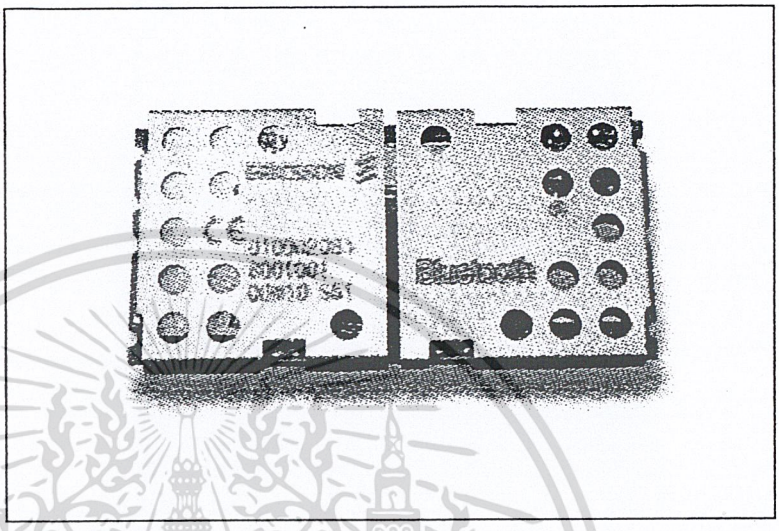
ภาคผนวก ก.

Data Sheet

ROK 101 007
Bluetooth Module

Key Features

- Pre-qualified Bluetooth 1.0B Module
- RF output power class 2
- FCC and ETSI approved
- 460 kb/s max data rate over UART
- Multiple interface for different applications
  - -UART for data
  - -PCM for voice
  - -USB for voice and data
- I²C interface
- Internal crystal oscillator
- HCI firmware included
- Multi Point Operation
- Built-in shielding



Suggested Applications

- Computers and peripherals
- Handheld devices and accessories
- Access points

Description

ROK 101 007 is a short-range module for implementing Bluetooth functionality into various electronic devices. The module consists of three major parts; a baseband controller, a flash memory, and a radio that operates in the globally available 2.4–2.5 GHz free ISM band.

Both data and voice transmission is supported by the module. Communication between the module and the host controller is carried out using a high-speed USB interface compliant with USB Specifications 1.1 or an UART/PCM interface. When using the USB interface, the module appears as a USB slave device and therefore requires no PC resources.

ROK 101 007, which is compliant with Bluetooth version 1.0B, is a Class 2 Bluetooth Module (0 dBm) and is type-approved. The module supports all Bluetooth profiles.

ERICSSON

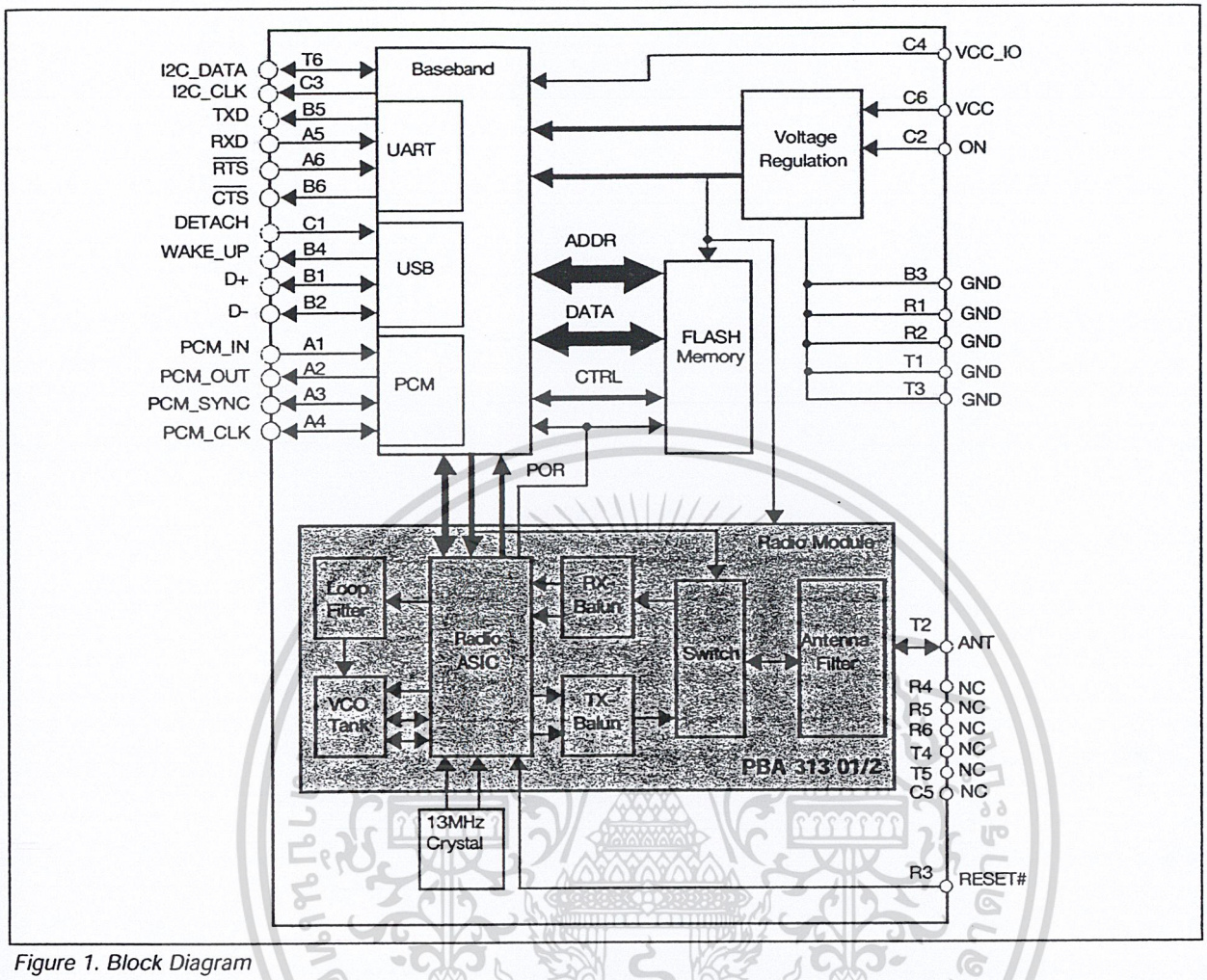


Figure 1. Block Diagram

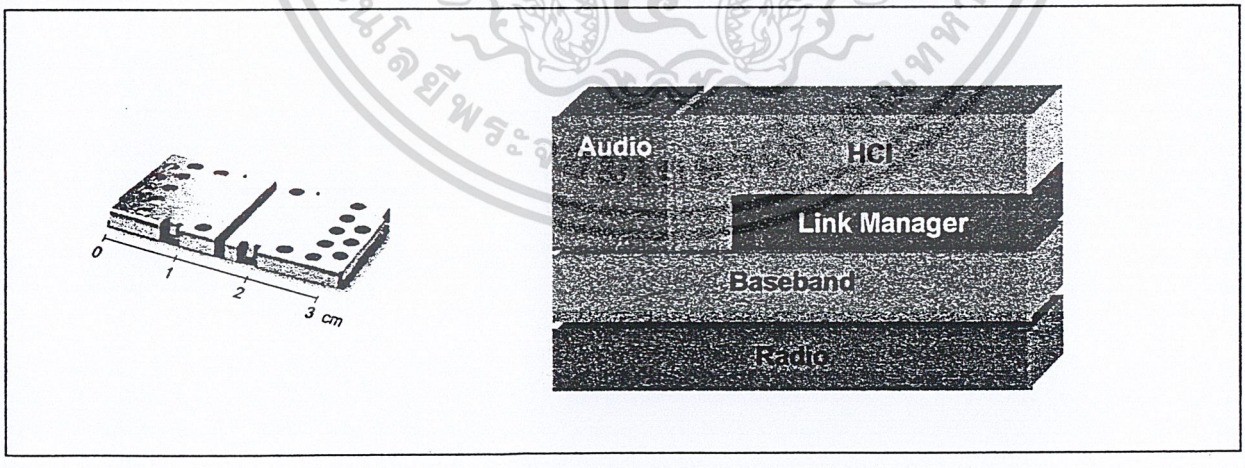


Figure 2. Actual size of the Ericsson Bluetooth Module, and also showing the HW and FW stack.

Absolute Maximum Ratings

Parameter	Symbol	Min	Typ	Max	Unit
Temperature					
Storage temperature	T_{Stg}	-30		+85	°C
Operating temperature	T_{Amb}	0		+75	°C
Power Supply					
V_{CC}	V_{CC}	-0.3		+5.25	V
V_{CC_IO}	V_{CC_IO}	-0.8		+3.6	V
Digital Inputs					
Input low voltage	V_{IL}	-0.5			V
Input high voltage	V_{IH}			V_{CC_IO} +0.3	V

Recommended Operating Conditions

Temperature				
Ambient temperature, Test	T_{amb}		+23	°C
Power Supply				
Positive Supply Voltage	V_{CC}		+3.3	V
I/O Ports Supply Voltage	V_{CC_IO}		+3.3	V

Electrical Characteristics

DC Specifications

Unless otherwise noted, the specification applies for T_{Amb} = 0 to +75°C, 3.175 < V_{CC} < 5.25V

Parameter	Condition	Symbol	Min	Typ	Max	Unit
Power Supply						
Supply Voltage		V_{CC}	3.175	3.3	5.25	V
I/O Ports Supply Voltage	See note 10	V_{CC_IO}	2.7	3.3	3.6	V
Digital Inputs						
Logical Input High	Except ON signal	V_{IH1}	0.7 x V_{CC_IO}		V_{CC_IO}	V
Logical Input Low	Except ON signal	V_{IL2}	0		0.3 x V_{CC_IO}	V
Logical Input High	ON signal only	V_{IH2}	2.0		V_{CC}	V
Logical Input Low	ON signal only	V_{IL2}	0		0.4	V
Digital Outputs						
Logical Output High		V_{OH}	0.9 x V_{CC_IO}		V_{CC_IO}	V
Logical Output Low		V_{OL}	0		0.1 x V_{CC_IO}	V

5Parameter	Condition	Symbol	Min	Typ	Max	Unit
Average Current Consumption	Average $I_{CC} + I_{CC\ IO}$					
Standby		I_{STA}		5.95		mA
Shutdown - SW		I_{SSW}		2.35		mA
Shutdown – HW	See note 1	I_{SHW}		1		µA
Page Scan Mode R0	Page Scan Enable Page scan window: continuous Page scan interval : n/a (0s)	I_{PSM1}		50		mA
Page Scan Mode R1	Page Scan Enable Page scan window: 11.25ms Page scan interval : 1.28s	I_{PSM2}		6.35		mA
Page Scan Mode R2	Page Scan Enable Page scan window: 11.25ms Page scan interval : 2.56s	I_{PSM3}		6.15		mA
Inquiry Scan with Page Scan Mode R0.	Inquiry Scan Enable Page Scan Window : 2.56s –11.25ms Page Scan Interval : 2.56s Inquiry Scan Window : 11.25ms Inquiry Scan Interval : 2.56s	I_{ISM1}		50		mA
Inquiry Scan with Page Scan Mode R1.	Inquiry Scan Enable Page Scan Window : 11.25ms Page Scan Interval : 1.28s Inquiry Scan Window : 11.25ms Inquiry Scan Interval : 2.56s	I_{ISM2}		6.55		mA
inquiry Scan with Page Scan Mode R2.	Inquiry Scan Enable Page Scan Window : 11.25ms Page Scan Interval : 2.56s Inquiry Scan Window : 11.25ms Inquiry Scan Interval : 2.56s	i_{ISM3}		6.35		mA
Connect State	Established connection with data transfer	I_{CS}		26		mA
$V_{CC\ IO}$ Current	Only $I_{CC\ IO}$, $V_{CC\ IO=TBD}$	$I_{CC\ IO}$		TBD		mA

RF Specifications

Parameter	Condition	Symbol	Min	Typ	Max	Unit
General						
Frequency Range			2.402		2.480	GHz
Double Sided IF Bandwidth				1		MHz
Antenna load				50		Ω
VSWR	RX mode			3:1		
VSWR	TX mode, see note 2			3:1		
Receive Performance						
Sensitivity level	P_{IN}: -70dBm, 75kHz offset (max),				0.1%	BER
Max input level	P_{IN}: -20dBm, 75kHz offset (max),				0.1%	BER
$C/I_{co\text{-}channel}$	C: -60 dBm				14	dB
C/I_{1MHz}	C: -60 dBm				+4	dB
C/I_{2MHz}	C: -60 dBm				-30	dB
$C/I_{\geq 3MHz}$	C: -67 dBm				-40	dB
Blocking, $C/I_{DC\text{-}5GHz}$	See figure 6					
Out-of-band blocking	30-1910 MHz		+4			dBm
	1910-2000 MHz		-10			dBm
	2000-2399 MHz		-27			dBm
	2484-3000 MHz		-27			dBm
	3.00-12.75 GHz		-10			dBm
Spurious Emissions	30 MHz to 1 GHz				-57	dBm
Spurious Emissions	1 GHz to 12.75 GHz				-47	dBm
Transmitter Performance						
Frequency deviation	see notes 3,4 and figure 3	f_{Mod}	140		175	kHz
Initial frequency error	see note 5		-48		+48	kHz
TX power			-2	1.5	4	dBm
TX carrier drift in 1 slot (366 µs)		f_{Drift1}	-25		25	kHz
TX carrier drift in 3 slots (1598 µs)		f_{Drift2}	-40		40	kHz
TX carrier drift in 5 slots (2862 µs)	see figure 4	f_{Drift3}	-40		40	kHz
20 dB bandwidth	Measured with RBW: 10 kHz and peak detector				1.000	MHz
Spurious Emissions	30 MHz – 1GHz				-36	dBm
Spurious Emissions	1 GHz – 12.75 GHz				-30	dBm
Spurious Emissions	1.8 Hz – 1.9 GHz				-47	dBm
Spurious Emissions	5.15 GHz – 5.3 GHz				-47	dBm
Timing performance						
LPO_CLK frequency	Trimmed, see note 6	$f_{LPO\text{—}CLK}$		3,2		kHz
Tolerance of LPO_CLK	see note 6		-250		+250	ppm
System clock frequency		$f_{SYS\text{—}CLK}$		13.0000		MHz
Tolerance of system clock	see note 5		-20		+20	ppm
Channel switching time	see figure 5			150		µs
Received Signal Strength Indicator						
RSSI	Input power = -40 dBm		15		25	n/a
RSSI	Input power = -60 dBm		3		11	n/a

RF Specifications continued...

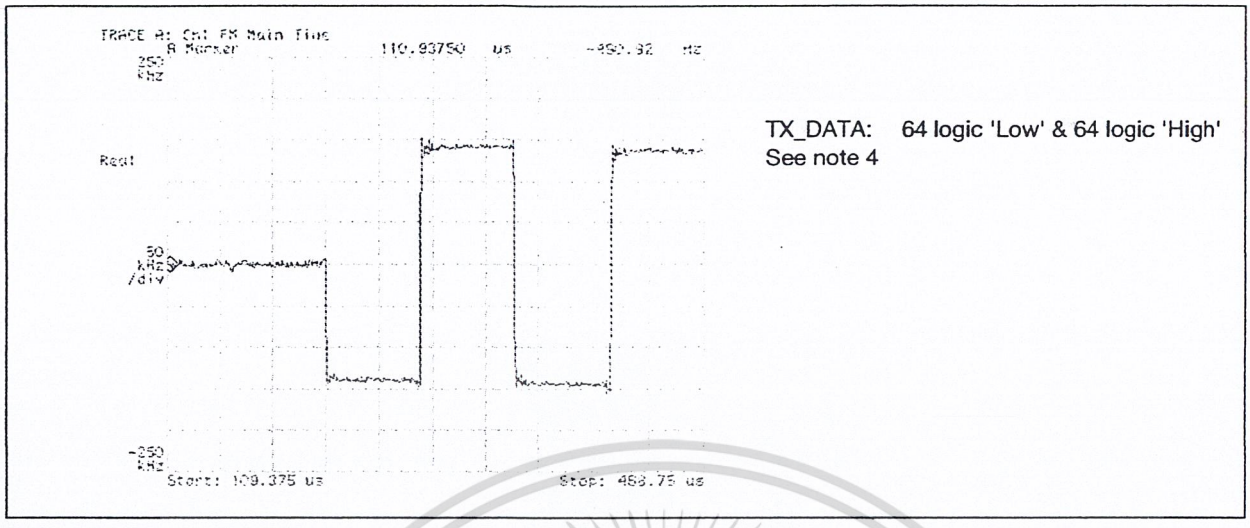


Figure 3. Frequency Deviation

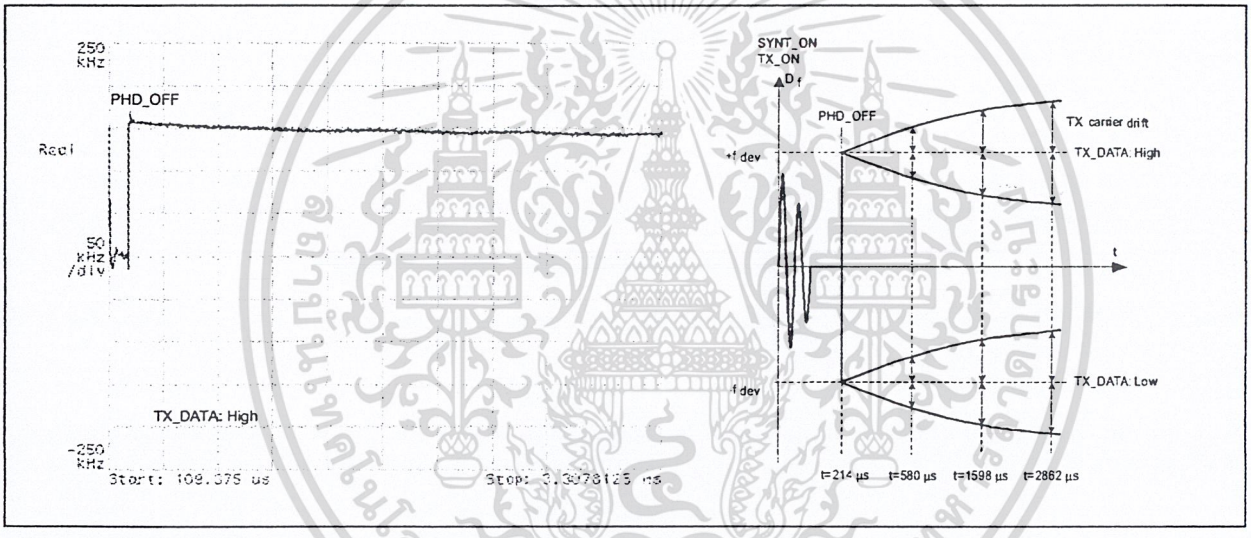


Figure 4. Frequency drift.

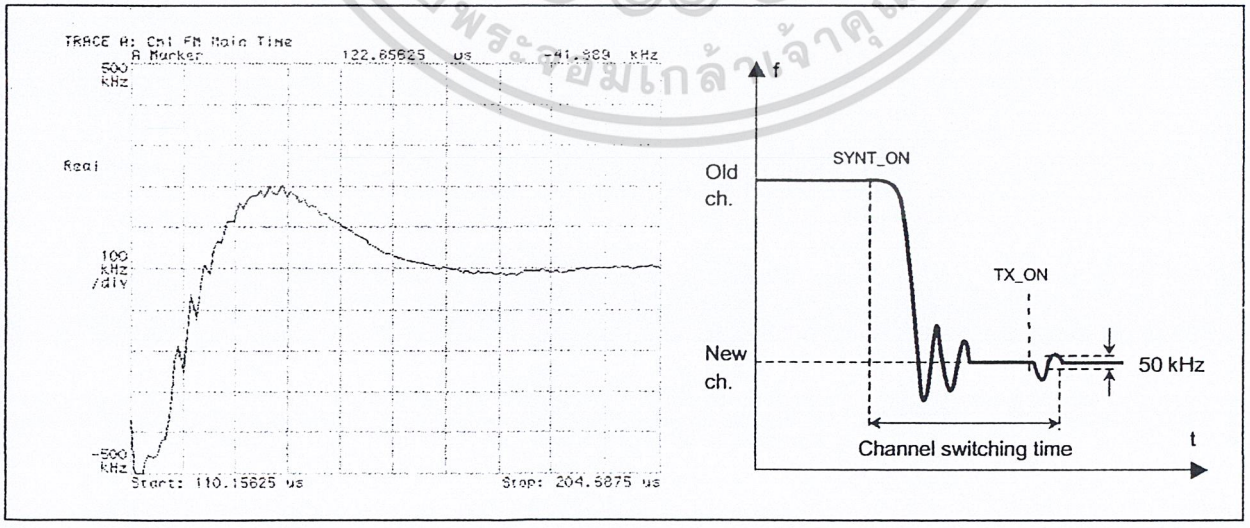


Figure 5. Channel switching time

RF Specifications continued...

C/I Blocking
The blocking characteristics can be basically split into two regions: In-band and Out-of-band. Blocking is performed both on the chip and on the module level.

- **In-band**
- Filtering on chip
- C/I @ 2MHz: -30 dB -> 0.1%BER
- C/I ≥ 3MHz: -40 dB -> 0.1%BER

- **Out-of-band**
- Antenna filter, DC to 1.9 GHz and 3:rd harmonic.
- Switch, low freq. and 2:nd harmonic.
- RX-balun, low freq. and 2:nd harmonic.
- On-chip IF filter

Figure 6 shows the combination blocking effect of the antenna switch, antenna filter and RX balun. In addition to the blocking characteristics shown in figure 6, there is antenna isolation and filtering on the chip.
Marker 1 shows the region where the Bluetooth band is located. Markers 2 - 4 show the blocking at the telecom frequency bands.
An example of the total blocking characteristics can be seen in figure 7.

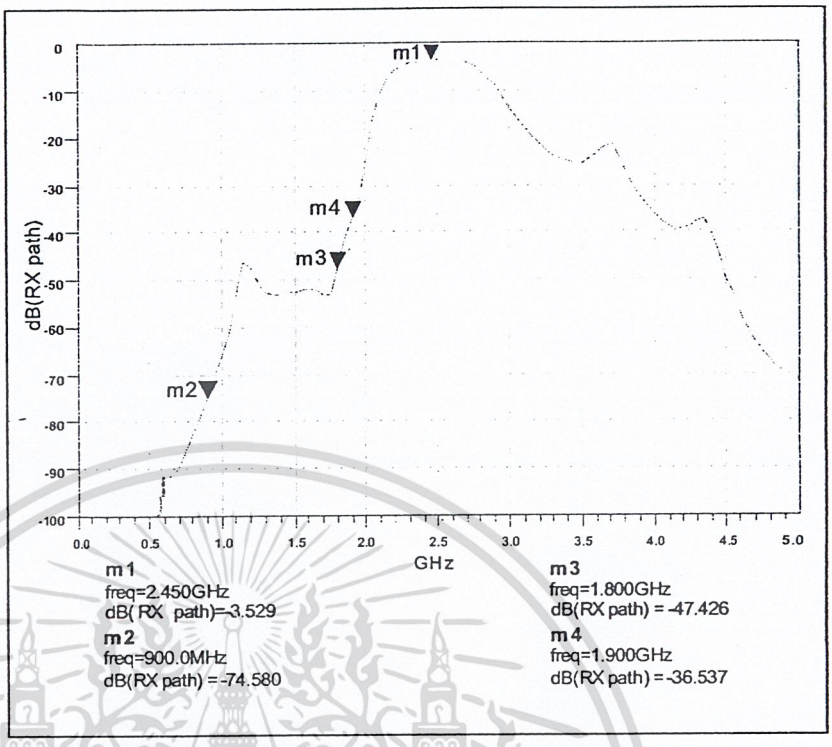


Figure 6. Typical blocking characteristics excluding antenna isolation and on chip filtering.

Example 1		Example 2	
Interference of +33 dBm at 2015 MHz.		Interference of +33 dBm at 1910 MHz.	
Antenna isolation	15 dB	Antenna isolation	25 dB
Antenna filter, Antenna switch, RX-balun	27 dB	Antenna filter, Antenna switch, RX-balun	36 dB
Interference level before IF filter +33-15-27=	-9 dBm	Interference level before IF filter +33-25-36=	-28 dBm
0.1% BER requires a C/I of more than -40 dB at the IF filter.			
0.1% BER carrier level -40 + (-9)=	-49 dBm	0.1% BER carrier level -40 + (-28)=	-68 dBm

Figure 7. Blocking examples.

Pin Description				
Pin	**Pin Name**	**Type**	**Direction**	**Description**
A1	PCM_IN	CMOS	In	PCM data, see notes 7,9
A2	PCM_OUT	CMOS	Out	PCM data, see notes 7,9
A3	PCM_SYNC	CMOS	In/Out	Sets the PCM data sampling rate, see notes 7,9
A4	PCM_CLK	CMOS	In/Out	PCM clock that sets the PCM data rate, see notes 7,9
A5	RXD	CMOS	Input	RX data to the UART, see note 9
A6	RTS	CMOS	Input	Flow control signal, Request To Send data from UART, see notes 7,9
B1	D+	CMOS	In/Out	USB data pin, see notes 9,10
B2	D-	CMOS	In/Out	USB data pin, see notes 9,10
B3	GND	Power	Power	Signal ground
B4	WAKE_UP	CMOS	Output	Indicates that the module wants to be attached to the USB, Active High. See notes 9,10
B5	TXD	CMOS	Output	TX data from the UART, see note 9
B6	CTS	CMOS	Output	Flow control signal, Clear To Send data from UART, see note 9
C1	DETACH	CMOS	Input	Indicates that the USB host wants to detach the module, Active High. See notes 7,9
C2	ON	Power	Input	When tied to V_{CC}, the module is enabled.
C3	I2C_CLK	CMOS	Output	I2C clock signal, see note 9
C4	VCC_IO	Power	Power	External supply rail to the Input / Output ports
C5	NC	-	-	Do not connect
C6	VCC	Power	Power	Supply Voltage
R1	GND	Power	Power	Signal ground
R2	GND	Power	Power	Signal ground
R3	RESET#	CMOS	Input	Active low reset, see notes 8,9
R4	NC	-	-	Do not connect
R5	NC	-	-	Do not connect
R6	NC	-	-	Do not connect
T1	GND	Power	Power	Signal Ground
T2	ANT	RF	In/Out	50Ω Antenna connection
T3	GND	Power	Power	Signal Ground
T4	NC	Power	Power	Test point, internal voltage regulator - Do not connect
T5	NC	-	-	Do not connect
T6	I2C_DATA	CMOS	In/Out	I2C data signal, see note 9

Notes

1. Current consumption is based upon when the module is when 'ON' is low and 'VCC_IO' is grounded.
2. During the TX mode, the VSWR specification states the limits that are acceptable before any other RF parameters are strongly effected, i.e. frequency deviation and initial frequency error.
3. Frequency deviation measurements are now recorded differentially, (f Mod1 - f Mod0) / 2.
4. Provided that the TX INV register (bit 0) has been set in the enable register at startup.
5. Tolerance for the system clock takes into account both the complete temperature range and aging of the crystal.
6. LPO_CLK frequency is pre-trimmed within a tolerance of ±250ppm.
7. 100kΩ pull-up resistors to V_{CC_IO} are used on the module. PCM signals direction is programmable
8. RESET# signal must be fed from an open drain output.
9. CMOS buffers are low voltage TTL compatible signals.
10. To be compliant with the USB specification, VCC_IO ≥ 3.11V

Mechanical Specification

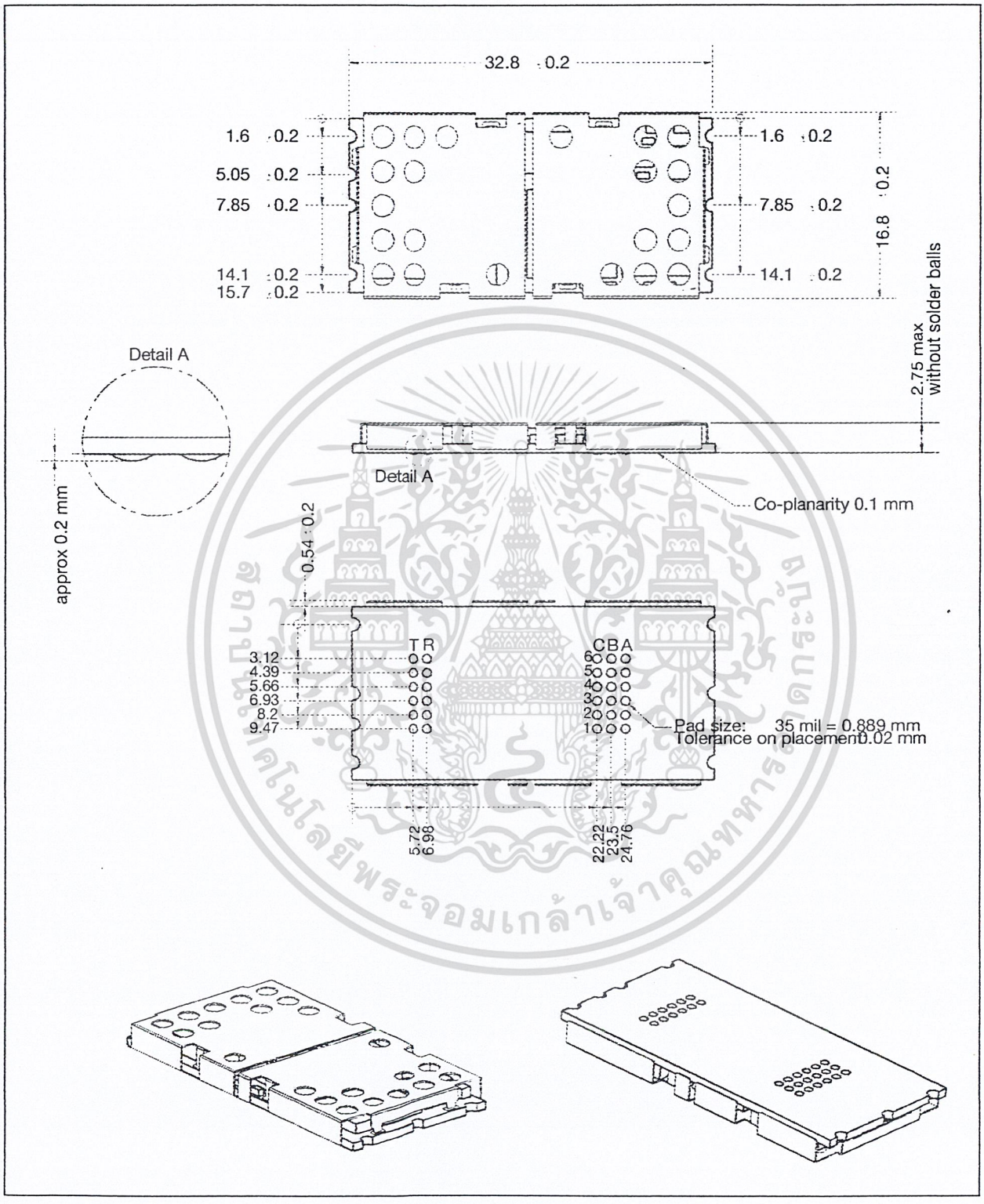


Figure 8. Mechanical dimensions.

Application Block Schematics

USB Application

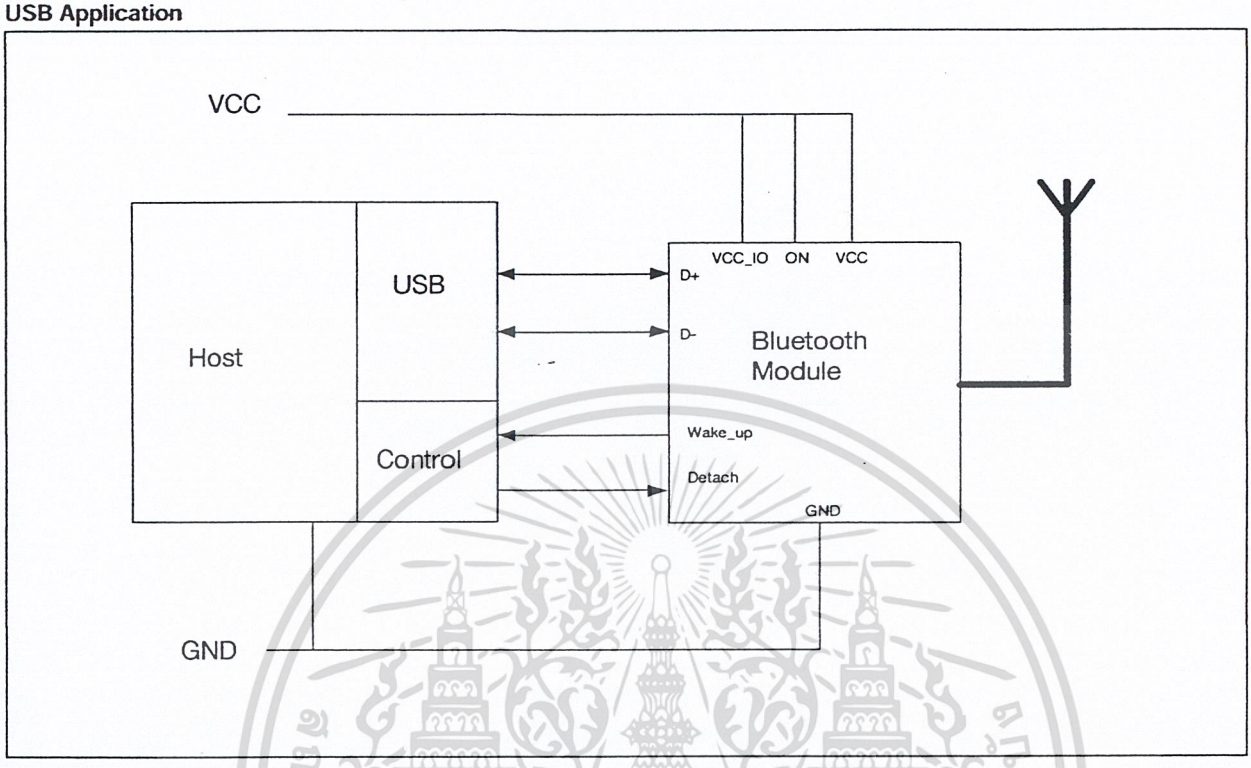


Figure 9. A typical USB configuration.

UART and PCM Application

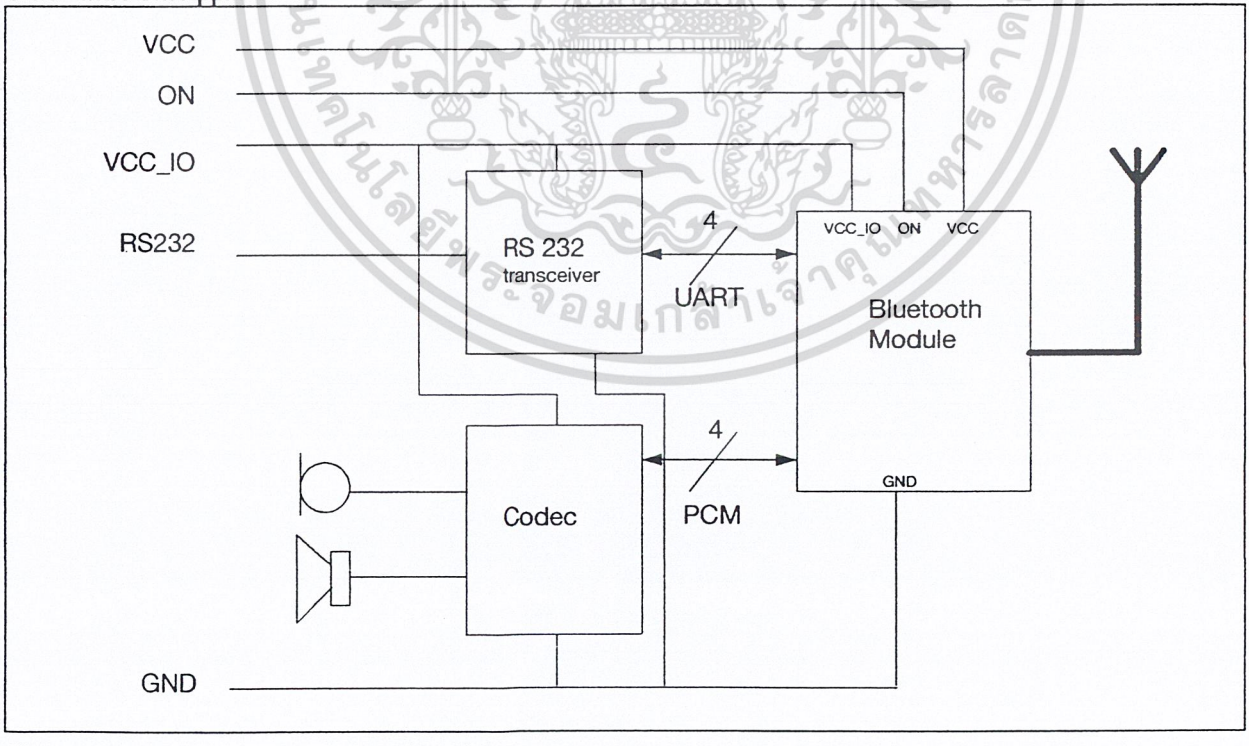


Figure 10. A typical UART or PCM configuration.

Functional Description

The ROK 101 007 is a complete Bluetooth module that has been specified and designed according to the Bluetooth System v1.0B. Its implementation is based on a high-performance integrated radio transceiver (PBA 313 01 /2) working with a baseband controller, a flash memory and surrounding secondary components features low energy consumption for use in battery operated devices.

Block Diagram

ROK 101 007 has five major operational blocks. Figure 11 illustrates the interaction of the various blocks. The functionality of each block is as follows:

1. Radio functionality is achieved by using the Bluetooth Radio, PBA 313 01/ 2. Six operational blocks are shown for the radio section and their operation is as follows:

1a) VCO-tank is a part of the phase locked loop. The modulation is performed directly on the VCO. To ensure high performance the VCO-tank is laser trimmed.

1b) Loop filter, filters the tuning voltage of the VCO-tank.

1c) RX-balun handles transformation from unbalanced to balanced transmission.

1d) TX-balun handles biasing of the output amplifier stage and transformation from balanced to unbalanced transmission.

1e) Antenna switch directs the power either from the antenna filter to the receive ports or from the ASIC output ports to the antenna filter.

1f) Antenna filter band-pass filters the radio signal.

2. The baseband controller is an ARM7-Thumb based chip that controls the operation of the radio transceiver via one of the interface methods; USB or UART. Additionally, the baseband controller has a PCM Voice interface and I²C interface.
3. A Flash memory is used together with the baseband controller. Please, refer also to the Firmware section.
4. The power management block regulates and filters the supply voltage. V_{CC} is typically 3.3V and two regulated voltages are produced, 2.8V and 2.2V_{NOM}.
5. An internal clock is mounted on the module. The clock frequency is 13MHz and is generated from a crystal oscillator that guarantees a timing accuracy within ± 20ppm.

Bluetooth Module stack

The Host Controller Interface (HCI) handles the communication by the transport layer through the UART or USB interface with the host. The Baseband and radio provides a secure and reliable radio link for higher layers.
The following sections describe the Bluetooth module stack in more detail. It is implemented in accordance with and complies with the Specification of the Bluetooth System v1.0B.

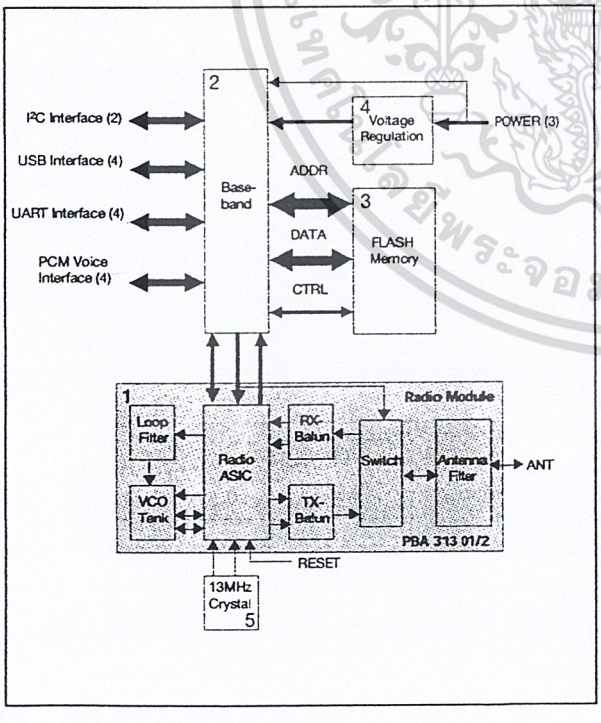


Figure 11. Simplified Block Diagram

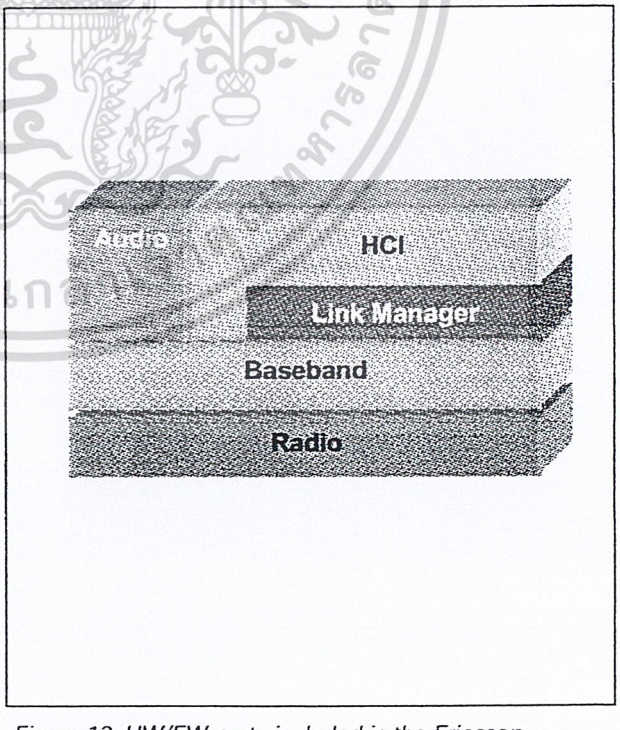


Figure 12. HW/FW parts included in the Ericsson Buetooth module.

Bluetooth Radio Interface
The Bluetooth module is a class 2 device with 4dBm maximum output power with no power control needed. Nominal range of the module with a typical antenna is up to a range of 10 m (at 0 dBm). It is compliant with FCC and ETSI regulations in the ISM band.

Baseband
Bluetooth uses an ad-hoc net structure with a maximum of eight active units in a single piconet. By default the first unit setting up a connection is the master of the point to point link. The master transmits in the even timeslots and the slave transmits in the odd timeslots.

For full duplex transmission, a Time-Division Duplex (TDD) scheme is used. Packets are sent over the air in timeslots, with a nominal length of 625 µs. A packet can be extended to a maximum of 5 timeslots (DM5 and DH5 packets) and is then sent by using the same RF channel for the entire packet.

Two types of connections are provided - Asynchronous Connectionless Link (ACL) for data and the Synchronous Connection Oriented Link (SCO) for voice. Three 64 kb/s voice channels can be supported simultaneously. Furthermore, there are also packages used for link control purposes.

A variety of different packet types with error correction schemes and data rates can be used over the air interface. Also asymmetric communication is available for high speed communication in one direction.

The Baseband provides the link-setup and control routines for the layers above. Furthermore, the Baseband also provides Bluetooth security like encryption, authentication and key management.

Please refer to the Specification of the Bluetooth System v1.0B part B for in-depth information regarding the Baseband.

Firmware (FW)

The module includes firmware for the host controller interface, HCI, and the link manager, LM.
The FW resides in the Flash and is available in object code format.

Link Manager (LM)
The Link Manager in each Bluetooth module can communicate with another Link Manager by using the Link Manager Protocol (LMP) which is a peer to peer protocol.
The LMP messages have the highest priority and are used for link-setup, security, control and power saving modes. The receiving Link Manager filter-out the message and does not need to acknowledge the message to the transmitting LM due to the reliable link provided by the Baseband and radio.

LM to LM communication can take place without actions taken by the host. Discovery of features at other Bluetooth enabled devices nearby can be found and saved for later use by the host.

Please refer to the Specification of the Bluetooth System v1.0B part C for in-depth information regarding the LMP.

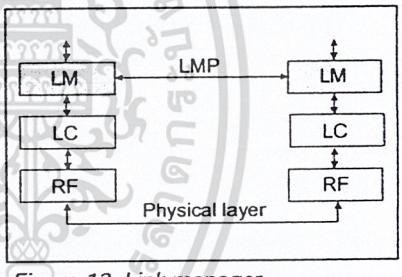


Figure 13. Link manager

Type	User Payload (bytes)	FEC	CRC	Symetric Max. rate	Asymetric Max.rate
ID	na	na	na	na	na
NULL	na	na	na	na	na
POLL	na	na	na	na	na
FHS	18	2/3	yes	na	na

Link control packets

Type	Payload Header (bytes)	User Payload (bytes)	FEC	CRC	Symetric Max. rate (kb/s)	Asymetric Max rate (kb/s)	
						Forward	Reverse
DM1	1	0-17	2/3	yes	108.8	108.8	108.8
DH1	1	0-27	no	yes	172.8	172.8	172.8
DM3	2	0-121	2/3	yes	258.1	387.2	54.4
DH3	2	0-183	no	yes	390.4	585.6	86.4
DM5	2	0-224	2/3	yes	286.7	477.8	36.3
DH5	2	0-339	no	yes	433.9	723.2	57.6
AUX1	1	0-29	no	no	185.6	185.6	185.6

ACL packets

Type	Payload header (bytes)	User Payload (bytes)	FEC	CRC	Symetric Max. rate (kb/s)
HV1	na	10	1/3	no	64.0
HV2	na	20	2/3	no	64.0
HV3	na	30	no	no	64.0
DV	1D	10+(0-9) D	2/3 D	Yes D	64.0+57.6 D

SCO packets

Table 1: Link Control Packets Table, ACL Packets Table, SCO packets

Host Control Interface (HCI)
The HCI provides a uniform command I/F to the Baseband and Link Manager and also to HW status registers.

There are three different types of HCI packets:

- HCI command packets – from host to Bluetooth module HCI.
- HCI event packets – from Bluetooth module HCI to host.
- HCI data packets – going both ways.

It is not necessary to make use of all different commands and events for an application. If the application is aimed at a pre-specified profile, the capabilities of such a profile is necessary to adjust to – see Specification of the Bluetooth System v1.0B Profiles.

a) With the HCI UART Transport Layer on top of HCI, the module will communicate with a host through the UART I/F. The PCM I/F is also available for communicating voice.
b) With the HCI USB Transport Layer on top of the HCI, the module will communicate with a host through the USB. Detach and Wake_up signals are also available for notebook implementations.

Please refer to the Specification of the Bluetooth System v1.0B part H:1-4 for in-depth information regarding the HCI and different transport layers

Please refer to the Specification of the Bluetooth System v1.0B part H:4 regarding the HCI and UART transport layers.

PCM Voice Interface
The standard PCM interface has a sample rate of 8 kHz (PCM_SYNC). The PCM clock is variable between 200 kHz and 2.0MHz. The PCM data can be linear PCM (13-16bit), µ-Law (8bit) or A-Law (8bit).

The PCM I/F can be either master or slave – providing or receiving the PCM_SYNC. Redirection of PCM_OUT and PCM_IN can be accomplished as well.

Over the air the encoding is programmable to be CVSD, A-Law or µ-Law. Preferably the robust CVSD encoding should be used.

USB Interface
The module is a USB high-speed class device (12Mbps) that has the full functionality of a USB slave and is compliant to the USB 1.1 specification. Data transfer occurs on the bi-directional ports, D+ & D-.

Additionally, there are two side band signals for a notebook application. Two side band signals Wake_up and Detach are used to control the state from which the notebook resumes. When the host is in a power down mode, Wake_up wakes the host up when the Bluetooth system receives an incoming connection. The host indicates that it is in Suspend mode by using the Detach signal.

I²C Interface
A master I²C I/F is available on the module. The control of the I²C pins are performed by Ericsson specific HCI commands available in the FW implementation – see Appendix C.

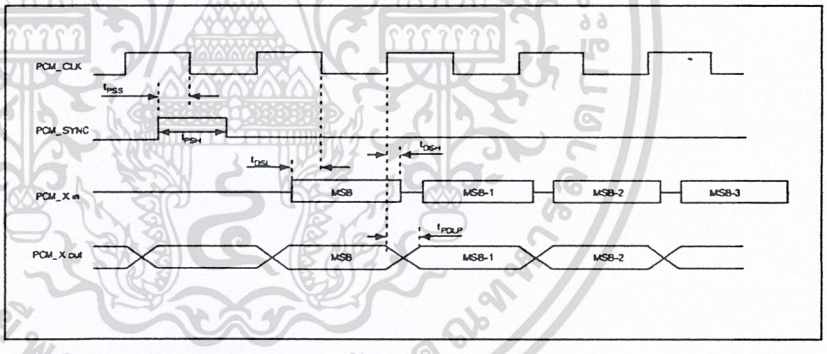


Figure 14. PCM timing

Name	Description	Min	Typ	Max	Unit
f_{pcmClk}	PCM data-transfer clock frequency 1	128		2000	kHz
$f_{pcmSync}$	PCM sample rate sync. frequency		8		kHz
t_{CCH}	PCM_CLK high period	200			ns
t_{CCL}	PCM_CLK low period	200			ns
t_{PSS}	PCM_SYNC (setup) to PCM CLK (fall)	100			ns
t_{PSH}	PCM_SYNC pulse length	200			ns
t_{DSL}	PCM_X in (setup) to PCM_CLK (fall)	100			ns
t_{DSH}	PCM_X in (hold) from PCM_CLK (fall)	100			ns
t_{PDLP}	PCM_X out valid from PCM_CLK (rise)	150			ns

Table 2. PCM parameters

Module HW Interfaces

UART Interface
The UART implemented on the module is an industry standard 16C450 and supports the following baud rates: 300, 600, 900, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400 and 460800 bits/s. 128 byte FIFOs are associated with the UART.

Four signals will be provided for the UART interface. TxD & RxD are used for data flow, and RTS & CTS is used for flow control.

Antenna
The ANT pin should be connected to a 50Ω-antenna interface, thereby supporting the best signal strength performance. Ericsson Microelectronics can recommend application specific antennas – see Appendix C.

Power-up Sequence

There is no need for a power up sequence if VCC, ON and VCC_IO are tied together.

A power up sequence, if used, shall be applied accordingly: Connection of the supply rails, GND and then V_{CC}; then the ON signal should be applied in order to initiate the internal regulators; and finally, the V_{CC_IO} supply rail can be activated.

The power-down sequence is similar to the power-up procedure but in the reverse format. Therefore, the disconnection of the signals shall be as follows: V_{CC_IO}, ON, V_{CC} and finally GND.

RESET#
The assignment of the RESET# input is to generate a reset signal to the complete Bluetooth module. During power-up the reset signal is set 'low' automatically so that power supply glitches are avoided. Therefore no reset input should be required after power-up.
When implementing an external RESET#, the signal should be fed from an open drain output.

Power
There are three inputs to the Voltage Management section (V_{CC}, V_{CC_IO}, ON). V_{CC} is the supply voltage that is typically 3.3V.

A separate power supply rail (V_{CC_IO}) is provided for the I/O ports, UART, PCM and USB. To be compliant with the USB 1.1 specification, $V_{CC_IO} \geq$ 3.11V. V_{CC_IO} can either be connected to V_{CC} or to a dedicated supply rail, which is the same as the logical interface of the host.

Shielding / EMC Requirements
The module has its own RF shielding and is approved according to the standards by FCC and ETSI.

If the approval number is not visible on the outside when the module is utilized in the final product, an exterior label must state that there is a transmitter module inside the product.

Ground
Ground should be distributed with very low impedance as a ground plane. Connect all GND pins to the ground plane.

Assembly Guidelines

Solder Paste
The ROK 101 007 module is made for surface mounting and the SSP connection pads have been formed after printing eutectic Tin/Lead solder paste. The solder paste to use is not critical as long as this is a normal eutectic solder paste. A preferred solder paste height is 150µm.

Soldering Profile
It must be noted that the module should not be allowed to be hanging upside down in the re-flow operation. This means that the module has to be assembled on the side of the PCB that is soldered last.

The re-flow process should be a regular surface mount soldering profile (full convection strongly preferred); the ramp-up should not be higher than 2°C/s and with a peak temperature of 210-235°C during 20-60 seconds.

Pad Size
It is recommended that the pads on the PCB should have a diameter of 0.7-0.9 mm. The surface finish on the PCB pads should be Nickel/Gold or a flat Tin/Lead surface or OSP (Organic Surface Protection).

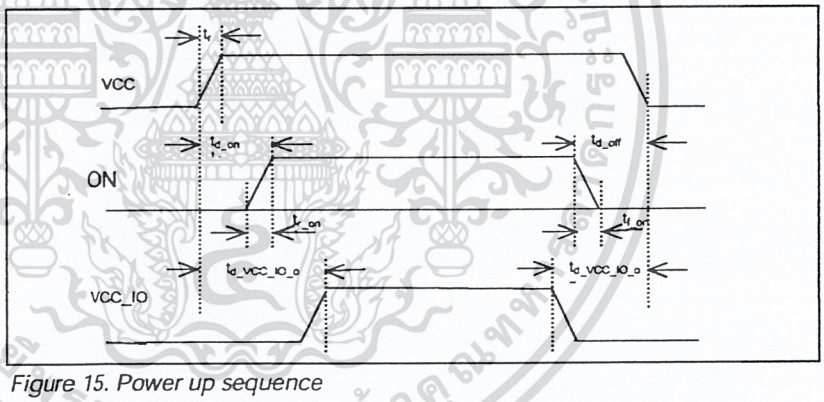


Figure 15. Power up sequence

Parameter	Min	Nom	Max	Unit
t_r			1	ms
t_{r_on}			60	µs
t_{f_on}			60	µs
t_{d_on}	t_r			µs
t_{d_off}				µs
$t_{dvcc_io_on}$	t_{d_on}			µs
$t_{dvcc_io_off}$	t_{d_off}			µs

Table 3. Power up parameters

Placement
The placement machine should be able to recognize odd BGA combinations (all ball recognition preferred) and be able to pick the component asymmetrical. The module contains a flat pick-area of 10mm diameter minimum. The weight of the module is typically 2.8gr.

Storage
Keep the component in its dry pack when not yet using the reel. After removal from the dry pack ensure that the modules are soldered onto the PCB within 48 hours.

Marking

Every module is marked with the following information on the:

a) Component designation: "ROK 101 007".
b) Ericsson's name and logotype.
c) Manufacturing code (place, year, week) and batch number.
d) CE logotype
e) Type approval RTA no. See manual

Ordering Information

Package	Part No.
30 GP	ROK 101 007/1
30 SSP	ROK 101 007/2

Information given in this data sheet is believed to be accurate and reliable. However no responsibility is assumed for the consequences of its use nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Ericsson Microelectronics. These products are sold only according to Ericsson Microelectronics' general conditions of sale, unless otherwise confirmed in writing.
Specifications subject to change without notice.

Packaging

All devices will be delivered in a package protecting them from electrostatic discharges and mechanical shock. The package will be marked with the following information:

a) Delivery address.
b) Purchase order-number
c) Type of goods and component designation.
d) Ericsson's name and logotype.
e) Date of manufacture and batch number.
f) Number of components in the package.

Abbreviations

ASIC - Application Specific Integrated Circuit
BER - Bit Error Rate
CMOS - Complementary Metal Oxide Semiconductor
C/I - Carrier to Interference Ratio
DCE - Data Circuit terminating Equipment
GP - Gold Print
HCI - Host Controller Interface
ISM - Industrial Scientific and Medical
PCB - Printed Circuit Board
PCM - Pulse Code Modulation
PDA - Personal Digital Assistant
PtP - Point to Point
Rx - Receive
SIG - Special Interest Group
SSP - Screen Solder Print
Tx - Transmit
UART - Universal Asynchronous Receiver Transmitter
USB - Universal Serial Bus
VCO - Voltage Controlled Oscillator

Microelectronics
Ericsson Microelectronics AB
SE-164 81 Kista-Stockholm, Sweden
Telephone: +46 8 757 50 00
www.ericsson.com/microe

1522-ROK 101 007 Rev. PA5
© Ericsson Microelectronics AB, April 2000

APPENDIX A

Getting Started

The ROK 101 007 Bluetooth module is easy to use when designing a Bluetooth application. However, there is a need for know-how in the Bluetooth System specification v1.0B as well as the Profile specification v1.0B when designing and end-customer product.

- Bluetooth module
- Know-how in Bluetooth specification regarding HCI commands
- Test board with UART/PCM or USB I/F
- Visual C++ for PC SW design
- Preferably HCIdriver, L2CAP, RFCOMM and SDP from Ericsson

The list above show some parts that would make designing convenient.

All information needed, regarding how to drive the HCI over UART is specified in part H4 of the Bluetooth System v1.0B further more part H1 and also Appendix IX with message charts is relevant.

Below follow an example of how to set up an ACL link between to Bluetooth modules by using the UART I/F and also a schematic of how to interface the module and control it by a host, normally a PC or microcontroller.

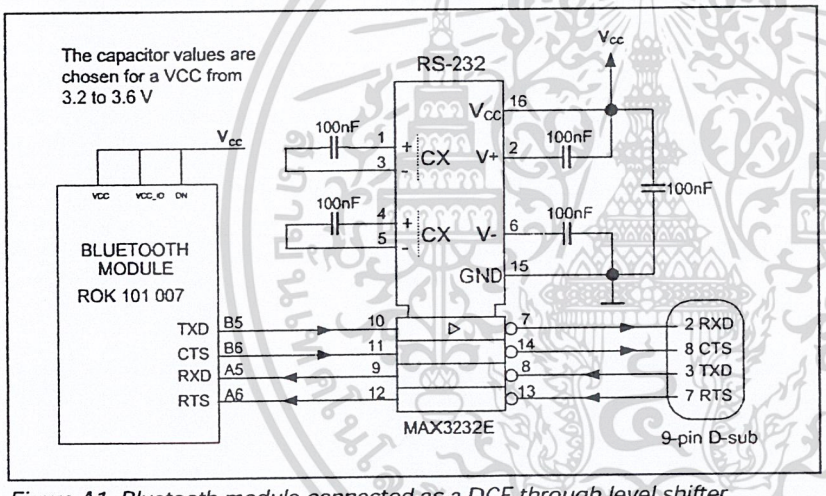


Figure A1. Bluetooth module connected as a DCE through level shifter

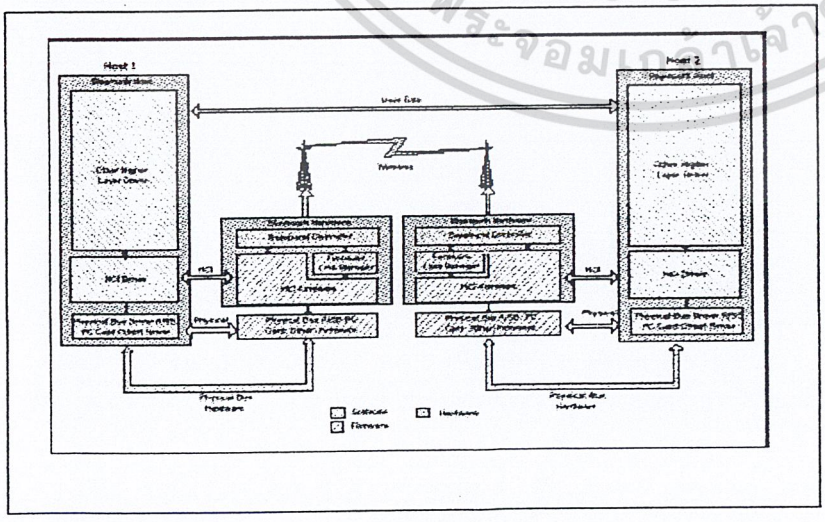


Figure A2. Bluetooth host to host communication

Principle schematic for UART inter- connect

The inter-connection to the level-shifter when designing a test-board could be according to the schematic below. (Figure A1.)

The Bluetooth module can be connected as a DCE/DTE and a modem/nullmodem cable could therefor be used in-between the test-board and the PC.

Setting up a Bluetooth point-to-point connection

The Host Controller Interface (HCI) in the module is a command I/F. The host presents commands to the HCI and receives events back from the HCI of the module. The module Link Manager provides link set up capability to the HCI.

- Host_B Bluetooth module is set in paging scan mode – listening for a Bluetooth device asking for a new connection
- Host_A Bluetooth module is set in paging mode asking for a connection to Host_B

This is accomplished by first setting up the connection between the Host and the module and thereafter creating the connection between the modules using HCI commands.

Host set-up via UART:

There are 4 different types of HCI-packets accepted on the UART I/F.

HCI packet type	HCI packet indicator
HCI command packet	0x01
HCI ACL data packet	0x02
HCI SCO data packet	0x03
HCI event packet	0x04

Table A1. HCI packets

The HCI packet indicator shall be sent immediately before the HCI packet. When the entire HCI packet has been received a new indicator should be expected.

The default speed setting is 57.6 kb/s and can be changed by sending a specified bit stream to the I/F - see Appendix C on how to change the speed setting of the UART.

When the speed set-up for the UART is made for both Host_ A & B, the Command Packets can be sent and Event Packets received by the hosts. See HCI over UART in part H:4 of the Bluetooth System v1.0B for detailed information regarding parameters and protocol.

Soft Reset
First HCI command packet to send should be the RESET packet.

A Command_Complete_Event with a status parameter should be returned to the host.

Buffer information
Buffer information should be exchanged between the module and respective host by using HCI commands.

- Read_Buffer_Size: Providing the host with information on buffer size for ACL and SCO data packets for the module returned with a Command_Complete_Event packet. The host shall use this information for controlling the transmission
- Host_Buffer_Size: Providing the module with information on buffer size for ACL and SCO packets to the host.

It is the host that manages the data buffers of the Host Controller on the module.

Timers
It could be necessary to set important timers used by the module for time out handling. The timers are all set by writing to registers using HCI commands.

The default values can be checked in Specification of the Bluetooth System v1.0B part H:1 or by using Read_xxx_xxx commands.

Bluetooth Address
The hosts, using the HCI command Read_BD_ADDR will find the Bluetooth address of the module by the Command_Complete_Event with the BD_ADDR as a parameter.

By Remote_Name_Request, the BD_ADDR of the remote module can also be found.

Inquiry
The HCI command Inquiry with the parameters LAP, Inquiry_Length, and Num_Responses can also be used for collecting BD_ADDR of remote Bluetooth units.

Creating a Point-to-point connection

Page Scan mode

Command	OCF	Command parameters	Return parameters
HCI_Write_Scan_Enable	0x0005	Scan_Enable	Status

Table A2. HCI Write Scan Enable OCF code

To set a Bluetooth module in the mode for being able to connect to (Host_B), page scan mode, there are some settings that should be performed.

The Write_Scan_Enable command with correct parameter (Scan_Enable=0x001A) will set the module in Page scan mode if the Command_Complete_Event is successful.

Furthermore the setting of authentication and encryption should be disabled (default) by using the:

- Write_Authentication_Enable
- Write_Encryption_Mode

The basic settings for getting into scan mode could be according to the below suggested script list.

- Read Buffer Size
- Set Event Filter
- Write Scan Enable: (Scan Enable: 0x03)
- Write Voice Setting: (Voice Channel Setting: 0x0060)
- Write Authentication Enable: (Authentication Enable: 0x00)

- Set Event Filter: (Connection Setup Filter: Connections from All Devices, Auto Accept: 0x02)
- Write Connection Accept Timeout: (Connection Accept Timeout: 0x2000)
- Write Page Timeout: (Page Timeout: 0x3000)

Command	OCF	Command parameters	Return parameters
HCI_Create_Connection	0x0005	BD_ADDR Packet_Type Packet_Scan_Repetition_Mode Packet_Scan_Mode Clock_Offset	

Table A3. HCI Create Connection OCF code

Page mode
The Create_Connection command is used to set-up a link to another Bluetooth device.

Create_Connection:
BD_ADDR: 0xYYYYYYYYYYYY,
Packet Type: 0x0007,
Page Scan Repetition Mode: 0x01,
Page Scan Mode: 0x00, Clock Offset: 0x0000

This command will cause the Link Manager to try to create a connection to the Bluetooth module with the appropriate BD_ADDR. The local Bluetooth module (Host_A) starts the paging process to set up a link to the page-scanning remote device (Host_B).

By LMP the negotiation between the two Bluetooth modules Link Managers (LM) the link set-up can be completed.

ACL link up and running
Host_A is the master of the point-to-point piconet and Host_B is slave. The unit starting the paging process is by definition the master. The link set-up is completed when the event Connection_Complete_Event is returned to both Host_A and Host_B with the connection handle as one of the parameters and the status parameter 0x00 (success). When Bluetooth link is up and running the HCI data packets can be sent from host to host.

The host must take care of generating the packages going from Host to Host Controller in the module over the UART I/F, in the same way the Host must arrange the packages received from the Host Controller.

Both sides need information on what kind of data is received, to be able to interpret the bit flow correctly.

For extensive information on setting up a Bluetooth link please refer to Bluetooth System v1.0B Appendix IX Message Sequence Charts

Adding an SCO link
When creating a voice connection using the PCM I/F, an ACL link must be up and running between the two devices, an SCO link can thereafter be added.

The control of the PCM I/F (FS, PCM_IN/OUT and PCM_CLK) is handled by Ericsson specific HCI command – see Appendix C.

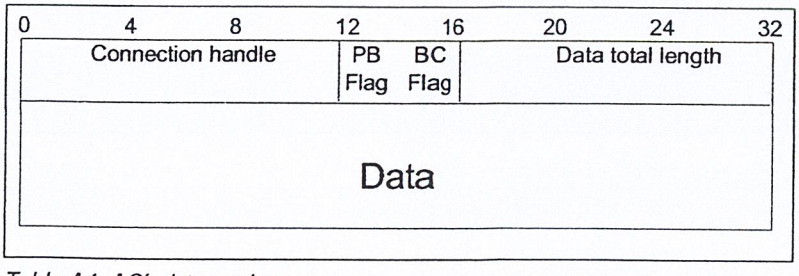


Table A4. ACL data packet

Driving SW

HCI- API
The Bluetooth module includes all HCI command capabilities according to Bluetooth System v1.0B. Furthermore, there is some Ericsson specific commands available for accessing HW registers and HW control – see Appendix C. SW for driving the module should be developed for the HCI interface.

There is source code SW available with Ericsson Bluetooth Developers Kit (EBDK), see Appendix B, which can be used for driving the module. This SW comes with the EBDK and has an HCI-API for application development on the HCI I/F.

Higher layer-API
Software (HCIdriver, L2CAP, RFCOMM and SDP) are available in a generic, source code format, i.e. to be adapted to various operating systems.

- HCIdriver – implements the HCI command driver used by the host
- L2CAP – handles protocol multiplexing, segmentation and re-assembly of packets
- RFCOMM – provides a serial port emulation over the L2CAP protocol
- SDP – Service Discovery Protocol provides information on the services available on a Bluetooth device

Additional SW for the application shall be developed for the actual application on top of the RFCOMM API. If the application is according to a SIG predefined profile, it should be implemented accordingly. New applications can be the driver of the specification of new profiles decided by the SIG – see Specification of the Bluetooth System v1.0B Profiles.

APPENDIX B

Development tools

Ericsson Bluetooth Development Kit (EBDK)
The easiest way of getting started is to use the Ericsson Bluetooth Developers Kit. It provides all parts necessary for developing applications for the Bluetooth module.

Available are:

- PC plug&play
- Demos using radio/baseband
- Macro capability
- C++ v5.0 Source code for use in applications
- HCIdriver, L2CAP, SDP and RFCOMM for applications using UART communication
- Pins for electrical measurements
- Antennas

Development can easily take place on the EBDK platform and thereafter the implementation of the full Bluetooth capability can be setup by developed SW/HW and the Bluetooth module.

Software (HCIdriver, L2CAP, RFCOMM and SDP) will be available in source code for PC.

- HCIdriver – implements the HCI command driver used by the host
- L2CAP – handles protocol multiplexing, segmentation and re-assembly of packets
- RFCOMM – provides a serial port emulation over the L2CAP protocol
- SDP – Service Discovery Protocol provides information on the services available on a Bluetooth device

Technical support is available from the EBDK distributor. Please contact Ericsson Microelectronics for ordering and information regarding the

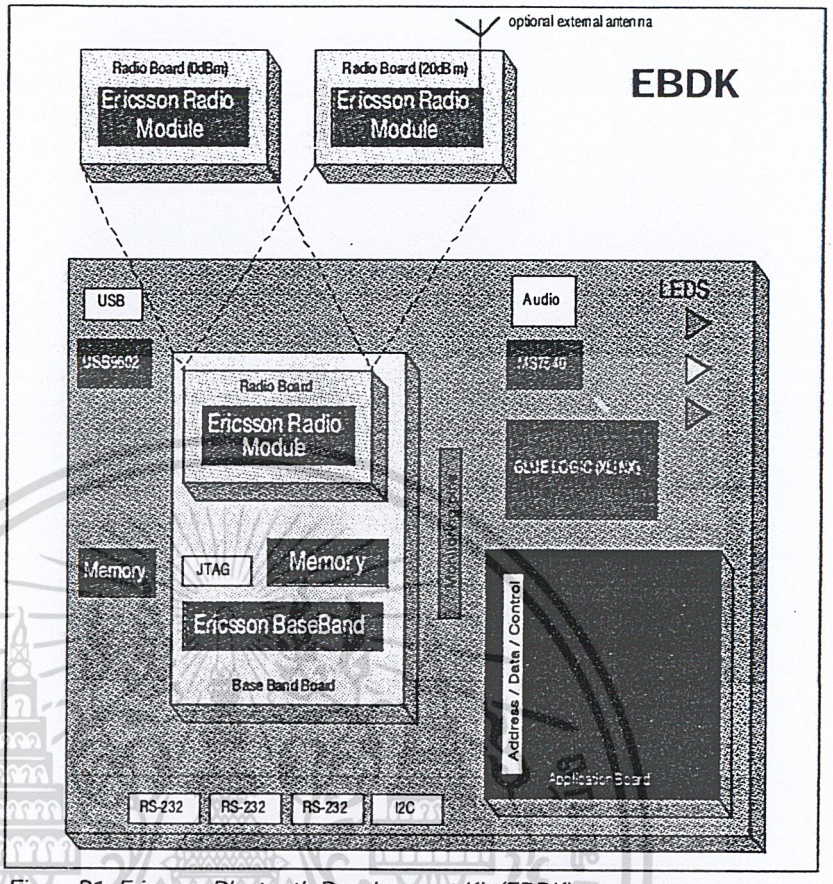


Figure B1. Ericsson Bluetooth Development Kit (EBDK)

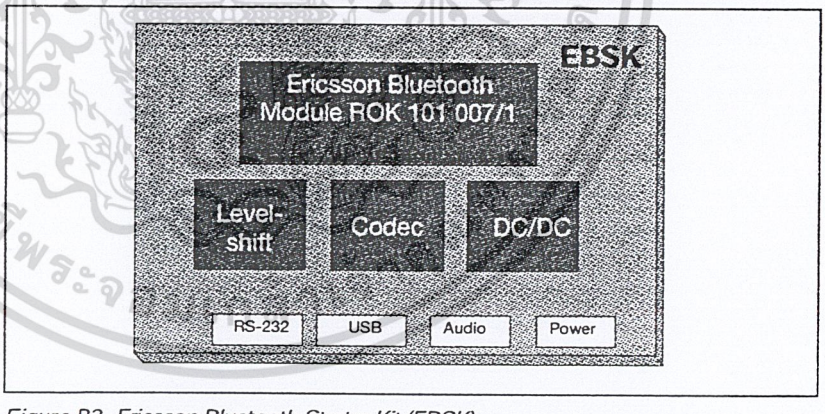


Figure B2. Ericsson Bluetooth Starter Kit (EBSK)

EBDK and regarding extra daughter board with ROK 101 007 as add-on to the EBDK.

Ericsson Bluetooth Starter Kit (EBSK)
A very small convenient kit, which preferably is used in point-to-multipoint configuration designs, based on the Bluetooth module ROK 101 007.

Please contact Ericsson Microelectronics for ordering and information regarding the Ericsson Bluetooth Starter Kit.

APPENDIX C

Bluetooth Information

This part will be updated and new information will be added continuously.

Radio

Fast Frequency hopping (1600channel hop/s) with 79(23) channels available (2.402 to 2.480 GHz) and a symbol rate of 1Ms/s over the air exploits the maximum channel bandwidth in the unlicensed ISM band.

To sustain a high transfer rate in busy radio environment, the frequency hopping together with advanced coding techniques maximizes the throughput.

During Page and Inquiry the hopping frequency is risen to 3200 hops/s to enhance the time needed for connection set-up.

Modulation technique is a binary Gaussian Frequency Shift Keying GFSK, with a BT product of 0.5. The channel bandwidth is 1 MHz and the frequency deviation from the carrier frequency of the RF channel is between +/-140 to +/-175 kHz for representing a '1'/'0'.

Country	Frequency range	RF channels	
Europe & USA	2400-2483.5 MHz	f = 2402 + k MHz	k = 0....78
Japan	2471-2497 MHz	f = 2473 + k MHz	k = 0....22
Spain	2445-2475 MHz	f = 2449 + k MHz	k = 0....22
France	2446.5-2483.5 MHz	f = 2454 + k MHz	k = 0....22

Table C1. Frequency ranges used.

A rapid process is ongoing to harmonise Spanish, French and Japanese frequency ranges with the rest of the world.

Data and parameter formats

There are exceptions in the Bluetooth system for data and parameter formats – general rules below.

- All values are in Binary and Hexadecimal little Endian formats
- Negative values must use 2's complement format
- Array parameter notation is parameterA[i], parameterB[i],...
- All parameter values are sent/received in little Endian format. The least significant byte is sent first – unless noted otherwise.

HCI Opcodes

The Opcodes have been changed during the ongoing standardisation work. Below is the description on how to send opcodes to the module.

Below is the general HCI command

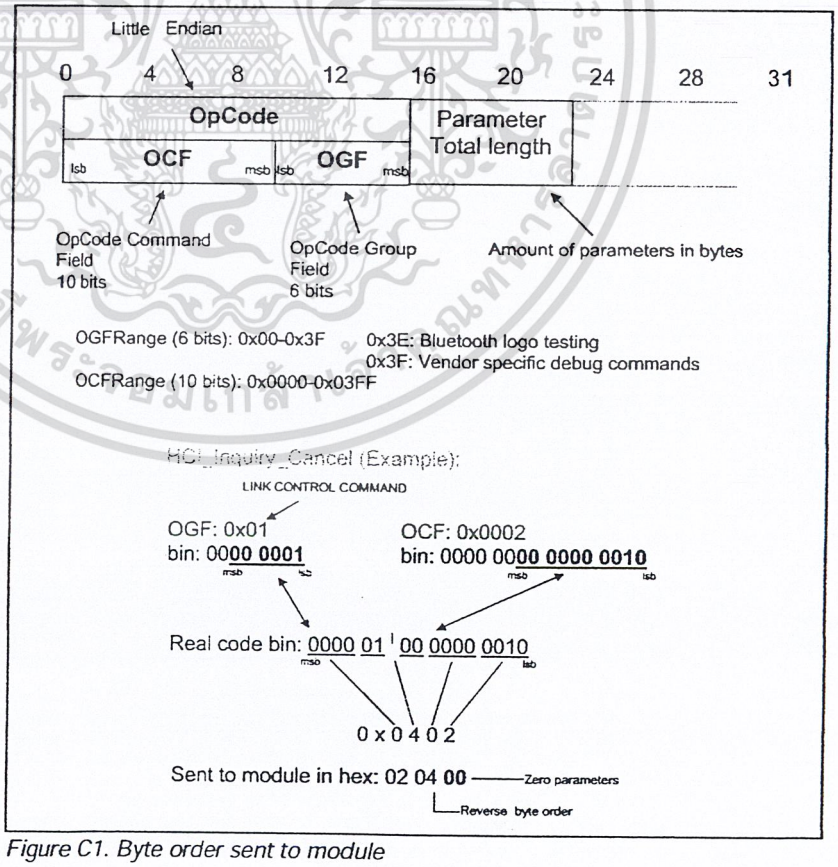


Figure C1. Byte order sent to module

ภาคผนวก ข.

Bluetooth Assigned Numbers



1 BLUETOOTH BASEBAND

1.1 THE GENERAL- AND DEVICE-SPECIFIC INQUIRY ACCESS CODES (DIACS)

The Inquiry Access Code is the first level of filtering when finding Bluetooth devices and services. The main purpose of defining multiple IACs is to limit the number of responses that are received when scanning devices within range.

#	LAP value	Usage
0	0x9E8B33	General/Unlimited Inquiry Access Code (GIAC)
1	0x9E8B00	Limited Dedicated Inquiry Access Code (LIAC)
2-63	0x9E8B01-0x9E8B32, 0x9E8B34-0x9E8B3F	RESERVED FOR FUTURE USE

Table 1.1: The Inquiry Access Codes

The Limited Inquiry Access Code (LIAC) is only intended to be used for limited time periods in scenarios where both sides have been explicitly caused to enter this state, usually by user action. For further explanation of the use of the LIAC, please refer to the Generic Access Profile [7].

In contrast it is allowed to be continuously scanning for the General Inquiry Access Code (GIAC) and respond whenever inquired.

1.2 THE CLASS OF DEVICE/SERVICE FIELD

The Class of Device/Service (CoD) field has a variable format. The format is indicated using the 'Format Type field' within the CoD. The length of the Format Type field is variable and ends with two bits different from '11'. The version field starts at the least significant bit of the CoD and may extend upwards.

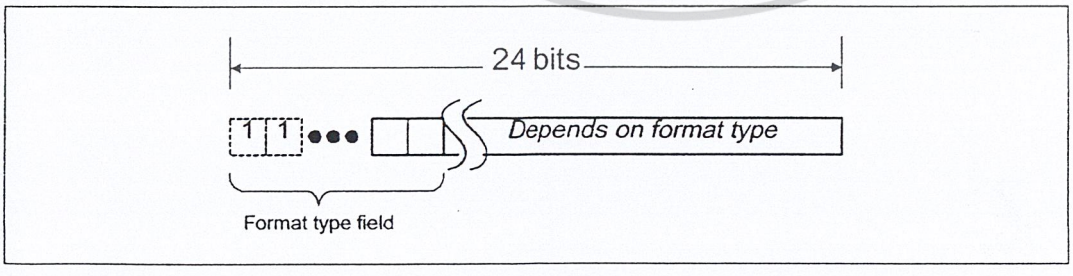


Figure 1.1: General format of Class of Device/Service

In the 'format #1' of the CoD (Format Type field = 00), 11 bits are assigned as a bit-mask (multiple bits can be set) each bit corresponding to a high level generic category of service class. Currently 7 categories are defined. These

are primarily of a 'public service' nature. The remaining 11 bits are used to indicate device type category and other device-specific characteristics.

Any reserved but otherwise unassigned bits, such as in the Major Service Class field, should be set to 0.

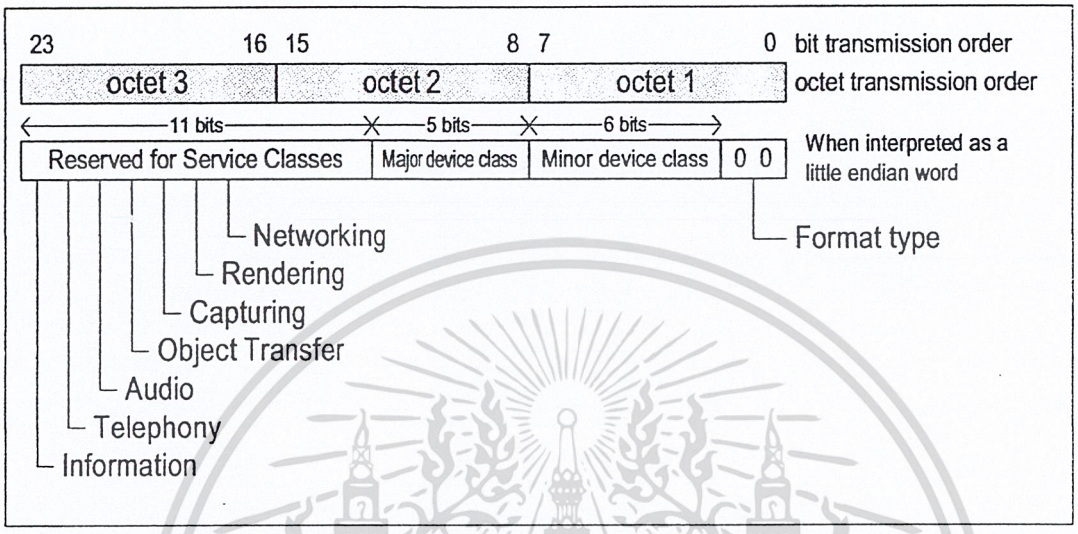


Figure 1.2: The Class of Device/Service field (format type 1). Note the order in which the octets are sent on the air and stored in memory.

1.2.1 Major Service Classes

Bit no	Major Service Class
13	Limited Discoverable Mode[1]
14	(reserved)
15	(reserved)
16	(reserved)
17	Networking (LAN, Adhoc, ...)
18	Rendering (Printing, Speaker, ...)
19	Capturing (Scanner, Microphone, ...)
20	Object Transfer (v-Inbox, v-Folder, ...)
21	Audio (Speaker, Microphone, Headset service, ...)
22	Telephony (Cordless telephony, Modem, Headset service, ...)
23	Information (WEB-server, WAP-server, ...)

Table 1.2: Major Service Classes

1. As defined in [7]

1.2.2 Major Device Classes

The Major Class segment is the highest level of granularity for defining a Bluetooth Device. The main function of a device is used to determine the major class grouping. There are 32 different possible major classes. The assignment of this Major Class field is defined in Table 1.3.

Code (bits)					Major Device Class
12	11	10	9	8	bit no of CoD
0	0	0	0	0	Miscellaneous[1]
0	0	0	0	1	Computer (desktop, notebook, PDA, organizers, ...)
0	0	0	1	0	Phone (cellular, cordless, payphone, modem, ...)
0	0	0	1	1	LAN Access Point
0	0	1	0	0	Audio (headset, speaker, stereo, ...)
0	0	1	0	1	Peripheral (mouse, joystick, keyboards, ...)
x	x	x	x	x	Range 0x06 to 0x1E reserved
1	1	1	1	1	Unclassified, specific device code not assigned

Table 1.3: Major Device Classes

1. Used where a more specific Major Device Class code is not suited (but only as specified in this document. Devices that do not have a major class code assigned can use the all-1 code until 'classified')

1.2.3 The Minor Device Class field

The 'Minor Device Class field' (bits 7 to 1 in the CoD), are to be interpreted only in the context of the Major Device Class (but independent of the Service Class field). Thus the meaning of the bits may change, depending on the value of the 'Major Device Class field'. When the Minor Device Class field indicates a device class, then the primary device class should be reported, e.g. a cellular phone that can also work as a cordless handset should use 'Cellular' in the minor device class field.

1.2.4 Minor Device Class field - Computer Major Class

Code (bits)						Minor Device Class
7	6	5	4	3	2	bit no of CoD
0	0	0	0	0	0	Unclassified, code for device not assigned
0	0	0	0	0	1	Desktop workstation
0	0	0	0	1	0	Server-class computer
0	0	0	0	1	1	Laptop
0	0	0	1	0	0	Handheld PC/PDA (clam shell)
0	0	0	1	0	1	Palm sized PC/PDA
x	x	x	x	x	x	Range 0x06-0x7F reserved

Table 1.4: Sub Device Class field for the 'Computer' Major Class

1.2.5 Minor Device Class field - Phone Major Class

Code (bits)						Minor Device Class
7	6	5	4	3	2	bit no of CoD
0	0	0	0	0	0	Unclassified, code not assigned
0	0	0	0	0	1	Cellular
0	0	0	0	1	0	Cordless
0	0	0	0	1	1	Smart phone
0	0	0	1	0	0	Wired modem or voice gateway
x	x	x	x	x	x	Range 0x05-0x7F reserved

Table 1.5: Sub Device Classes for the 'Phone' Major Class

1.2.6 Minor Device Class field - LAN Access Point Major Class

Code (bits)			Minor Device Class
7	6	5	bit no of CoD
0	0	0	Fully available
0	0	1	1-17% utilized
0	1	0	17 - 33% utilized
0	1	1	33 - 50% utilized
1	0	0	50 - 67% utilized
1	0	1	67 - 83% utilized
1	1	0	83 - 99% utilized
1	1	1	No Service Available[1]

Table 1.6: The LAN Access Point Load Factor field

1. "Device is fully utilized and cannot accept additional connections at this time, please retry later"

The exact loading formula is not standardized. It is up to each LAN Access Point implementation to determine what internal conditions to report as a utilization percentage. The only requirement is that the number reflects an ever-increasing utilization of communication resources within the box. As a recommendation, a client that locates multiple LAN Access Points should attempt to connect to the one reporting the lowest load.

Code (bits)			Minor Device Class
4	3	2	bit no of CoD
0	0	0	Unclassified (use this value if no other apply)
x	x	x	range 0x01-0x0F reserved

Table 1.7: Reserved sub-field for the LAN Access Point

1.2.7 Minor Device Class field - Audio Major Class

Code (bits)						Minor Device Class
7	6	5	4	3	2	bit no of CoD
0	0	0	0	0	0	Unclassified, code not assigned
0	0	0	0	0	1	Device conforms to the Headset profile [9]
x	x	x	x	x	x	Range 0x02-0x7F reserved

Table 1.8: Sub Device Classes for the 'Audio' Major Class

2 LINK MANAGER PROTOCOL (LMP)

2.1 THE LINK MANGER VERSION PARAMETER

Parameter name	Assigned values	
VersNr	0	Bluetooth LMP 1.0, [2]
	1-255	(reserved)

Table 2.1: The LMP Version Parameter Values

2.2 THE LMP_COMPID PARAMETER CODES

This is the parameter used in the LMP Version procedure.

Code	Company
0	Ericsson Mobile Communications
1	Nokia Mobile Phones
2	Intel Corp.
3	IBM Corp.
4	Toshiba Corp.
5 - 65534	(reserved)
65535	Unassigned. For use in internal and interoperability tests before a Company ID has been assigned. May not be used in products.

Table 2.2: The LMP_CompId parameter codes

3 LOGICAL LINK CONTROL AND ADAPTATION PROTOCOL (L2CAP)

Please see Section 4.3 for assigned PSM values.

3.1 CHANNEL IDENTIFIERS

Destination CID	Protocol/usage	Reference
0x0000	Illegal, should not be used	[3]
0x0001	L2CAP signalling channel	[3]
0x0002	L2CA connection less data	[3]
0x0003 - 0x003F	(reserved)	

Table 3.1: Pre-defined L2CAP Channel Identifiers

3.2 PROTOCOL AND SERVICE MULTIPLEXOR (PSM)

Protocol	PSM	Reference
SDP	0x0001	[4]
RFCOMM	0x0003	[5]
TCS-BIN	0x0005	[6]
TCS-BIN-CORDLESS	0x0007	[6]

Table 3.2: Assigned Protocol and Service Multiplexor values (PSM)

4 SERVICE DISCOVERY PROTOCOL (SDP)

4.1 UNIVERSALLY UNIQUE IDENTIFIER (UUID) SHORT FORMS

The Bluetooth Service Discovery Protocol (SDP) specification defines a way to represent a range of UUIDs (which are nominally 128-bits) in a shorter form. A *reserved* range of 2^{32} values can be represented using 32-bits (denoted uuid32). Of these, a sub-range of 2^{16} values can be represented using only 16-bits (denoted uuid16). Any value in the 2^{32} range that is not assigned in this document is reserved pending future revisions of this document. In other words, no value in this range may be used except as specified in this or future revisions of this document. UUID values outside of this range can be allocated as described in [19] for any purpose the allocater desires.

4.2 BASE UNIVERSALLY UNIQUE IDENTIFIER (UUID)

The Base UUID is used for calculating 128-bit UUIDs from 'short UUIDs' (uuid16 and uuid32) as described in the SDP Specification [4].

Mnemonic	UUID
BASE_UUID	00000000-0000-1000-8000-00805F9B34FB

4.3 PROTOCOLS

Mnemonic	UUID	Name	Ref.
SDP	uuid16: *0x0001*[1]	sdp.bt	[4]
RFCOMM	uuid16: *0x0003*	com.bt	[5]
TCS-BIN	uuid16: *0x0005*	tcs.bt	[6]
L2CAP	uuid16: *0x0100*		[3]
IP	uuid16: *0x0009*		
UDP	uuid16: *0x0002*		
TCP	uuid16: *0x0004*		
TCS-AT	uuid16: *0x0006*	modem	
OBEX	uuid16: *0x0008*	obex	
FTP	uuid16: *0x000A*	ftp	
HTTP	uuid16: *0x000C*	http	
WSP	uuid16: *0x000E*	wsp	

Table 4.1: Protocol Universally Unique Identifiers and Names

1. 'Short UUID'

4.4 SERVICE CLASSES

Mnemonic	UUID	Profile[1]	AbstractName
ServiceDiscoveryServerServiceClassID	uuid16: *0x1000*		
BrowseGroupDescriptorServiceClassID	uuid16: *0x1001*		
PublicBrowseGroup	uuid16: *0x1002*		
SerialPort	uuid16: *0x1101*	[7]	serial.bt
LANAccessUsingPPP	uuid16: *0x1102*		
DialupNetworking	uuid16: *0x1103*	[13]	
IrMCSync	uuid16: *0x1104*	[17]	
OBEXObjectPush	uuid16: *0x1105*	[16]	
OBEXFileTransfer	uuid16: *0x1106*	[15]	
IrMCSyncCommand	uuid16: *0x1107*	[17]	
Headset	uuid16: *0x1108*	[7]	headset
CordlessTelephony	uuid16: *0x1109*	[10]	
Intercom	uuid16: *0x1110*	[11]	
Fax	uuid16: *0x1111*	[12]	
HeadsetAudioGateway	uuid16: 0x1112	[7]	
PnPInformation	uuid16: *0x1200*		
GenericNetworking	uuid16: *0x1201*	n/a	
GenericFileTransfer	uuid16: *0x1202*	n/a	
GenericAudio	uuid16: *0x1203*	n/a	
GenericTelephony	uuid16: *0x1204*	n/a	

Table 4.2: Service Class Identifiers and Names

1. If the specified Service Class directly and exactly implies a certain Profile, the Profile is indicated here (i.e. for concrete Service Classes). Leave empty for abstract Service Classes.

The Profile column in Table 4.2 indicates which Service Class identifiers that also directly corresponds to a Bluetooth Profile. It is not allowed to use the Service Class UUID unless the service complies with the specified Profile. These UUIDs might also appear as Profile Identifiers in the BluetoothProfileDescriptorList attribute.

4.5 ATTRIBUTE IDENTIFIER CODES

Mnemonic	Attribute ID	Reference
ServiceRecordHandle	0x0000	[4] *Bluetooth Service Discovery Protocol (SDP)*, Bluetooth SIG
ServiceClassIDList	0x0001	
ServiceRecordState	0x0002	
ServiceID	0x0003	
ProtocolDescriptorList	0x0004	
BrowseGroupList	0x0005	
LanguageBaseAttributeIDList	0x0006	
ServiceInfoTimeToLive	0x0007	
ServiceAvailability	0x0008	
BluetoothProfileDescriptorList	0x0009	
DocumentationURL	0x000A	
ClientExecutableURL	0x000B	
Icon10	0x000C	
IconURL	0x000D	
Reserved	0x000E-0x01FF	
ServiceName	0x0000 + b[1]	
ServiceDescription	0x0001 + b	
ProviderName	0x0002 + b	
VersionNumberList	0x0200	
ServiceDatabaseState	0x0201	
GroupID	0x0200	
Remote audio volume control	0x0302[2]	[7]
External network	0x0301	[10]
Service Version	0x0300	
Supported Data Stores List	0x0301	[17]
Supported Formats List	0x0303	[16]

Table 4.3: Attribute Identifiers

Mnemonic	Attribute ID	Reference
Fax Class 1 Support	0x0302	[12]
Fax Class 2.0 Support	0x0303	
Fax Class 2 Support	0x0304	
Audio Feedback Support	0x0305	

Table 4.3: Attribute Identifiers

1. 'b' in this table represents a base offset as given by the LanguageBaseAttributeIDList attribute. For the primary language, 'b' must be equal to 0x0100 as described in the SDP specification.
2. Items in *italic* are tentative values in this version of the document.

4.6 PROTOCOL PARAMETERS

Protocol	Parameter mnemonic	Index
L2CAP	PSM	1
TCP or UDP	Port	1
RFCOMM	Channel	1

Table 4.4: Protocol Parameters

4.7 HOST OPERATING ENVIRONMENT IDENTIFIERS

4.7.1 ClientExecutableURL substitution strings

The operating environment identifier strings have the following format[1]:

<cpu_type>-<manufacturer>-[<kernel>-]<os>[<version>][-<object_format>]

The general rule is that is that a new identifier should only be defined as required to differentiate incompatible operating environments concerning an executable file image. That is, for example different <version>-tags should not be used for compatible versions of the same operating system.

1. It is based on a format used by the GNU AutoConfig tools

Currently defined tags:

CPU-Type ID	Description
alpha	Digital Alpha* compatible
arm	ARM* core or compatible
i86	Any Intel* 80x86-family compatible CPU
i960	Intel* i960 compatible
jvm	Java Virtual Machine*
mips	MIPS MIPS* compatible
ppc	IBM/Motorola PowerPC* compatible
sh3	Hitachi SH-3* compatible
sh4	Hitachi SH-4* compatible
sparc	Sun Sparc* compatible
Kernel ID	**Description**
chorus, linux, javaos, os9, qnx, vxworks	
<os>	An 'OS identifier' as listed below, might appear in the <kernel> field when the requested OS platform is Java based.
OS+Version-Identifiers	
amigaos, beos4.5, ejava, epocc, epoce, epocq, epocs, gnu, jre1.1, jre1.2, macos, macosx, os2, os9, palmos, pjava, pjava1.1, photon, plan9, qnx, rtjava, win95, win98, win2000, wince, winnt4	
Object Format Identifiers[1]	
aout, bout, coff, elf, jar	
Manufacturer Identifiers	
amiga*, apple*, be*, ericsson*, ibm*, intel*, lucent*, microsoft*, microware*, motorola*, nokia*, palm*, psion*, qnx*, sun*, symbian*, toshiba*, unknown[2]	

1. Only applicable when the object format is not otherwise uniquely implied by the identifier string.
2. Use when no other applies.

For Linux, the 'manufacturer' field may be used to indicate Linux distribution if so required (in which case <version> indicates the version of the distribution). Otherwise use 'unknown'.

Linux Distribution Identifiers
caldera, debian, dlx, doslinux, linuxpro, linuxware, mandrake, mklinux, redhat, slackware, stampede, suse, turbolinux, yggdrasil

Example Operating Environment Identifier Strings		
i86-microsoft-win95	ppc-apple-macos	i86-redhat-linux-gnu6
i86-microsoft-win98	m68k-apple-macos	ppc-mklinux-linux-gnu
i86-microsoft-winnt4	ppc-apple-macosx	
alpha-microsoft-winnt4	i86-apple-macosx	
i86-microsoft-win2000	m68k-amiga-amigaos	
alpha-microsoft-win2000	ppc-amiga-amigaos	
i86-be-beos4.5	jvm-sun-jre1.2	
ppc-be-beos4.5	jvm-sun-pjava1.1	
arm-symbian-epoc3	jvm-sun-ejava	
i86-unknown-linux-gnu	m68k-palm-palmos-coff	
sh3-microsoft-wince	ppc-ibm-vxworks-pjava1.2	
arm-microsoft-wince	sparc-sun-javaos-jre1.2	

4.7.2 IconURL substitution strings

The IconURL operating environment identifier strings have the following general format:

<horizontal_pixels>x<vertical_pixels>x<color_depth>[m].<file_format>

The optional tag 'm' indicates monochrome or grayscale. The host is free to try to match/request any graphics file format as indicated by a <file_format> tag, however at a minimum files conforming to the Portable Network Graphic standard [18] should be made available at the resulting URL (indicated by <file_format>=png)[2].

File format tag	Description
png	Portable Network Graphics [18]
gif	Graphics Interchange File format
bmp	Windows bitmap

Currently defined IconURL Icon format identifier strings:

Example Icon format Identifier Strings	
32x32x8.png	256 color 32 by 32 icon (or 255 colors + transparent)
16x16x8.png	
16x16x1m.png	Black and white (or monochrome + transparent)
10x10x2m.png	4 gray-scales

2. The use of PNG, and whether a subset of PNG should be required, is currently pending further investigation.

บรรณานุกรม

Telefonaktiebolaget LM Ericsson, International Business Machines Corporation, Intel Corporation, Nokia Corporation, Toshiba Corporation, "Specification of the Bluetooth System, Specification Volume1, Core", Version 1.0b, December 1999

Telefonaktiebolaget LM Ericsson, International Business Machines Corporation, Intel Corporation, Nokia Corporation, Toshiba Corporation, "Specification of the Bluetooth System, Specification Volume2, Profiles", Version 1.0b, December 1999

Jennifer Bray, Charles F Sturman, "Bluetooth Connect Without Cables", Prentice-Hall Inc, 2001

"Bluetooth Application & Training Tool Kit", Ericsson Mobile Communication AB, 2000

T. Muller, "Bluetooth Security Architecture", A Bluetooth SIG White Paper, Revision 1.0, July 1999

Riku Mettala, "Bluetooth Protocol Architecture", A Bluetooth SIG White Paper, Revision 0.95, July 1999

Infrared Data Association, "IrMC (Ir Mobile Communications) Specification", Version 1.1, February 1999

P. Megowan, D. Suvak, andd D. Kogan, "Infrared Data Association, IrDA Object Exchange Protocol (IrOBEX)", Version 1.2 April 1999

The Internet Mail Consortium, "vCard – The Electronic Business Card Exchange Format", Version 2.1, September 1996

The Internet Mail Consortium,"vCalendar – The Electronic Calendaring and Scheduling Exchange Format", Version 1.0, September 1996

Wireless Application Protocol Forum, "Wireless Application Protocol", Version 1.0, 1998

802.11 Architecture presentation, (Ref: IEEE P802.11-96/49B), IEEE Std 802.11-1997, IEEE 802.11-1999, IEEE 802.11a-1999 and IEEE 802.11b-1999 March 1996

HomeRF Working Group, Home Networking Technologies, Comdex, November 1999

Brent Miller, "Mapping Salutation Architecture APIs to Bluetooth Service Discovery Layer", Version 1.0, July 1999

อภิเนตร อูนากูล, "OBJECT-ORIENTED ANALYSIS AND DESIGN", ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2543