โปรแกรมส่งโฆษณาผ่านบลูทูธอัตโนมัติ

Automatic Advertise Program Through Bluetooth

นางสาวกาญจนา ธะนีวรรณ์ รหัส 48362179

นายบรรชา จำปาศักดิ์ รหัส 48364791

นายเอกลักษณ์ รังสิกรรพุม รหัส 48365071

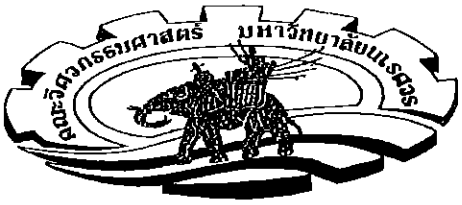
ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 25 มี.ค. 2553 /...../.....
เลขทะเบียน ใ๕๐๐๓๐๔๒
เลขเรียกหนังสือ.....
มหาวิทยาลัยนเรศวร

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2551



ใบรับรองโครงงานวิศวกรรม

หัวข้อโครงงาน	โปรแกรมส่งโฆษณาผ่านบลูทูธอัตโนมัติ		
ผู้ดำเนินโครงงาน	นางสาวกาญจนา	ธะนีวรรณ์	รหัส 48362179
	นายบรรชา	จำปาศักดิ์	รหัส 48364791
	นายเอกลักษณ์	รังสิกรรพุม	รหัส 48365071
อาจารย์ที่ปรึกษา	อาจารย์แสงชัย มังกรทอง		
สาขาวิชา	วิศวกรรมคอมพิวเตอร์		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2551		

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะกรรมการสอบโครงงานวิศวกรรม

..... ประธานกรรมการ

(อาจารย์แสงชัย มังกรทอง)

..... กรรมการ

(ดร.ไพศาล มุณีสว่าง)

..... กรรมการ

(อาจารย์เศรษฐา ตั้งค้าวานิช)

หัวข้อโครงงาน	โปรแกรมส่งโฆษณาผ่านบลูทูธอัตโนมัติ		
ผู้ดำเนินโครงงาน	นางสาวกาญจนา	ธะนีวรรณ์	รหัส 48362179
	นายบรรชา	จำปาศักดิ์	รหัส 48364791
	นายเอกลักษณ์	รังสิกรรพุม	รหัส 48365071
อาจารย์ที่ปรึกษา	อาจารย์แสงชัย มังกรทอง		
สาขาวิชา	วิศวกรรมคอมพิวเตอร์		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2551		

.....

บทคัดย่อ

โครงงานนี้เป็นการพัฒนาโปรแกรมในการส่งโฆษณาในรูปแบบข้อความหรือรูปภาพผ่านระบบสื่อสารไร้สายบลูทูธ (Bluetooth) โดยกระบวนการทำงานของโปรแกรมมีการทำงานอยู่ 3 ขั้นตอน คือ (1) การค้นหาอุปกรณ์บลูทูธภายในระยะทางที่ถูกค้นพบได้ (2) การค้นหาเซอร์วิสไฟล์ทรานเฟอร์ (File Transfer Service) ของอุปกรณ์บลูทูธที่ค้นพบ (3) การสร้างเซสชัน (Session) ในการเชื่อมต่อระหว่างอุปกรณ์บลูทูธรวมถึงการจับคู่ระหว่างอุปกรณ์ (Pair device) และการส่งไฟล์โฆษณา

จากผลการทดลองและการวิเคราะห์ผลการทดลอง พบว่าประสิทธิภาพโดยรวมของโปรแกรมขึ้นอยู่กับปัจจัยต่างๆดังนี้ อุปกรณ์บลูทูธ (Local Devices) ในแต่ละเวอร์ชันมี Class ต่างกัน, วิธีในการยืนยันกดรับโฆษณาต่างกัน, จำนวนเครื่องที่ถูกค้นพบในแต่ละรอบการทำงาน, อุปกรณ์บลูทูธ (Remote Devices) ในแต่ละรุ่นต่างกัน ซึ่งปัจจัยต่างๆจะมีผลต่อเวลาในการทำงานของโปรแกรมและทำให้ประสิทธิภาพโดยรวมเปลี่ยนแปลงไปด้วย

Project Title	Program Bluetooth automatic advertise transfer		
Name	Mrs. Kanjana	Taneewan	ID. 48362179
	Mr. Bancha	Jampasak	ID. 48364791
	Mr.Aekaluk	Rungsigunpum	ID. 48365071
Project Advisor	Mr. Sangchai	Mangkornthong	
Major	Computer Engineering		
Department	Electrical and Computer Engineering		
Academic Year	2008		

.....

ABSTRACT

This study is to develop program for sending advertisement in text or picture through Bluetooth communication system by using the process of this program that has 3 steps by following 1) find Bluetooth devices within a distance that is found 2) find File transfer Service of Bluetooth devices that is found 3) build session in connection between Bluetooth devices, pair devices and sending advertise file

From the result and the analyze of efficiency include in the study.It is found that depends on these elements.First, local devices in each version have different in class.Second, difference of method in accept to receive advertisement.Third, amount of remote devices that are discovered in a round process.Finally difference of remote devices.These elements have effect to time of program process and change efficiency too.

กิตติกรรมประกาศ

การที่โครงงานนี้สำเร็จลุล่วงไปได้ด้วยดีทางผู้จัดทำใคร่ขอแสดงความขอบพระคุณเป็นอย่างสูงต่ออาจารย์ที่ปรึกษาโครงงาน อาจารย์แสงชัย มังกรทอง ดร.ไพศาล มุณีสว่าง และอาจารย์เศรษฐา ตั้งค้าวานิชอาจารย์ประจำภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร ที่ได้กรุณาให้แนวความคิด ช่วยชี้แนะแนวทางในการทำโครงงาน ตลอดจนกรุณาเอื้อเฟื้อเอกสารต่างๆที่เกี่ยวข้องกับโครงงานนี้ อีกทั้งยังช่วยแนะนำแหล่งข้อมูลในการค้นคว้าข้อมูลเพิ่มเติม ทำให้เป็นประโยชน์ต่อการทำโครงงานของผู้จัดทำ

นางสาวกาญจนา ธะนีวรรณ์
นายบรรชา จำปาศักดิ์
นายเอกลักษณ์ รังสิกรรพุม

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ก

บทคัดย่อภาษาอังกฤษ ..... ข

กิตติกรรมประกาศ ..... ค

สารบัญ ..... ง

สารบัญ(ต่อ) ..... จ

สารบัญตาราง ..... ฉ

สารบัญรูป ..... ช

บทที่ 1 บทนำ

1.1 ที่มาและความสำคัญของโครงงาน 1

1.2 วัตถุประสงค์ของโครงงาน 1

1.3 ขอบข่ายของโครงงาน 2

1.4 ขั้นตอนของการดำเนินงาน 2

1.5 แผนการดำเนินงาน 3

1.6 ผลที่คาดว่าจะได้รับ 4

1.7 งบประมาณโครงงาน 4

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง

2.1 บลูทูธ 5

2.2 โครงสร้างการทำงาน 6

2.3 จาวาบลูทูธ เอพีไอ (Java Bluetooth API) 9

2.4 เทคนิคการส่งข้อมูล 14

2.5 ภาษาจาวา (Java Language) 17

2.6 Introduction Java 19

สารบัญ (ต่อ)

หน้า

บทที่ 3 วิธีการดำเนินการ

3.1 ศึกษาโครงสร้าง Bluetooth Protocol Stack Diagram 22

3.2 ออกแบบ Activity Diagram 23

3.3 ออกแบบ Class Diagram 24

3.4 ดำเนินการเขียนโปรแกรม 27

บทที่ 4 ผลการทดลอง

4.1 ตารางบันทึกผลเวลาการทำงานของโปรแกรม 31

บทที่ 5 บทสรุป

5.1 สรุปผลการทดลอง 34

5.2 วิเคราะห์ผลการทดลอง 34

5.3 ปัญหาและแนวทางการแก้ไข 34

เอกสารอ้างอิง 35

ภาคผนวก 36

ประวัติผู้เขียนโครงงาน 48

สารบัญตาราง

ตารางที่ หน้า

1.1 แผนการดำเนินงานโครงงาน Program Bluetooth Automatic Advertise Transfer 3

2.1 การแบ่งคลาสของอุปกรณ์ตามกำลังส่ง 16

2.2 สรุปข้อมูลด้านต่างๆของเทคโนโลยี Bluetooth 16

2.3 แสดงลำดับความสำคัญของโอเปอเรเตอร์ทางคณิตศาสตร์ 20

2.4 แสดงลำดับความสำคัญของชนิดข้อมูลต่างๆจากสูงสุดไปต่ำสุด 21

4.1 เวลาค้นหาอุปกรณ์ที่ใช้ในการส่งโฆษณาตามจำนวนเครื่อง 31

4.2 เวลาหน่วงที่ใช้ในการส่งโฆษณาแบบเลือกกดรับตามจำนวนเครื่อง 32

4.3 เวลาหน่วงที่ใช้ในการส่งโฆษณาแบบกดรับทันทีไม่ทำการ pairing devices 32

4.4 เวลาหน่วงที่ใช้ในการส่งโฆษณาแบบเลือกไม่กดรับตามจำนวนเครื่อง 32

สารบัญรูป

รูปที่ หน้า

2.1 Scatternet ที่ประกอบด้วย 3 Piconet 7

2.2 โครงสร้างของ Bluetooth Protocol stack 8

2.3 โครงสร้างของ Bluetooth Protocol stack 10

2.4 การใช้ Bluetooth-Enabled Application 11

2.5 ไดอะแกรม Device discovery 13

3.1 Bluetooth Protocol Stack Diagram 22

3.2 Activity Diagram 23

3.3 Class BluetoothLocalDevice 24

3.4 Class BluetoothRemoteDevice 24

3.5 Class BluetoothLocalService 25

3.6 Class BluetoothConnectClients 25

3.7 Class BluetoothJFrame 26

3.8 Class BluetoothDeviceLists 26

3.9 Class StructLists 27

3.10 ภาพแสดงโค้ดในส่วนของ คลาส BluetoothLocalDevice 27

3.11 ภาพแสดงโค้ดในส่วนของ คลาส BluetoothRemoteDevices 28

3.12 ภาพแสดงโค้ดในส่วนของ คลาส BluetoothRemoteServices 28

3.13 ภาพแสดงโค้ดในส่วนของ คลาส BluetoothConnectClients 29

3.14 ภาพแสดงโค้ดในส่วนของ คลาส StructLists 29

3.15 ภาพแสดงโค้ดในส่วนของ คลาส RemoteDeviceLists 30

3.16 ภาพแสดงโค้ดในส่วนของ คลาส BluetoothJFrame 30

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

ในปัจจุบันการโฆษณาสินค้ามีบทบาทและความสำคัญต่อสินค้าทุกประเภท การโฆษณาสินค้าของห้างสรรพสินค้าเป็นข่าวสารที่มีความสำคัญสำหรับลูกค้า เพื่อให้ลูกค้าได้รับการโฆษณาสินค้าต่างๆ เช่น โปรโมชั่นของสินค้า การลดราคาของสินค้า เป็นต้น ซึ่งข้อมูลการโฆษณาจะถูกส่งให้กับลูกค้าเพื่อให้ได้รับข้อมูลการโฆษณาต่างๆเกี่ยวกับสินค้าได้ทั่วถึงกัน ซึ่งทำให้ห้างสรรพสินค้าขายสินค้าได้มากขึ้น โปรแกรมจะทำการค้นหาอุปกรณ์บลูทูธแล้วจะทำการส่งไฟล์โฆษณาในรูปแบบไฟล์ภาพหรือไฟล์ข้อความ เพื่อให้ลูกค้าได้รับข่าวโฆษณาเกี่ยวกับโปรโมชั่น หรือ สินค้าลดราคาต่างๆ ผ่านระบบบลูทูธ โดยโปรแกรมนี้จะทำงานโดยอัตโนมัติ

การโฆษณาสินค้าในระบบบลูทูธ สำหรับลูกค้าที่มีอุปกรณ์สื่อสารแบบเคลื่อนที่ ที่มีระบบบลูทูธติดตั้งอยู่ เพื่อให้ได้รับโฆษณาสินค้าได้อย่างสะดวกและรวดเร็ว เป็นการส่งเสริมโฆษณาให้เข้าถึงตัวลูกค้ามากขึ้น ดังนั้นการโฆษณาผ่านระบบบลูทูธจึงเป็นสิ่งสำคัญที่ควรศึกษาและพัฒนาให้มีประสิทธิภาพมากขึ้น

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 เพื่อจัดทำโปรแกรมส่งโฆษณาผ่านบลูทูธอัตโนมัติ ผ่านระบบการส่งข้อมูลแบบไร้สายด้วย อุปกรณ์บลูทูธ

1.2.2 เพื่อศึกษาการใช้งานเกี่ยวกับภาษา Java API for Bluetooth (JSR-82) with BlueCove library

1.2.3 เพื่อศึกษาและพัฒนาโปรแกรมที่ทำงานผ่านระบบบลูทูธ

1.2.4 ช่วยลดต้นทุนในการโฆษณาเช่น ใบปลิว เอกสารโฆษณาตีพิมพ์ เป็นต้น เพราะต้นทุนในการผลิต โปรแกรมค่อนข้างต่ำ

1.3 ขอบข่ายของโครงงาน

1.3.1 จัดทำโปรแกรมส่งโฆษณาผ่านบลูทูธอัตโนมัติ โดยโปรแกรมจะทำงานแบบ อัตโนมัติ และสามารถส่งได้เฉพาะโทรศัพท์เคลื่อนที่บางรุ่นเท่านั้น เช่น Nokia N72, Nokia N70, Nokia N-Gage QD

1.3.2 โปรแกรมสามารถส่งโฆษณาประเภท ไฟล์ภาพ, ไฟล์ข้อความ และขนาดไฟล์ไม่เกิน 2 MB เท่านั้น

1.4 ขั้นตอนของการดำเนินงาน

1.4.1 ศึกษาโครงสร้างและการทำงาน Bluetooth Protocol Stack เพื่อออกแบบการเขียนโปรแกรม

1.4.2 เขียนโปรแกรมเพื่อจำลองการค้นหาอุปกรณ์บลูทูธและเชื่อมต่อกับอุปกรณ์รับส่งโฆษณาผ่านบลูทูธแบบเคลื่อนที่

1.4.3 ทดสอบการทำงานและเก็บผลการทดลอง

1.4.4 ทำการปรับปรุงและแก้ไขข้อผิดพลาดที่ตรวจพบในส่วนต่างๆของโครงงาน

1.4.5 วิเคราะห์และสรุปผลการทดลอง

1.4.6 จัดทำรูปเล่มรายงาน

1.5 แผนการดำเนินงาน

ตารางที่ 1.1 แผนการดำเนินงานโครงงาน Program Bluetooth Automatic Advertise Transfer

กิจกรรม	ปี 2551						ปี 2552		
	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
1. ศึกษาโครงสร้างและการทำงาน Bluetooth Protocol Stack	←	→							
2. เขียนโปรแกรมเพื่อจำลองการค้นหาอุปกรณ์บลูทูธและเชื่อมต่อกับอุปกรณ์ รับส่งโฆษณาผ่านบลูทูธแบบเคลื่อนที่			↔						
3. ทดสอบการทำงานและเก็บผลการทดลอง				←	→				
4. ทำการปรับปรุงและแก้ไขข้อผิดพลาดที่ตรวจพบในส่วนต่างๆของโครงงาน						←	→		
5. วิเคราะห์และสรุปผลการทดลอง								↔	
6. จัดทำรูปเล่มรายงาน									↔

1.6 ผลที่คาดว่าจะได้รับ

1.6.1 เข้าใจหลักการเขียนโปรแกรมส่งโฆษณาผ่านอุปกรณ์ไร้สายในระบบบลูทูธได้

1.6.2 ได้โปรแกรมส่งโฆษณาผ่านบลูทูธแบบอัตโนมัติ

1.7 งบประมาณของโครงงาน

1.7.1 ถ่ายเอกสารและค่าเข้าเล่มโครงงานฉบับสมบูรณ์	1,000	บาท
1.7.2 ค่าหนังสือ	1,000	บาท
1.7.3 อื่นๆ	1,000	บาท
รวมเป็นเงิน (สองพันบาทถ้วน)	3,000	บาท

หมายเหตุ : ถัวเฉลี่ยทุกรายการ

บทที่ 2
หลักการและทฤษฎีที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงหลักการและทฤษฎีต่างๆ ที่นำมาประยุกต์ใช้เพื่อทำโครงงานนี้ ซึ่งทฤษฎีที่ใช้ ได้แก่ โครงสร้างการทำงานของ Bluetooth Protocol Stack, ภาษาJava API for Bluetooth (JSR-82) with BlueCove library และเทคนิคการส่งข้อมูลแบบไร้สาย ซึ่งต้องศึกษาถึงรายละเอียดของหลักการและทฤษฎีที่นำมาใช้ให้เข้าใจ เพื่อที่จะนำไปประยุกต์ใช้ในโครงงานนี้ได้

2.1 บลูทูธ (Bluetooth)

บลูทูธคือ ระบบสื่อสารข้อมูลไร้สายแบบสองทางด้วยคลื่นความถี่สูงในย่าน 2.4GHz โดยใช้ช่วงความถี่2.400 ถึง 2.4835 GHz แบ่งออกเป็น 79 ช่องสัญญาณ แต่ละช่องสัญญาณมีแบนด์วิดธ์หรือความกว้าง 1 MHz และใช้ช่องสัญญาณนี้เพื่อรับส่งข้อมูลสลับช่องไปมา 1,600 ครั้งต่อวินาที รัศมีทำการของอุปกรณ์บลูทูธจะอยู่ที่ 5 ถึง 10 เมตร แต่สามารถขยายรัศมีทำการได้ 30 ถึง 100 เมตร โดยต้องต่อสายอากาศและวงจรขยายกำลังส่งเพิ่มเติม

อุปกรณ์บลูทูธมีระบบป้องกันข้อมูลโดยใช้การป้อนรหัสก่อนการเชื่อมต่อ และป้องกันการดักสัญญาณระหว่างสื่อสาร โดยระบบจะสลับช่องสัญญาณไปมา มีความสามารถในการเลือกเปลี่ยนความถี่ที่ใช้ในการติดต่อเองอัตโนมัติ โดยที่ไม่จำเป็นต้องเรียงตามหมายเลขช่อง ใช้พลังงานต่ำ

อัตราการถ่ายทอดข้อมูลของอุปกรณ์บลูทูธจะขึ้นอยู่กับโปรไฟล์หรือรูปแบบการทำงาน โดยมีอัตราการถ่ายทอดข้อมูลสูงสุด 1 เมกะบิตต่อวินาที แต่โดยทั่วไปจะใช้งานกันอยู่ที่ 721 กิโลบิตต่อวินาที

2.1.1 จุดเด่นของ Bluetooth

1. เป็นเทคโนโลยีที่ราคาไม่แพง
2. สัญญาณที่ใช้มีลักษณะเป็น omni-directional สามารถผ่านทะลุกำแพงและสิ่งกีดขวางได้ดี ไม่จำเป็นต้องอยู่ในแนวเดียวกัน
3. สามารถจัดการได้พร้อมกันทั้งข้อมูลและเสียง
4. ไม่จำเป็นต้องเซตการติดตั้งพิเศษสำหรับการเชื่อมต่อ แต่อุปกรณ์จะตรวจพบกันเองโดยอัตโนมัติ
5. ใช้ Frequency hopping ลดความเสี่ยงในการลักลอบดักข้อมูล
6. เป็นเทคโนโลยีที่ได้รับการยอมรับและนำมาใช้อย่างแพร่หลายในเกือบทุกชนิดของ

อุปกรณ์จำพวก Consumer Electronics devices ได้แก่ โทรศัพท์เคลื่อนที่, PDA, กล้องถ่ายรูป, วีดีโอเกมส์ ฯลฯ

2.1.2 จุดมุ่งหมายของ Bluetooth

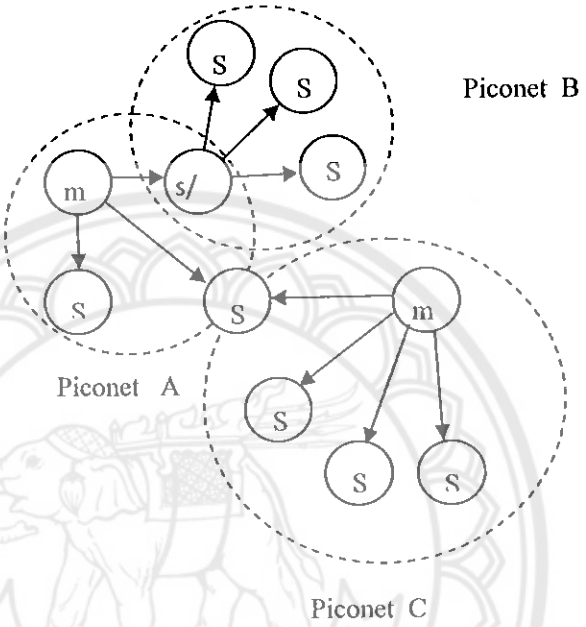
Bluetooth เป็นมาตรฐานเปิด (Open Specification) ที่ออกแบบมาเพื่อใช้สำหรับการส่งข้อมูลไร้สายระยะใกล้ (Short-Range Wireless) ถูกกำหนดขึ้นโดยกลุ่มบริษัทที่ให้ความสนใจเรื่องนี้เป็นพิเศษที่มีชื่อเรียกว่า Bluetooth Special Interest Group (Bluetooth SIG) โดยมีจุดมุ่งหมายในการวางมาตรฐานหลักๆดังนี้

1. เป็นมาตรฐานเปิด ทุกคนสามารถใช้ข้อมูลที่ส่งผ่าน Bluetooth นี้ได้โดยไม่ต้องเสียค่าธรรมเนียมใดๆทั้งสิ้น เพื่อให้เกิดความแพร่หลายในการใช้งานและมีการพัฒนาระบบได้อย่างรวดเร็วและกว้างขวาง
2. รับส่งข้อมูลแบบไร้สายในระยะใกล้ จุดมุ่งหมายหลักของ Bluetooth คือ การทำให้อุปกรณ์ต่างๆ สามารถส่งข้อมูลถึงกันได้โดยไม่ต้องใช้สาย โดยใช้คลื่นวิทยุเป็นตัวกลางในการส่งข้อมูลเพื่อให้อุปกรณ์ที่เชื่อมต่อถึงกันได้เกิดข้อจำกัดในเรื่องของตำแหน่งการวางอุปกรณ์และยังทำให้อุปกรณ์มากกว่า 2 ตัวสามารถเชื่อมต่อกันได้ในเวลาเดียวกัน และเนื่องจากเป็นการสื่อสารในระยะใกล้จึงใช้กำลังส่งไม่มากนัก ทำให้ประหยัดพลังงานพอที่จะนำไปใช้ในอุปกรณ์เล็กๆ เช่น โทรศัพท์มือถือ หรือ PDA ได้
3. สามารถรองรับการรับส่งเสียงและข้อมูลได้ในเวลาเดียวกัน มาตรฐานในขั้นต้นนี้ถูกกำหนดไว้ว่าระบบจะต้องมีความเร็วในการรับส่งข้อมูลเพียงพอสำหรับการส่งเสียงและตัวข้อมูลไปพร้อมๆกันได้ แต่ในอนาคตจะมีการเพิ่มเติมมาตรฐานเพื่อรองรับการส่งภาพพร้อมๆไปกับเสียงและข้อมูลได้
4. สามารถใช้งานได้ทุกที่ทั่วโลก กลุ่ม Bluetooth SIG ต้องการให้อุปกรณ์อะไรก็ตามที่ออกแบบถูกต้องตามมาตรฐาน Bluetooth สามารถใช้งานร่วมกันได้ไม่ว่าจะผลิตจากผู้ผลิตใดหรืออยู่ ณ ตำแหน่งใดบนโลก จากจุดมุ่งหมายข้อนี้ทำให้ต้องใช้ความถี่คลื่นวิทยุที่สามารถใช้งานได้ในทุกประเทศ

2.2 โครงสร้างการทำงาน

เทคโนโลยี Bluetooth ประกอบด้วย 3 ส่วนหลักคือ Radio technology, Profile และ Protocol stack โดย Bluetooth จัดเป็นโปรโตคอลสื่อสารชนิดหนึ่ง ไม่ว่าจะเป็น HTTP, FTP, SMTP เป็นต้น ใช้คลื่นวิทยุความถี่ 24 GHz Industrial-Scientific-Medical (ISM) band มีลักษณะโครงสร้างสถาปัตยกรรมในลักษณะของ client-server โดยอุปกรณ์ที่จะขอทำการเชื่อมต่อเข้าไปใช้ (initiate the communication) คือ master หรือ client ส่วนผู้ที่ได้รับการติดต่อขอเข้าไปใช้คือ slave หรือ server และเมื่ออุปกรณ์ Bluetooth-enabled devices เชื่อมต่อซึ่งกันและกันเรียกว่า piconet โดย

แต่ละ piconet สามารถเชื่อมต่อระหว่างกันได้ ซึ่ง 1 piconet จะประกอบด้วย 1 master จะติดต่อกับ single slave โดยใช้ point-to-point communication และอุปกรณ์หนึ่งใน piconet สามารถติดต่อสื่อสารกับอีกหนึ่งอุปกรณ์ใน piconet อื่นเพื่อจัดสร้างเป็นรูปแบบ scatternet นอกจากนี้ master ใน piconet หนึ่งอาจจะกลายเป็น slave ในอีก piconet หนึ่งก็ได้ ทั้งนี้เพื่อสร้าง bridge ขึ้นระหว่างกันดังรูปที่ 2.1



รูปที่ 2.1 Scatternet ที่ประกอบด้วย 3 Piconet

ในส่วนของ Profile จะเกี่ยวข้องกับ cross-platform interoperability และความต้องการของ consistency โดยเป็นตัวกำหนดบทบาทและความสามารถเฉพาะตัวสำหรับชนิดของแอพพลิเคชั่นในอุปกรณ์ที่ใช้ Bluetooth กล่าวคือ อุปกรณ์ที่ใช้ Bluetooth จะไม่สามารถติดต่อกันได้ถ้าไม่เป็นไปตาม Profile นั้น ถึงแม้จะมี Protocol stack ก็ไม่เพียงพอ และต่อไปนี้คือ Profile ที่สำคัญ (the - number of Profiles growing)

2.2.1 Generic Access Profile เป็น Profile ขั้นต่ำสุดที่อุปกรณ์ที่ทำงานด้วย Bluetooth จะต้องสนับสนุน มีหน้าที่กำหนดการเชื่อมต่อการค้นหา และการจัดการ รวมทั้งเรื่องความปลอดภัย

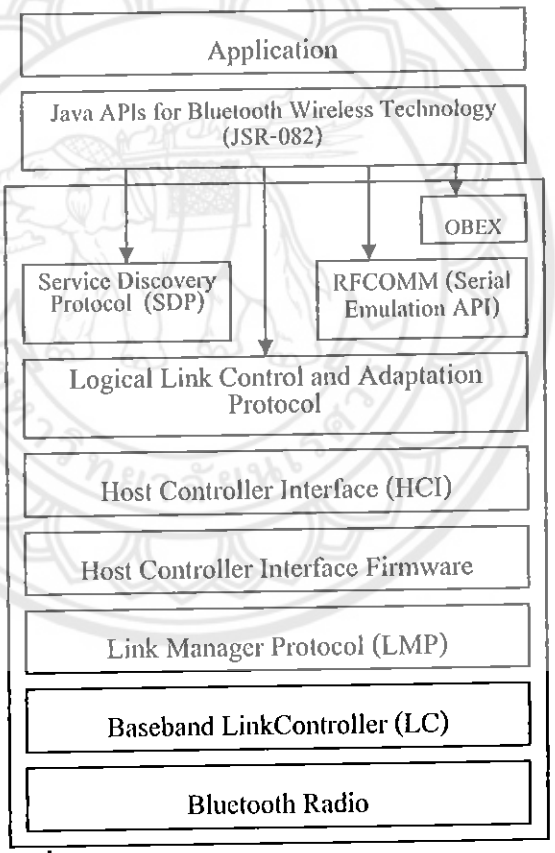
2.2.2 Services Discovery Application Profile กำหนดลักษณะและขั้นตอนสำหรับแอพพลิเคชั่นในอุปกรณ์ที่ทำงานด้วย Bluetooth ค้นหาเซอร์วิสที่ลงทะเบียนอยู่ในอุปกรณ์ที่มี Bluetooth อื่นเพื่อดึงข้อมูลที่ต้องการมาใช้

2.2.3 Serial Port Profile กำหนดความต้องการสำหรับอุปกรณ์ที่ใช้ Bluetooth ในการเซตค่าสำหรับการเชื่อมต่อ emulate serial cable และการใช้โปรโตคอล RFCOMM

2.2.4 LAN Access Profile กำหนดอุปกรณ์ที่ใช้ Bluetooth สามารถเข้าใช้เซอร์วิสของ LAN อย่างไรด้วย PPP และแสดงกลไกของ PPP ที่สามารถถูกใช้ในการจัดสร้างเน็ตเวิร์กที่ประกอบด้วยอุปกรณ์ที่ใช้ Bluetooth

2.2.5 Synchronization Profile กำหนดความต้องการของแอพพลิเคชั่นสำหรับอุปกรณ์ที่ใช้ Bluetooth ซึ่งจำเป็นต้องมีการปรับหรืออัปเดตข้อมูลระหว่างอุปกรณ์ให้ตรงกัน

สำหรับส่วน Protocol stack จะจัดเตรียมให้สำหรับ higher-level protocols และ APIs ต่างๆ ช่วยให้เราสามารถควบคุมอุปกรณ์ Bluetooth ด้วยตัวเอง และติดต่อสื่อสารกับอุปกรณ์ Bluetooth อื่น กล่าวคือ ถ้าอุปกรณ์ Bluetooth ปราศจาก Protocol stack ก็เสมือนอุปกรณ์ computer peripheral ปราศจากไดรเวอร์ ในรูปที่ 2.2 คือโครงสร้างสถาปัตยกรรมของ Protocol stack ที่ประกอบด้วย software stack (สำหรับทำงานบนจาวาแพลตฟอร์ม) อินเตอร์เฟสกับ firmware stack



รูปที่ 2.2 โครงสร้างของ Bluetooth Protocol stack

2.2.6 Radio layer คือ physical wireless connection เพื่อหลีกเลี่ยงการแทรกแซงรบกวนกับอุปกรณ์อื่น ซึ่งการสื่อสารใน ISM band การโมดูเลชันจะใช้ fast frequency hopping

2.2.7 Baseband layer

รับผิดชอบในการควบคุมและส่ง data packets ในส่วนของ radio link พร้อมทั้งจัดเตรียม transmission channels สำหรับทั้งข้อมูลและเสียง โดย Baseband layer จะดูแลในส่วนของ Synchronous Connection-Oriented (SCO) links สำหรับเสียงและ Asynchronous Connectionless (ACL) links สำหรับข้อมูลซึ่ง SCO packets จะไม่มีการ retransmitted แต่ ACL packets จะทำเพื่อให้มั่นใจว่าข้อมูลไม่เปลี่ยนแปลง โดย SCO links มีการเชื่อมต่อแบบ point-to-point

2.2.8 Host Controller Interface (HCI)

เป็นตัวแบ่งกั้นระหว่างซอฟต์แวร์และฮาร์ดแวร์ โดย L2CAP และเลเยอร์ที่อยู่เหนือขึ้นไปจัดให้อยู่ในส่วนของซอฟต์แวร์ ส่วน LMP และเลเยอร์ที่ต่ำกว่าจัดเป็นส่วนของฮาร์ดแวร์ กล่าวได้ว่า HCI คือ Driver interface สำหรับ physical bus ซึ่งเชื่อมทั้งสองส่วนนี้

2.2.9 Logical Link Control and Adaptation Protocol (L2CAP)

รับ application data และปรับเปลี่ยนให้เป็น Bluetooth format นอกจากนี้พารามิเตอร์ Quality ofServices (QoS) ถูกแลกเปลี่ยนที่เลเยอร์นี้ด้วย

Bluetooth ได้จัดเตรียมในเรื่องความปลอดภัยไว้ถึง 3 วิธีคือ

1. Pseudo-random frequency hopping ทำให้ยากในการดักจับข้อมูล
2. Authentication ให้ผู้ใช้สามารถจำกัดการเชื่อมต่อเฉพาะอุปกรณ์ที่ระบุไว้
3. Encryption การเข้ารหัสข้อมูลเฉพาะสำหรับผู้มีสิทธิ

ตามที่ได้กล่าวมาแล้วว่าอุปกรณ์ที่ทำงานด้วย Bluetooth จะต้องสนับสนุน Generic Access Profileซึ่งกำหนดรูปแบบความปลอดภัยไว้ถึง 3 โหมดด้วยกันคือ

Mode 1 เป็นโหมดที่การปฏิบัติงานไม่ต้องมีเรื่องของความปลอดภัย

Mode 2 เป็นโหมด service-level enforced security กล่าวคือเมื่ออุปกรณ์ปฏิบัติงานในโหมดนี้จะยังไม่เริ่มมีขั้นตอนความปลอดภัยก่อนที่ channel จะถูกสร้างขึ้น ซึ่งโหมดนี้จะช่วยทำให้แอพพลิเคชั่นมีนโยบายในการแอ็กเซสที่ต่างกัน และจะรันในลักษณะขนาน

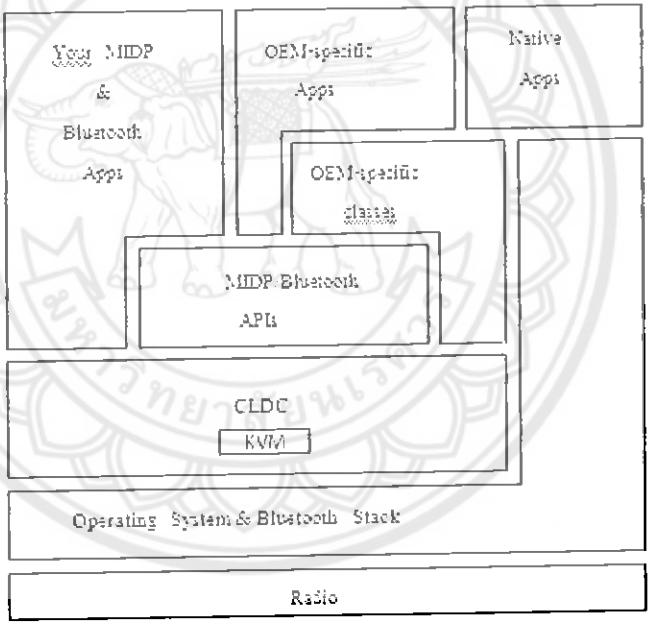
Mode 3 รู้จักกันในฐานะโหมด link-level enforced security ที่โหมดนี้ขั้นตอนเรื่องความปลอดภัยจะถูกกำหนดขึ้นก่อนการเซตค่าการเชื่อมต่อจะเสร็จสิ้น

2.3 Java Bluetooth API

ในขณะที่ฮาร์ดแวร์ของ Bluetooth ได้พัฒนาก้าวหน้าอย่างรวดเร็ว แต่การพัฒนาแอพพลิ-เคชั่นสำหรับ Bluetooth ยังคงช้าและไม่มีมาตรฐานที่แน่ชัด จนกระทั่งเกิด JSR 82 ขึ้นมา ซึ่งเป็นมาตรฐานเปิดไม่ขึ้นกับผู้ผลิตเฉพาะรายใด เป็นมาตรฐานสำหรับสร้างและพัฒนาแอพพลิเคชั่นเพื่อใช้ในอุปกรณ์ที่มี Bluetooth ด้วยภาษาวาจา โดยซ่อนความซับซ้อนต่างๆของ Bluetooth protocol

stack ให้อยู่เบื้องหลังเซตของ Java API แทน ทำให้ผู้พัฒนามุ่งความสนใจมาที่การสร้างและพัฒนาแอพพลิเคชั่น แทนที่จะเป็นรายละเอียดในส่วนระดับล่างของ Bluetooth

ตัว JSR 82 ประกอบด้วย 2 package optional ที่เป็นอิสระต่อกันคือ Core Bluetooth API และ Object Exchange (OBEX) API ซึ่งแนะนำโดย Infrared Data Association (IrDA) มีลักษณะเป็น high-level API และ transport-independent สำหรับการแลกเปลี่ยนออบเจ็กต์ เช่น electronic business cards ซึ่ง OBEX สามารถใช้ตามลำพังได้โดยไม่จำเป็นต้องใช้ร่วมกับ Core Bluetooth API โดย API ถูกออกแบบเพื่อให้นักพัฒนาจาวาสามารถสร้าง Bluetooth Profiles ขึ้นมาใหม่บน API นี้ ตราบเท่าที่ core layer specification ไม่มีการแลกเปลี่ยน และเพื่อให้มีความยืดหยุ่นในการใช้ specification ไม่ได้จำกัดต่อ API สำหรับ Bluetooth Profiles เท่านั้น JSR 82 ยังได้รวม APIs สำหรับ OBEX และ L2CAP เพื่อที่ว่า Bluetooth Profiles ในอนาคตสามารถสนับสนุนให้ใช้ในจาวาได้ สำหรับรูปที่ 2.3 แสดง API ที่ถูกให้โดย specification ตามโครงสร้างของ CLDC/MIDP

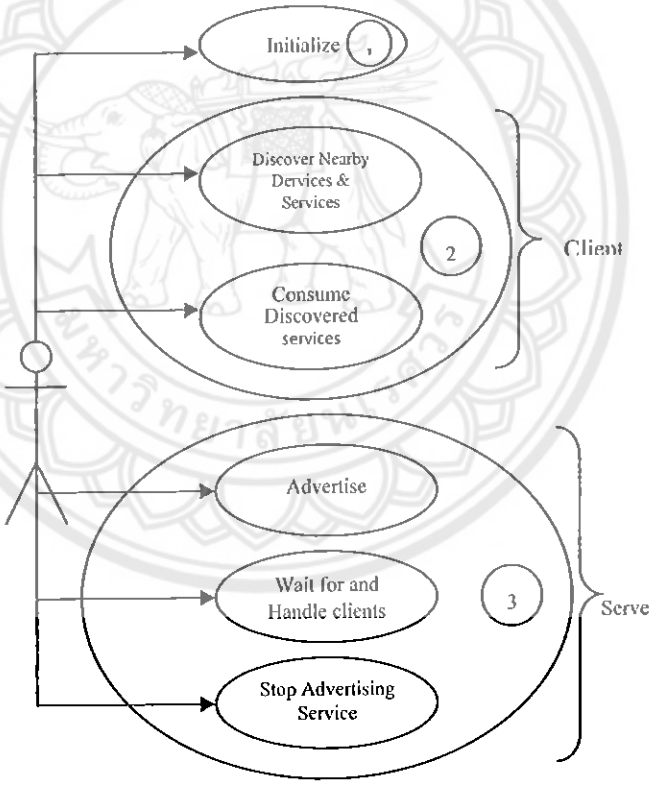


รูปที่ 2.3 โครงสร้างของ Bluetooth Protocol stack

JSR 82 API สามารถทำงานได้ทั้ง Native Bluetooth stacks และ Java Bluetooth stacks ซึ่งในกรณีของ Java Bluetooth stacks นั้น APIs จะเรียก stack โดยตรง ส่วนใน Native Bluetooth stacks จะเรียก APIs ผ่านทาง virtual machine ซึ่งอินเตอร์เฟสและที่ได้กล่าวมาแล้วว่านักพัฒนาจาวาสามารถเพิ่มขยาย Profiles ใหม่ได้ ซึ่ง JSR 82 ต้องการ Connected Device Configuration (CDC) ซึ่งเป็นซูเปอร์เซตของ CLDC และมี JABWT อยู่บนทั้งสอง CLDC และ CDC based profiles ซึ่งเราสามารถจะใช้ JABWT กับ J2ME Profile ใดๆ และถ้า Generic Connection Framework Optional

Package สำหรับ J2SE (JSR 197) ถูกสนับสนุนให้ใช้แล้ว JSR 82 APIs ควรจะทำงานร่วมกับ J2SE ได้อย่างสบาย โดย Java APIs for Bluetooth ได้กำหนดไว้ 2 แพ็กเกจที่ขึ้นอยู่กับ CLDC javax.microedition.io package คือ javax.bluetooth : core Bluetooth API และ javax.obex:APIs for the Object Exchange (OBEX) protocol

แอพพลิเคชั่นสำหรับอุปกรณ์ที่มี Bluetooth (Bluetooth-enabled application) สามารถที่จะระบุให้เป็นได้ทั้ง client หรือ server โดยแอพพลิเคชั่นดังกล่าวจะต้อง initialize ต่อ Bluetooth stack ก่อน และเมื่อ client จะใช้ remote services มันจะต้องค้นหาอุปกรณ์สำหรับเซอร์วิสที่น่าสนใจ โดย server จะให้เซอร์วิสที่สะดวกต่อ client ซึ่งลงทะเบียนใน Service Discovery Database (SDDB) คล้ายกับเป็นการโฆษณา หลังจากนั้นจึงรอคอยการเชื่อมต่อเข้ามาและยอมรับเมื่อมีการเชื่อมต่อจาก client ติดต่อเข้ามาขอใช้เซอร์วิสและเมื่อเซอร์วิสไม่ได้ใช้โดยแอพพลิเคชันก็จะถูกนำออกจาก SDDB ซึ่งกิจกรรมทั้งหมดได้แสดงไว้ในรูปที่ 2.4



รูปที่ 2.4 การใช้ Bluetooth-Enabled Application

กล่าวได้ว่าแอพพลิเคชั่นใดๆ สำหรับ Bluetooth จะประกอบด้วย 5 ส่วนคือ

1. Stack initialization
2. Device management
3. Device discovery
4. Services discovery
5. Communication

2.3.1 Stack initialization

รับผิดชอบสำหรับการควบคุมอุปกรณ์ Bluetooth device ดังนั้นเราจึงจำเป็นที่จะต้อง initialize ต่อ Bluetooth stack ก่อนที่จะทำงานอื่น ซึ่งขบวนการในการทำ initialization ประกอบด้วยหลายขั้นตอน โดยมีวัตถุประสงค์เพื่อให้อุปกรณ์พร้อมใช้งาน แต่โชคไม่ดีที่ Bluetooth specification ปล่อยให้ถูกกำหนดโดยผู้ผลิต จึงทำให้แต่ละผู้ผลิตจัดการ stack initialization ที่แตกต่างกัน

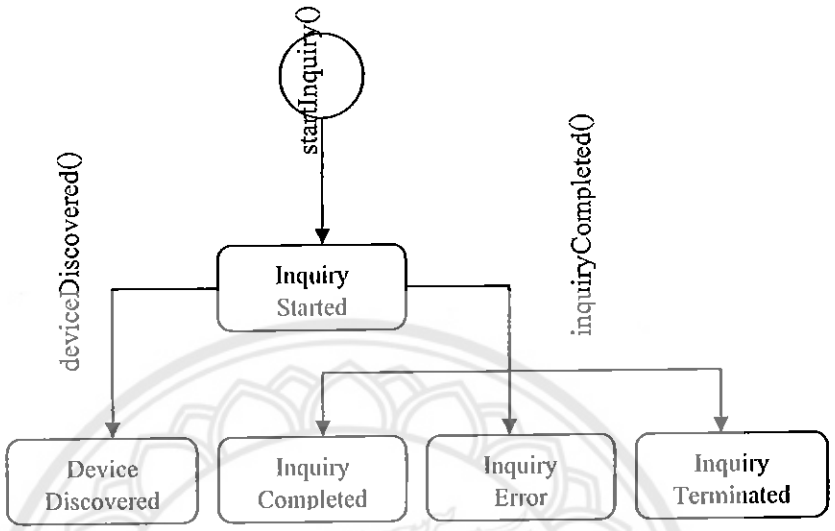
Java Bluetooth APIs มีคลาส LocalDevice และ RemoteDevice ซึ่งได้ให้ความสามารถของ device-management ที่กำหนดอยู่ใน Generic Access โดยคลาส LocalDevice จะขึ้นอยู่กับคลาส javax.bluetooth.DeviceClass โดยเป็นตัวแทนของอุปกรณ์ local Bluetooth ที่ให้เมธอดสำหรับดึงข้อมูลชนิดของอุปกรณ์และเซอร์วิสที่เสนอของ local device ได้แก่ getBluetoothAddress(), getDeviceClass() , getFriendlyName() , getRecord() , updateRecord() , getDiscoverable() , setDiscoverable(), getDiscoveryAgent() และ getProperty() ส่วนคลาส RemoteDivice จะแทน remote device (a device within a range of reach) และจัดหาเมธอดสำหรับดึงข้อมูลที่เกี่ยวข้องกับอุปกรณ์ได้แก่getBluetoothAddress(),getFrindlyName(),getRemoteDevice(),authenticate(), authorize(),encrypt(), isAuthenticated(),isAuthorized(), isEncrypted() และ isTrustedDevice ()

นอกจากนี้คลาส RemoteDevice ยังคงจัดเตรียมเมธอดให้สำหรับ authentication ,authorize หรือ encrypt ข้อมูลที่ส่งผ่านระหว่าง local และ remote devices

2.3.2 Device discovery

เนื่องจากอุปกรณ์ไร้สายมีลักษณะพกพาเคลื่อนที่ จึงต้องมีกลไลที่แต่ละอุปกรณ์สามารถตรวจค้นซึ่งกันและกันได้ ซึ่ง core Bluetooth API ให้คลาสและอินเตอร์เฟส DiscoveryAgent และ DiscoveryListener สำหรับdiscovery services ที่จำเป็น โดยอุปกรณ์ Bluetooth สามารถที่จะใช้ออบเจ็กต์ของ DiscoveryAgent เพื่อได้รับรายชื่อของอุปกรณ์ที่สามารถแอ็กเซสได้ ซึ่งมี 3 วิธีคือ DiscoveryAgent.startInquiry method จะวางอุปกรณ์เข้าไปใน inquiry mode และเพื่อที่จะใช้ประโยชน์ จากโหมดนี้แอพพลิเคชันจะต้องระบุ event listener ซึ่งจะตอบสนองต่อ Inqiryrelated events โดย DiscoverListener.devicesDiscovered จะถูกเรียกในแต่ละเวลาที่ Inquiry พบอุปกรณ์

และเมื่อ inquiry เสร็จสิ้นลงหรือถูกยกเลิก DiscoveryListener.inquiryCompleted จะถูกกระตุ้น ซึ่งมีไดอะแกรมดังแสดงในรูปที่ 2.5



รูปที่ 2.5 ไดอะแกรม Device discovery

แต่ถ้าอุปกรณ์ไม่ต้องการที่จะคอยให้ตรวจพบก็สามารถที่จะใช้DiscoveryAgent.retrieve-Devices method ดึงรายชื่อที่มีอยู่แล้ว ซึ่งขึ้นกับพารามิเตอร์ที่ผ่านมาด้วย โดยเมธอดนี้จะให้ค่ารายชื่อของอุปกรณ์ซึ่งถูกพบโดย inquiry ก่อนหน้านี้ หรือค่ารายชื่อของอุปกรณ์ที่มีอยู่แล้ว ซึ่ง local device ได้ให้แก่ Bluetooth Control Center ที่ติดต่อกันบ่อยๆ

2.3.3 Service discovery

ทันทีที่ local device ได้ถูกตรวจพบอย่างน้อยหนึ่ง remote device มันจะเริ่มค้นหาเซอร์วิสที่สะดวก ซึ่งก็คือแอพพลิเคชัน Bluetooth สำหรับใช้ในงานที่เราต้องการ แต่เนื่องจากการค้นหาเซอร์วิส (Service discovery) มีลักษณะคล้ายกับการค้นหาอุปกรณ์ (Device discovery) ดังนั้น DiscoveryAgent ยังคงให้เมธอดสำหรับใช้ในงานดังกล่าว เพื่อค้นหาเซอร์วิสจากอุปกรณ์ Bluetooth server และ initiate service discovery transactions ได้แก่ เมธอด selectService() และ searchServices() สำหรับ initiates service discovery เมธอด cancelServiceSearch() สำหรับ cancels ต่อ Service discovery รวมทั้งเมธอด servicesDiscovered() ใช้เพื่อแสดงว่าเซอร์วิสที่ถูกค้นพบแล้ว และเมธอด serviceSearchCompleted() ใช้เพื่อแสดงว่า Service discovery ได้เสร็จสิ้นแล้ว

ข้อสังเกต API จะจัดเตรียมกลไกสำหรับค้นหาเซอร์วิสของ remote devices แต่จะไม่ได้ให้สำหรับเซอร์วิสบน local device และก่อนที่เซอร์วิสจะสามารถให้ค้นหาใช้งานได้ จะต้องมีการ

ลงทะเบียนต่ออุปกรณ์ Bluetooth server เสียก่อน กล่าวคือ เซิร์ฟเวอร์จะมีหน้าที่รับผิดชอบหน้าที่ดังต่อไปนี้

1. สร้าง service record สำหรับรายละเอียดที่เซอร์วิสได้นำเสนอ
2. เพิ่ม service record ไปที่ Service Discovery DataBase (SDDB) ของเซิร์ฟเวอร์ เพื่อที่ว่าไคลเอนต์จะได้รับทราบ
3. จัดทำค่าในส่วนของความปลอดภัยที่เกี่ยวกับเซอร์วิส
4. ยอมรับการเชื่อมต่อจากไคลเอนต์
5. อัปเดต service record ใน SDDB เมื่อใดก็ตามที่แอตทริบิวส์ของเซอร์วิสมีการเปลี่ยนแปลง
6. หยุดการทำงานหรือการนำ service record ใน SDDB ออกเมื่อเซอร์วิสไม่ได้ใช้

2.3.4 Communication

สำหรับ local device ที่จะใช้เซอร์วิสของ remote device ทั้งสอง จะต้องใช้โปรโตคอลสื่อสารร่วมกัน เพื่อแอพพลิเคชันสามารถแอ็กเซสได้อย่างกว้างขวางในหลายเซอร์วิส Java APIs for Bluetooth จะต้องให้กลไกที่ยอมให้มีการเชื่อมต่อแก่เซอร์วิสใดๆ ที่ใช้โปรโตคอล RFCOMM , L2CAP (low-level protocol for managing data packets up to 64 kilobytes long) หรือ OBEX แต่ถ้าเซอร์วิสเปลี่ยนไปใช้โปรโตคอลอื่น เช่น TCP/IP เลเยอร์ที่อยู่เหนือโปรโตคอลเหล่านี้ แอพพลิเคชันสามารถที่จะแอ็กเซสเซอร์วิสเมื่อได้รับการสนับสนุนการติดตั้งโปรโตคอลเพิ่มเติมในแอพพลิเคชันโดยใช้ CLDC Generic Connection Framework และเพราะว่าโปรโตคอล OBEX สามารถใช้ในหลายสื่อตัวกลาง ไม่ว่าจะเป็น wired,infrared,Bluetooth radio และอื่นๆ ดังนั้น JSR 82 จะต้องสนับสนุนให้ OBEX API (javax.obex) สามารถทำงานได้อย่างอิสระจาก core Bluetooth API (javax.bluetooth) สำหรับในส่วนของโปรโตคอล RFCOMM ซึ่งเป็นเลเยอร์ที่อยู่บนโปรโตคอล L2CAP emulates การเชื่อมต่อแบบ RS-232 serial โดย Serial Port Profile (SPP) จะทำให้การติดต่อสื่อสารระหว่างอุปกรณ์ Bluetooth ง่ายขึ้น โดยให้ stream-based interface เชื่อมต่อกับโปรโตคอล RFCOMM ซึ่งมีความสามารถและข้อจำกัดดังนี้

1. ถ้ามี 2 อุปกรณ์สามารถใช้ได้เพียง 1 RFCOMM session รวมกันเท่านั้น ณ เวลาหนึ่ง
2. ในจำนวนมากถึง 60 logical serial connections สามารถถูก multiplexed ได้ในเซสชันนี้
3. อุปกรณ์ Bluetooth เพียง 1 ชนิดสามารถมีมากถึง 30 active RFCOMM services
4. ใน 1 อุปกรณ์สามารถสนับสนุนเพียง 1 การเชื่อมจากไคลเอนต์ต่อเซอร์วิสใดได้ ณ เวลาหนึ่งความต้องการของจาวาบน Bluetooth
5. หน่วยความจำรวมขั้นต่ำ 512 กิโลไบต์ (ทั้ง ROM และ RAM) สำหรับความต้องการเรื่องหน่วยความจำในส่วนของแอพพลิเคชันนั้นเป็นส่วนที่เพิ่มเติมต่างหาก
6. มีการเชื่อมต่อเข้ากับ Bluetooth wireless network

7. J2ME Connected Limited Device Configuration (CLDC)

2.4 เทคนิคการส่งข้อมูล

เมื่อได้ข้อตกลงด้านความถี่ที่ใช้งานจำนวนช่องสัญญาณ และแบนด์วิดธ์ของแบนด์วิดธ์ของแต่ละช่องสัญญาณมาเป็นกรอบแล้ว ต่อไปก็คือการเลือกวิธีมอดูเลตข้อมูลเข้าไปกับคลื่นพาหะ เพราะการมอดูเลตแต่ละแบบจะส่งผลต่อความเร็วในการส่งข้อมูลด้วยและเนื่องจากความจำกัดในด้านแบนด์วิดธ์ของข้อมูลที่กว้างเพียง 1 เมกะเฮิร์ตต่อช่องสัญญาณ บวกกับความต้องการความเร็วในการส่งข้อมูลที่สูงที่สุด Bluetooth จึงได้เลือกใช้การมอดูเลตแบบ Gaussian Frequency-Shift Keying (GFSK)

การมอดูเลตด้วยวิธีนี้สามารถส่งข้อมูลได้ 1 บิตต่อความถี่คลื่นพาหะ 1 เฮิรตซ์ นั่นหมายความว่าแต่ละช่องสัญญาณสามารถส่งข้อมูลได้ที่ความเร็ว 1 เมกะบิตต่อวินาที (Mbit/s) โดยถ้าบิตข้อมูลเป็น '1' จะเกิดการเปลี่ยนแปลงความถี่ทางบวกจากความถี่พาหะ ในขณะที่บิตข้อมูล '0' จะทำให้เกิดความเปลี่ยนแปลงความถี่ทางลบจากความถี่พาหะ

การรับส่งข้อมูลจะแบ่งข้อมูลออกเป็นแพ็กเก็ตย่อยๆ แล้วส่งในแบบฮาร์ฟดูเพล็กซ์ (Half Duplex) เพื่อประหยัดช่องสัญญาณ (มิฉะนั้นต้องใช้สองช่องสัญญาณเพื่อส่งและรับข้อมูลได้พร้อมๆกัน) จังหวะการรับส่งข้อมูลทั้งหมดกำหนดโดยอุปกรณ์ที่เป็นมาสเตอร์ในลักษณะของการโพล ซึ่งอุปกรณ์ที่เป็นสเลฟจะต้องตอบกลับมายังมาสเตอร์ในทุกๆแพ็กเก็ตเพื่อให้มาสเตอร์รู้ว่ายังคงสามารถติดต่อกับสเลฟอยู่ได้

เมื่อมีการแบ่งข้อมูลออกเป็นแพ็กเก็ต ทำให้แต่ละแพ็กเก็ตต้องมีข้อมูลส่วนหัว (Header) เพิ่มเข้ามาเพื่อให้ทางฝั่งรับสามารถประกอบข้อมูลทั้งหมดเข้าด้วยกันได้อย่างถูกต้อง นอกจากนั้นก่อนการส่งแต่ละครั้งจะต้องมีการส่งข้อมูลเพื่อทำการซิงโครไนซ์สัญญาณนาฬิกาทางฝั่งส่งและรับให้เท่ากันเพื่อให้รับ-ส่งข้อมูลกันได้อย่างถูกต้อง ซึ่งเมื่อรวมปริมาณข้อมูลทั้งหมดที่จำเป็นต้องส่งในแต่ละครั้งจะทำให้ความเร็วในการส่งข้อมูลจริงลดลงจาก 1 เมกะบิตต่อวินาที เหลือ 732.2 กิโลบิตต่อวินาทีในทิศทางหนึ่งและ 57.6 กิโลบิตต่อวินาทีในอีกทิศทางหนึ่ง

นอกจากความเร็วและความสะดวกสบายแล้ว สิ่งที่จำเป็นอย่างยิ่งสำหรับการสื่อสารข้อมูลในปัจจุบันก็คือ ความปลอดภัยของข้อมูล โดยเฉพาะอุปกรณ์ Bluetooth ที่สามารถทำงานได้ในทุกที่ยิ่งมีความจำเป็นที่จะต้องมีการรักษาความปลอดภัยของข้อมูลเป็นอย่างดี เทคนิคการส่งข้อมูลที่ Bluetooth ใช้คือ เทคนิคการกระโดดข้ามทางความถี่ (Frequency Hopping Spread Spectrum : FHSS)

2.4.1 Frequency Hopping

เทคนิค FHSS ที่ Bluetooth ใช้นี้จะแบ่งข้อมูลที่ต้องการส่งออกเป็นแพ็กเก็ต การส่งข้อมูลในแพ็กเก็ตแรกจะเลือกความถี่ของช่องสัญญาณช่องหนึ่งสำหรับการส่ง หลังจากส่งเสร็จสิ้น

ก็จะกระโดดไปเลือกใช้ช่องสัญญาณความถี่อื่นในการส่งแพ็กเก็ตที่สอง และจะกระโดดไปใช้ความถี่อื่นเรื่อยๆ ตลอดช่วงความถี่ที่สามารถใช้งานได้ การกระโดดไปใช้งานช่องความถี่ต่างๆ นี้เรียกว่า Hopping จุดเด่นของการใช้เทคนิคนี้ในการส่งสัญญาณมีอยู่ 2 ข้อคือ

1. เกิดการชนกันของการเลือกใช้ช่องสัญญาณน้อย เนื่องจากช่วงความถี่ ISM ที่ Bluetooth ใช้นั้นเป็นช่วงความถี่ที่ไม่ต้องขออนุญาต ทำให้มีอุปกรณ์หลายชนิดที่ใช้ความถี่ช่วงนี้อยู่ ประกอบกับรูปแบบการใช้งานอุปกรณ์ Bluetooth ส่วนใหญ่อยู่ในลักษณะที่สามารถเคลื่อนที่ไปใช้งานที่ตำแหน่งใดๆ ก็ได้ ดังนั้นจึงมีความเสี่ยงที่จะเกิดการใช่ช่องสัญญาณใดๆสำหรับส่งสัญญาณแบบไม่เปลี่ยนช่อง แต่ถ้าใช้เทคนิค FHSS โดยที่กำหนดช่วงเวลาในการจับช่องสัญญาณของการส่งข้อมูลแต่ละครั้งให้สั้นก็จะทำให้โอกาสที่จะเกิดการใช้งานช่องความถี่เดียวกันลดลง และถึงแม้จะเกิดการชนกันของข้อมูลขึ้นก็จะเสียข้อมูลไปเพียงแพ็กเก็ตเดียว เมื่อส่งข้อมูลซ้ำในครั้งถัดไป (Retransmit)ก็จะเปลี่ยนไปใช้ความถี่อื่นซึ่งโอกาสที่จะไปใช้ช่องความถี่ซ้ำกันอีกมีได้น้อยมากเพราะรูปแบบในการกระโดดของอุปกรณ์แต่ละตัวจะไม่เหมือนกัน ความเร็วในการกระโดดที่ Bluetooth กำหนดไว้อยู่ที่ 1,600 ครั้งต่อวินาที
2. มีความปลอดภัยของข้อมูลสูง อุปกรณ์ที่เป็นมาสเตอร์จะคอยควบคุมจังหวะการรับส่งข้อมูลทั้งหมด นั่นหมายความรวมถึงรูปแบบการกระโดดเปลี่ยนช่องสัญญาณด้วย โดยรูปแบบการกระโดดนี้จะกำหนดจากแอดเดรสของอุปกรณ์ที่เป็นมาสเตอร์ ซึ่งแอดเดรสนี้จะไม่มีทางซ้ำกันเลยในอุปกรณ์ทุกๆตัว นั่นหมายความว่าอุปกรณ์ที่เป็นสเลฟเท่านั้นที่จะรู้แอดเดรสของมาสเตอร์เพื่อไปคำนวณรูปแบบการกระโดดที่ถูกต้องเพื่อรับข้อมูลทีละแพ็กเก็ตในลำดับที่ถูกต้อง แล้วประมวลผลขึ้นใหม่ให้เหมือนกับข้อมูลที่ส่งมา

ตารางที่ 2.1 การแบ่งคลาสของอุปกรณ์ตามกำลังส่ง

คลาส	กำลังส่งสูงสุด	กำลังส่งต่ำสุด	ระยะใช้งาน (โดยประมาณ)
1	100 mW (20 dBm)	1 mW (0 dBm)	100 เมตร
2	2.5 mW (4 dBm)	0.25 mW (-6 dBm)	10-20 เมตร
3	1 mW (0 dBm)		5-10 เมตร

ตารางที่ 2.2 สรุปข้อมูลด้านต่างๆของเทคโนโลยี Bluetooth

ช่วงความถี่	ISM Band 2.4000 – 2.4835 GHz	ยกเว้นฝรั่งเศสที่อยู่ในช่วง 2.4465 – 2.4835 GHz
วิธีการมอดูเลต	Gaussian Frequency – Shift Keying (GFSK)	BT Product = 0.5, Modulation Index = 0.28 – 0.38
ความเร็วในการส่งข้อมูล	1 Mbit/sec	การมอดูเลตด้วยวิธี GFSK ทำให้สามารถส่งข้อมูลได้ 1 บิตต่อความถี่พาหะ 1 เฮิรตซ์
ความเร็วที่ใช้ส่งข้อมูลได้จริง	723.2 kbit/sec ในทิศทางหนึ่งและ 57.6 kbit/sec ในอีกทิศทางหนึ่ง	เกิดจากความเร็วทั้งหมด 1 Mbit/sec แล้วตัดโอเวอร์เฮดต่างๆที่ต้องใช้ในการสื่อสาร
ความถี่ในการกระโดดเปลี่ยนช่อง	1,600 ครั้งต่อวินาที	มีเวลา 625 us ต่อการกระโดด 1 ครั้ง
ความไวของอุปกรณ์ตัวรับ	ต้องมีค่า Bit Error Rate (BER) ที่ดีกว่า 0.1 % ที่ระดับความแรงของสัญญาณอินพุต – 70 dBm หรือน้อยกว่า	ความไวที่ -70 dBm นี้รับสัญญาณจากตัวส่งของอุปกรณ์ใดๆที่ตรงตามมาตรฐาน Bluetooth
ความแรงของสัญญาณที่ส่ง	แบ่งเป็น 3 คลาส คือ 0 dBm, 4 dBm และ 20 dBm	ระยะทำงานขึ้นอยู่กับความแรงของสัญญาณ โดย คลาส 3 มีระยะทำงานอยู่ในช่วง 5-10 เมตร และ คลาส 1 ใช้ได้ไกลสุด 100 เมตร

2.5 ภาษา Java

ภาษา Java ถูกคิดค้นขึ้นมาในปี 1991 โดย James Gosling จากบริษัท SUN microsystem เดิมทีใช้ชื่อว่าภาษา Oak ซึ่งตั้งตามต้นไม้ใหญ่ในสวนของบ้าน ที่ใช้เป็นที่ทำงาน ต่อมาได้เปลี่ยนชื่อมาเป็น java ตามชนิดกาแฟที่ทีมวิศวกรผู้พัฒนาดื่มกันเป็นประจำ จึงมีสัญลักษณ์เป็นรูปถ้วยกาแฟร้อน และได้ประกาศอย่างเป็นทางการเมื่อเดือนพฤษภาคม 1995

ภาษาจาวา นั้นสามารถรันได้บนทุกแพลตฟอร์ม จึงได้รับความนิยมอย่างมากในปัจจุบัน คุณสมบัติอันโดดเด่นนี้ต่างจากภาษาอื่นๆ เพราะ จาวาไม่ได้รันบนระบบปฏิบัติการ แต่รันบน Java Platform หรือ Java Virtual Machine อันเป็นตัวกลางติดต่อระหว่าง ตัวโปรแกรมที่เขียนจากจาวากับระบบปฏิบัติการ ไม่ว่าจะนำโปรแกรมไปรันบนระบบใด ก็จะไม่มีความแตกต่าง ตรงกับประโยคที่นิยมกล่าวกันว่า Write once Run anywhere เพราะเราสามารถนำไปรันได้ไม่ว่าจะทั้งเครื่องพีซี โทรศัพท์มือถือ สมาร์ทการ์ด ปาล์ม พีดีเอ ฯลฯ ซึ่งในอนาคต คาดการณ์กันว่าจะเป็นภาษาที่ได้รับความนิยมมากที่สุด นับว่าเป็นภาษาที่น่าเรียนรู้ไว้มากที่สุด

ภาษาจาวา เป็นภาษา Object-Oriented Programming สมบูรณ์แบบ ที่มีคำสั่งพื้นฐานคล้ายกับภาษา C++ ดังนั้นจาวาจึงทำงานด้วยหลักการพื้นฐานของ object oriented ดังนี้

2.5.1 Encapsulation

เป็นกระบวนการซ่อนรายละเอียดการทำงานและข้อมูลไว้ภายใน ไม่ให้ภายนอกสามารถมองเห็นได้ และเมื่อภายนอกมองไม่เห็นสิ่งที่ถูกซ่อนไว้ภายในแล้ว ก็จะไม่สามารถทำการเปลี่ยนแปลง แก้ไข หรือสร้างความเสียหายให้กับสิ่งต่างๆที่อยู่ภายในได้

หากจะเปรียบเทียบหลักการของ encapsulation แล้ว ก็เหมือนกับการซ่อนกระบวนการทำงานและข้อมูลไว้หลังกำแพง ซึ่งสิ่งที่อยู่ด้านนอกของกำแพงจะไม่สามารถเปลี่ยนแปลงกระบวนการทำงานหรือเข้าถึงข้อมูลที่อยู่หลังกำแพงได้

ข้อดีของ encapsulation คือ สามารถสร้างความปลอดภัยให้กับข้อมูลได้ เนื่องจากข้อมูลจะถูกเข้าถึงได้จากผู้ที่มีสิทธิ์เท่านั้น

2.5.2 Inheritance

หากสิ่งหนึ่งมีลักษณะคล้ายกับอีกสิ่งหนึ่งมากๆจะมีส่วนที่แตกต่างกันก็เพียงเล็กน้อยเท่านั้น เราก็ไม่จำเป็นต้องสร้างสิ่งนั้นขึ้นมาใหม่ทั้งหมด แต่สามารถนำหลักการของ object oriented ที่เรียกว่า inheritance มาใช้ได้

หลักการของ inheritance คือ ทำการสร้างสิ่งใหม่ขึ้นด้วยการสืบทอดหรือรับเอา (inherit) คุณสมบัติบางอย่างมาจากสิ่งเดิมที่มีอยู่แล้ว คือ ทำการสร้างเพิ่มเติมจากสิ่งที่มีอยู่ได้เลย

ข้อดีของ inheritance คือ จากการที่สามารถนำสิ่งที่เคยสร้างขึ้นแล้วกลับมาใช้ใหม่ (re-use) ได้ ทำให้ช่วยประหยัดเวลาการทำงานลงไปได้มาก เนื่องจากไม่ต้องเสียเวลาพัฒนาใหม่หมด

2.5.3 Polymorphism

การที่เมธอดชื่อเดียวกัน สามารถรับอาร์กิวเมนต์ที่แตกต่างกันได้หลายรูปแบบนี้ มีศัพท์เรียกเฉพาะว่า เมธอดชื่อนั้นถูกโอเวอร์โหลด (overload) สมมุติว่าเราใช้เมธอด calculate ในการคำนวณอะไรสักอย่าง เราสามารถใช้ประโยชน์จาก polymorphism ได้ อย่างเช่น หากผู้ใช้งานโปรแกรมส่งข้อมูลเข้ามาเป็นเลขจำนวนเต็มเพราะไม่ต้องการผลการคำนวณที่ละเอียดมากนัก เราก็

สามารถเรียกเมธอด public void calculate(x) ให้ทำงานได้ แต่หากผู้ใช้งานโปรแกรมส่งข้อมูลเข้ามาเป็นเลขจำนวนจริง เราก็สามารถเรียกเมธอดอีกตัวหนึ่ง คือ public void calculate(double y) ที่ทำการคำนวณได้ละเอียดกว่า ให้ทำงานได้ ทั้งนี้ทั้งนั้นขึ้นอยู่กับความต้องการในขณะหนึ่งๆว่าต้องการให้เมธอด calculate ทำงานในรูปแบบใด

สรุปแล้วข้อดีของ polymorphism ก็คือ การทำให้สิ่งหนึ่งสามารถทำงานได้หลากหลายรูปแบบตามความต้องการที่เกิดขึ้นในขณะหนึ่งๆ

2.6 Introduction Java

2.6.1 ชนิดข้อมูลพื้นฐาน (Primitive Data Type)

ชนิดข้อมูลพื้นฐานในภาษาจาวาแบ่งออกได้เป็น 4 กลุ่ม ดังนี้

1. Logical ได้แก่ Boolean
2. Textual ได้แก่ char
3. Integral ได้แก่ byte, short, int และ long
4. Floating-point ได้แก่ float และ double

2.6.1.1 ชนิดข้อมูลในกลุ่ม Logical – ชนิดข้อมูล Boolean

ชนิดข้อมูล Boolean สามารถมีค่าได้เพียง 2 ค่า คือ true (จริง) และ false (เท็จ)

2.6.1.2 ชนิดข้อมูลในกลุ่ม Textual – ชนิดข้อมูล char และ String

1. char เป็นชนิดข้อมูลแบบตัวอักษร (character) มีขนาด 16 บิต ใช้สำหรับเก็บข้อมูลที่มีความยาวไม่ เกิน 1 ตัวอักษรเท่านั้น โดยการเขียน character literals จะต้องคลอบคลุมด้วยเครื่องหมาย ''(single)

2. String ความจริงแล้ว String ไม่ได้เป็น primitive data type อย่างเช่นชนิดข้อมูล char แต่ String เป็นคลาสที่สามารถเก็บข้อมูลได้มากกว่า 1 ตัวอักษร โดยการเขียน string literals จะต้องคลุมด้วยเครื่องหมาย " " (double quote)

2.6.1.3 ชนิดข้อมูลในกลุ่ม Integral – ชนิดข้อมูล byte, short, int และ long

ชนิดข้อมูลแบบ integral จะมีชนิดข้อมูลโดยปริยายเป็น int กล่าวคือ หากกำหนดเลขจำนวนเต็มขึ้นมาจำนวนหนึ่ง เลขจำนวนเต็มนั้นจะถูกกำหนดให้มีชนิดข้อมูลเป็น int ซึ่งหากต้องการให้เลขจำนวนเต็มนั้นถูกกำหนดชนิดข้อมูลเป็นอย่างอื่น เช่น long ก็ต้องระบุ L หรือ l ต่อท้ายเลขจำนวนเต็มนั้น เป็นต้น

2.6.1.4 ชนิดข้อมูลในกลุ่ม Floating-point – ชนิดข้อมูล float และ double

ชนิดข้อมูลแบบ floating-point จะมีชนิดข้อมูลเป็น double กล่าวคือ หากกำหนดเลขจำนวนจริงขึ้นมาจำนวนหนึ่ง เลขจำนวนจริงนั้นจะถูกกำหนดให้มีชนิดข้อมูลเป็น double ซึ่งหาก

ต้องการให้เลขจำนวนจริงนั้นถูกกำหนดชนิดข้อมูลเป็นอย่างอื่น เช่น float ก็ต้องระบุ F หรือต่อท้ายเลขจำนวนจริงนั้น

2.6.2 การดำเนินการทางคณิตศาสตร์

ตัวอย่างการดำเนินการทางคณิตศาสตร์ เช่น a + b * c

นิพจน์ (Expression) คือการนำเอาโอเปอเรเตอร์และโอเปอแรนด์หลายๆตัวมารวมเข้าเป็นประโยคเดียวกัน ซึ่งจากด้านบนนั้นเราจะเรียก a + b * c ว่า " นิพจน์ "

โอเปอเรเตอร์(Operater) คือ ตัวดำเนินการ ซึ่งอาจเป็นการดำเนินการทางคณิตศาสตร์หรือทางตรรกศาสตร์ก็ได้ ซึ่งจากนิพจน์ด้านบน เราจะเรียก + และ * ว่า "โอเปอเรเตอร์"

โอเปอแรนด์ (Operand) คือ ตัวถูกดำเนินการ อาจเป็น literals, ตัวแปร หรือ นิพจน์ก็ได้ ซึ่งจากนิพจน์ด้านบน เราจะเรียก a,b และ c ว่า "โอเปอแรนด์"

การคำนวณค่าของนิพจน์จะเริ่มจากโอเปอเรเตอร์ที่มีลำดับความสำคัญสูงกว่าก่อน ตารางนี้แสดงลำดับความสำคัญของโอเปอเรเตอร์จากสูงสุดไปต่ำสุด

ตารางที่ 2.3 แสดงลำดับความสำคัญของโอเปอเรเตอร์ทางคณิตศาสตร์

โอเปอเรเตอร์	ลำดับความสำคัญ
()	สูงสุด
-	
(เครื่องหมายลบหน้าตัวเลข ซึ่งเป็น unary operator คือ โอเปอเรเตอร์ที่ต้องการโอเปอเรเตอร์เพียงตัวเดียว)	
*, / (หาร), % (หารเอาเศษ)	
+,-	ต่ำสุด

2.6.3 การแปลงชนิดของข้อมูล

เมื่อต้องการดำเนินการทางคณิตศาสตร์ระหว่างโอเปอแรนด์ที่มีชนิดข้อมูลที่แตกต่างกัน เราจะต้องแปลงชนิดของข้อมูลของโอเปอแรนด์ให้เป็นชนิดเดียวกันก่อน จึงจะสามารถดำเนินการกันได้ โดยแบ่งการแปลงชนิดข้อมูลออกได้เป็น 2 ประเภท คือ

1. Implicit type conversion เป็นการแปลงชนิดข้อมูลที่ภาษาจาวาทำให้โดยอัตโนมัติตามความเหมาะสม

2. Explicit type conversion (หรือ Casting) เป็นการแปลงชนิดข้อมูลที่ผู้เขียนโปรแกรมต้องทำด้วยตัวเองการทำ exclicit type conversion นั้น โดยปกติจะแปลงจากชนิดข้อมูลที่มีนัยสำคัญ

(significant)ต่ำกว่าไปเป็นชนิดข้อมูลที่มีนัยสำคัญสูงกว่า เพราะหากทำการแปลงจากชนิดข้อมูลที่มีนัยสำคัญสูงกว่าไปเป็นชนิดข้อมูลที่มีนัยสำคัญต่ำกว่าแล้ว อาจทำให้สูญเสียค่าที่แท้จริงของข้อมูลไป

ตารางที่ 2.4 แสดงลำดับความสำคัญของชนิดข้อมูลต่างๆจากสูงสุดไปต่ำสุด

ชนิดข้อมูล	นัยสำคัญ (significant)
double	สูงสุด
float	
long	
int	
short	
byte	ต่ำสุด

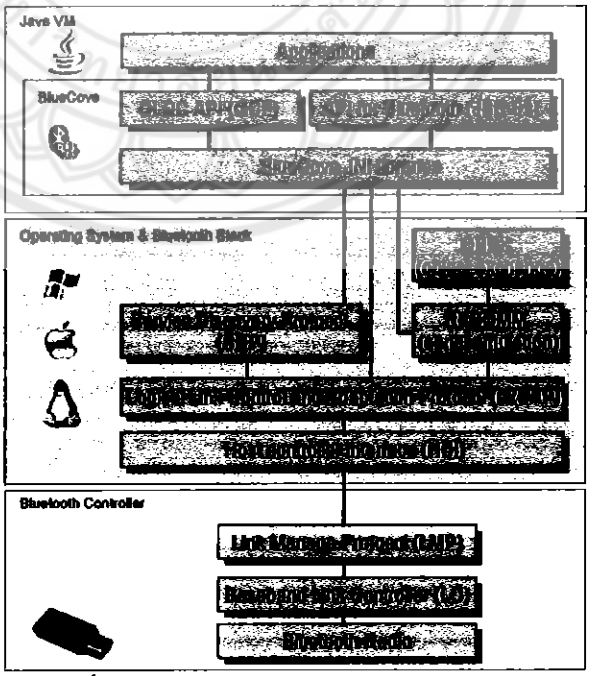
บทที่ 3
วิธีการดำเนินการ

ในบทนี้จะกล่าวถึงขั้นตอนการดำเนินงานเพื่อให้ได้มาซึ่งโปรแกรมส่งโฆษณาผ่านบลูทูธอัตโนมัติ ซึ่งมีขั้นตอนดังต่อไปนี้

1. ศึกษาโครงสร้างการทำงาน Bluetooth Protocol Stack Diagram
2. ออกแบบ Activity Diagram
3. ออกแบบ Class Diagram
4. ดำเนินการเขียนโปรแกรม

3.1 ศึกษาโครงสร้าง Bluetooth Protocol Stack Diagram

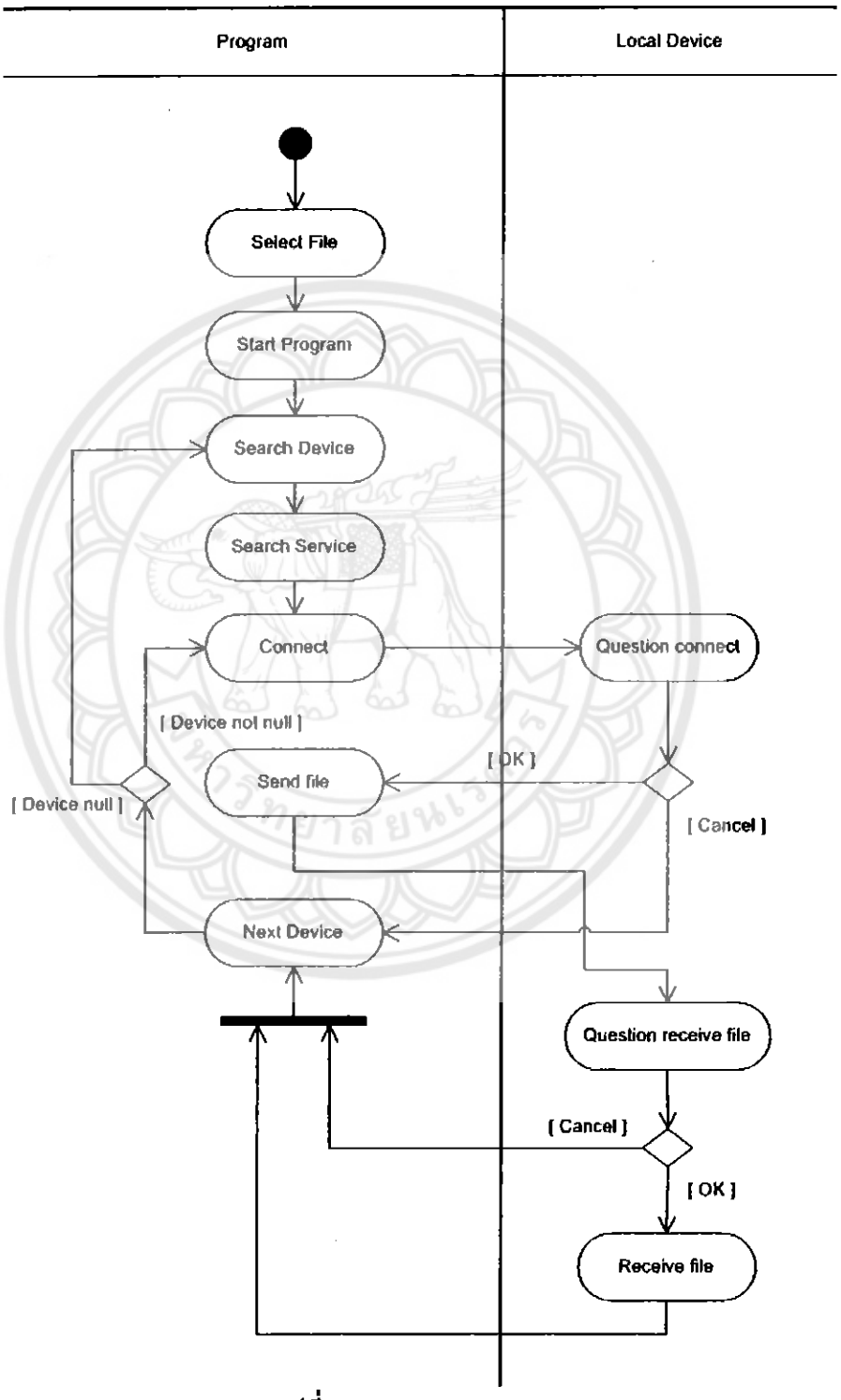
Bluetooth Stack คือโปรโตคอลที่ใช้ควบคุมการทำงานของอุปกรณ์บลูทูธหรือจะเรียกว่าบลูทูธไดรเวอร์ซึ่งจะถูกรันไปพร้อมกับระบบปฏิบัติการเป็นส่วนที่ทำหน้าที่ควบคุมการทำงานของ Bluetooth Controller ส่วน BlueCove เป็นส่วนที่ทำงานอยู่ในระดับ Application ซึ่งทำหน้าสร้างและจัดการอุปกรณ์บลูทูธ เช่น การค้นหาอุปกรณ์ การดึงข้อมูล การค้นหาเซอร์วิส การสร้างการเชื่อมต่อ เป็นต้น



รูปที่ 3.1 Bluetooth Protocol Stack Diagram

3.2 ออกแบบ Activity Diagram

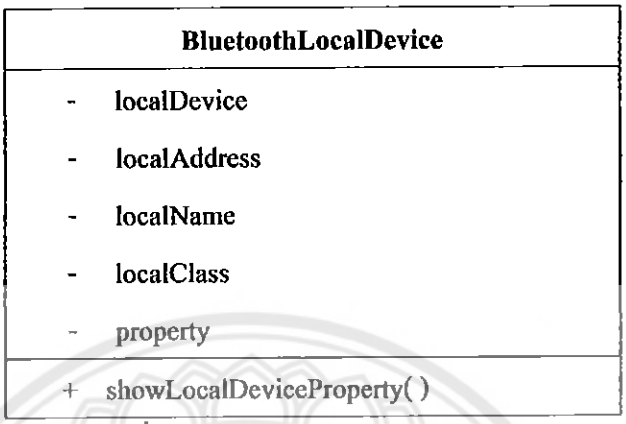
Activity Diagram คือ โครงสร้างการทำงานหลักของโปรแกรมรวมถึงการกระทำต่างๆที่เกิดขึ้นกับตัวโปรแกรมโดยมีจุดเริ่มต้นและจุดสิ้นสุดการทำงานของโปรแกรม



รูปที่ 3.2 Activity Diagram

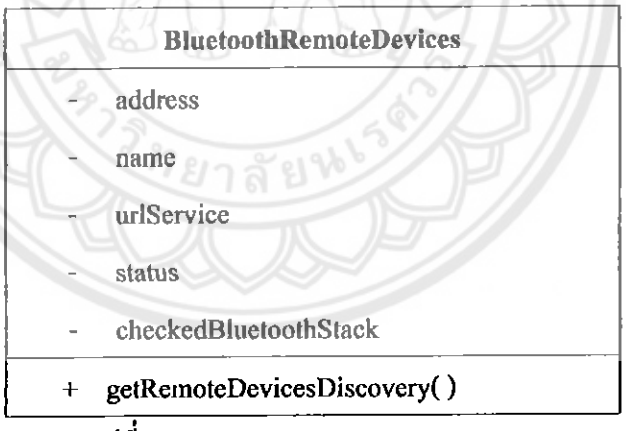
3.3 ออกแบบ Class Diagram

Class "BluetoothLocalDevice" ทำหน้าที่ค้นหา Protocol Stack ของ Bluetooth Local Device เพื่อเรียกไดรเวอร์บลูทูธให้ทำงานพร้อมทั้งคืน address, name, class, property ของอุปกรณ์



รูปที่ 3.3 Class BluetoothLocalDevice

Class "BluetoothRemoteDevices" ทำหน้าที่ค้นหา Bluetooth Remote Devices ทั้งหมดที่ถูกค้นพบในระยะการทำงานของ Bluetooth Local Device พร้อมทั้งคืน address, name ของอุปกรณ์



รูปที่ 3.4 Class BluetoothRemoteDevice

1506062
ห้องสมุดคณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

Class "BluetoothRemoteServices" ทำหน้าที่ค้นหาเซอร์วิส (File Transfer Service) ของ Bluetooth Remote Device ทั้งหมดที่ถูกค้นพบ

ปธ.
ก4267
2551

BluetoothRemoteServices
- OBEX_FILE_TRANSFER - btDevice
+ getRemoteServicesSearch( )

รูปที่ 3.5 Class BluetoothLocalService

Class "BluetoothConnectClients" ทำหน้าที่สร้างเซสชัน (Session) การเชื่อมต่อระหว่าง Local Device กับ Remote Device แต่ละอุปกรณ์พร้อมทั้งส่งไฟล์โฆษณาให้กับ Remote Device

BluetoothConnectClients
+ getBluetoothConnectClients( )

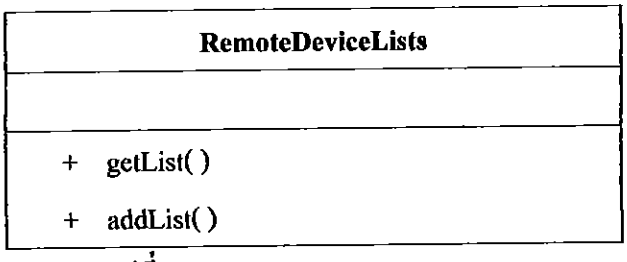
รูปที่ 3.6 Class BluetoothConnectClients

Class "BluetoothJFrame" ทำหน้าที่สร้าง Graphic User Interface ของโปรแกรมและเป็น Class หลักที่เรียกการทำงานของ Class อื่นๆ

BluetoothJFrame
- devicesDiscovered - checkedProcess - tableModel
+ main( ) + searchRemoteDevices( ) + searchRemoteServices( ) + connectedClients( )

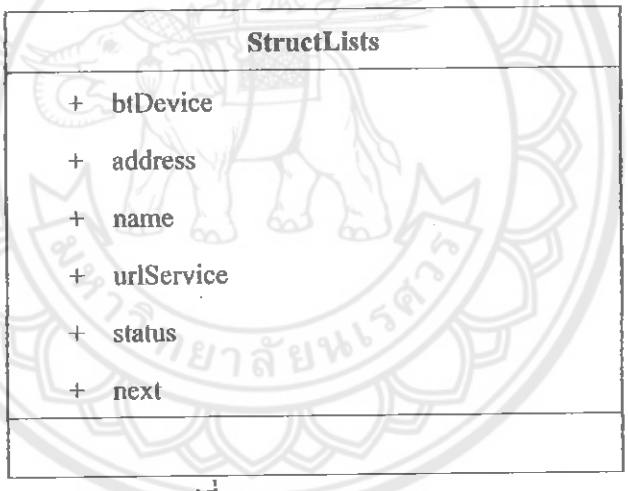
รูปที่ 3.7 Class BluetoothJFrame

Class "RemoteDeviceLists" ทำหน้าสร้างลิงค์ลิสของอุปกรณ์ Remote Device ทั้งหมดเพื่อนำลิงค์ลิสเข้า Class "BluetoothConnectClients" เพื่อสร้างการเชื่อมต่อและส่งไฟล์โฆษณา



รูปที่ 3.8 Class BluetoothDeviceLists

Class "StructLists" ทำหน้าที่สร้างโครงสร้างของลิสที่ใช้เก็บข้อมูลของอุปกรณ์ Remote Device



รูปที่ 3.9 Class StructLists

3.4 ดำเนินการเขียนโปรแกรม

Class "BluetoothLocalDevice"

```
 */
public class BluetoothLocalDevice {

    public LocalDevice localDevice;
    public String localAddress;
    public String localName;
    public DeviceClass localClass;
    public String property;

    public BluetoothLocalDevice() {
        try {
            localDevice = LocalDevice.getLocalDevice();

            if (localDevice != null) {

                localAddress = localDevice.getBluetoothAddress();
                localName = localDevice.getFriendlyName();
                localClass = localDevice.getDeviceClass();
                property = LocalDevice.getProperty("bluecove.stack");
            }
        } catch (BluetoothStateException btsE) {
            System.out.println("Local Device not found Error with : " + btsE);
        }
    }

    public void showLocalDeviceProperty() {
        System.out.println("Local Device found");
        System.out.println("-----------------------------------------");
        System.out.println("Local Device Address : " + localAddress);
```

รูปที่ 3.10 ภาพแสดงโค้ดในส่วนของ คลาส BluetoothLocalDevice

Class "BluetoothRemoteDevices"

```
 */
public class BluetoothRemoteDevices {

    public static String address;
    public static String name;
    public static String urlService;
    public static boolean status;
    public static boolean checkedBluetoothStack = false;

    public static void getRemoteDevicesDiscovery() {
        try {

            final Object inquiryCompletedEvent = new Object();
            final RemoteDeviceLists remoteDeviceLists = new RemoteDeviceLists();

            DiscoveryListener listener = new DiscoveryListener() {

                public void deviceDiscovered(RemoteDevice btDevice, DeviceClass cod) {
                    try {
                        boolean checkedBTdevice = true;
                        address = btDevice.getBluetoothAddress();
                        name = btDevice.getFriendlyName(false);
                        status = false;
                        urlService = null;

                        StructLists ptr = BluetoothJFrame.devicesDiscovered;
                        while (ptr != null) {
                            if (ptr.btDevice == btDevice) {
                                checkedBTdevice = false;
```

รูปที่ 3.11 ภาพแสดงโค้ดในส่วนของ คลาส BluetoothRemoteDevices

Class "BluetoothRemoteServices"

```
 */
public class BluetoothRemoteServices {

    public static final UUID OBEX_FILE_TRANSFER = new UUID(0x1106);
    public static RemoteDevice btDevice;

    public static void getRemoteServicesSearch() {

        try {

            UUID serviceUUID = OBEX_FILE_TRANSFER;

            final Object serviceSearchCompletedEvent = new Object();

            DiscoveryListener listener = new DiscoveryListener() {

                public void deviceDiscovered(RemoteDevice btDevice, DeviceClass cod) {
                }

                public void inquiryCompleted(int discType) {
                }

                public void servicesDiscovered(int transID, ServiceRecord[] servRecord)
                    for (int i = 0; i < servRecord.length; i++) {
                        String urlService = servRecord[i].getConnectionURL(ServiceRecor
                        if (urlService != null) {
                            StructLists ptr = BluetoothJFrame.devicesDiscovered;
```

รูปที่ 3.12 ภาพแสดงโค้ดในส่วนของ คลาส BluetoothRemoteServices

Class "BluetoothConnectClients"

```
 */
public class BluetoothConnectClients {

    public static void getBluetoothConnectClients() {
        StructLists ptr = BluetoothJFrame.devicesDiscovered;
        while (ptr != null) {
            String urlService = ptr.urlService;
            if (urlService != null) {
                if (ptr.status == false) {
                    try {
                        ClientSession clientSession = (ClientSession) Connector.open(ur
                        HeaderSet hsConnectReply = clientSession.connect(null);
                        if (hsConnectReply.getResponseCode() == ResponseCodes.OBEX_HTTP

                            HeaderSet hsOperation = clientSession.createHeaderSet();
                            hsOperation.setHeader(HeaderSet.NAME, "Advertise.txt");
                            hsOperation.setHeader(HeaderSet.TYPE, "text");

                            Operation putOperation = clientSession.put(hsOperation);

                            byte data[] = "ขณะนี้ท่านอยู่ ในระยะสัญญาณ Big C Super Center".
                            OutputStream os = putOperation.openOutputStream();
                            os.write(data);
                            os.close();

                            ptr.status = true;
                            putOperation.close();
```

รูปที่ 3.13 ภาพแสดงโค้ดในส่วนของ คลาส BluetoothConnectClients

Class "StructLists"

```
public class StructLists {

    public RemoteDevice btDevice;
    public String address;
    public String name;
    public String urlService;
    public boolean status;
    public StructLists next;

    public StructLists(RemoteDevice btDevice, String address, String name, String urlSe
        this.btDevice = btDevice;
        this.address = address;
        this.name = name;
        this.urlService = urlService;
        this.status = status;
        next = null;
    }

    public StructLists() {
        btDevice = null;
        address = null;
        name = null;
        urlService = null;
        status = false;
        next = null;
    }
}
```

รูปที่ 3.14 ภาพแสดงโค้ดในส่วนของ คลาส StructLists

Class "RemoteDeviceLists"

```
public class RemoteDeviceLists {

    public StructLists getList(RemoteDevice btDevice, String address, String name, Stri
        StructLists newList = new StructLists(btDevice, address, name, urlService, stat

        return newList;
    }

    public StructLists addList(StructLists currentList, RemoteDevice btDevice, String a
        StructLists newList = getList(btDevice, address, name, urlService, status);

        if (currentList == null) {
            return newList;
        } else {
            StructLists ptr = currentList;
            while (ptr.next != null) {
                ptr = ptr.next;
            }
            ptr.next = newList;
        }

        return currentList;
    }
}
```

รูปที่ 3.15 ภาพแสดงโค้ดในส่วนของ คลาส RemoteDeviceLists

Class "BluetoothJFrame"

```
 */
public class BluetoothJFrame extends javax.swing.JFrame {

    // Variables declaration - do not modify
    private static boolean checkedProcess = false;
    // variables to get local device
    private LocalDevice localDevice;
    private String localAddress;
    private String localName;
    private DeviceClass localClass;
    private String property;
    private boolean errorBluetoothStack = false;
    // variables for table model
    private DefaultTableModel tableModel;
    // variables for lists of devices
    private static StructLists devicesDiscovered = null;
    // variables for getting devices discovery
    private static int countDevicesDiscovered = 0;
    private static int tempProgressBar = 0;
    // variables for send a receive file
    private static final UUID OBEX_FILE_TRANSFER = new UUID(0x1106);
    private static RemoteDevice btDevice;
    private static String btName;
    //variables for counting completed or incompleted files
    private static int countSendCompleted = 0;
    private static int countSendIncompleted = 0;
    // variables for path file
    private static String pathFile;
    private static String fileName;
    private static int fileSize;
```

รูปที่ 3.16 ภาพแสดงโค้ดในส่วนของ คลาส BluetoothJFrame

บทที่ 4

ผลการทดลอง

หลักการทำงานของโครงงานนี้จะใช้หลักของการทำ Synchronized เพื่อรอ event ของอินเตอร์เฟส DiscoveryListener ที่กำลังจะสร้างการเชื่อมต่อไปยัง Remote Devices โดยsynchronized จะถูกเรียกใช้งานในแต่ละเวลาที่ Local Device inquiry หาอุปกรณ์บลูทูธ

ผลการทดลองเริ่มตั้งแต่การทดลองการทำงานของโปรแกรมเพื่อทดสอบว่าโปรแกรมสามารถค้นหาอุปกรณ์บลูทูธและสามารถส่งไฟล์โฆษณาได้ พร้อมตารางบันทึกผลเวลาหน่วงและเวลาเฉลี่ยการทำงานของโปรแกรม

4.1 ตารางบันทึกผลเวลาการทำงานของโปรแกรม

ตารางที่ 4.1 เวลาค้นหาอุปกรณ์ที่ใช้ในการส่งโฆษณาตามจำนวนเครื่อง

ครั้งที่	1 เครื่อง(วินาที)	จำนวน 5 เครื่อง (วินาที)	จำนวน 10 เครื่อง (วินาที)
1	17	33	53
2	15	31	51
3	17	33	53
4	16	32	52
5	16	32	52
เวลาค้นหาเฉลี่ย	16.2	32.2	52.2

ตารางที่ 4.2 เวลาหน่วงที่ใช้ในการส่งโฆษณาแบบเลือกกดรับตามจำนวนเครื่อง

ครั้งที่	1 เครื่อง(วินาที)	จำนวน 5 เครื่อง (วินาที)	จำนวน 10 เครื่อง (วินาที)
1	32	82	145
2	33	83	146
3	33	83	146
4	34	84	147
5	32	82	145
เวลาหน่วงเฉลี่ย	32.8	82.8	145.8

ตารางที่ 4.3 เวลาหน่วงที่ใช้ในการส่งโฆษณาแบบกดรับทันทีไม่ทำการ pairing devices

ครั้งที่	1 เครื่อง(วินาที)	จำนวน 5 เครื่อง (วินาที)	จำนวน 10 เครื่อง (วินาที)
1	25	65	85
2	21	63	83
3	22	65	85
4	23	64	84
5	23	64	84
เวลาหน่วงเฉลี่ย	22.8	64.2	84.2

ตารางที่ 4.4 เวลาหน่วงที่ใช้ในการส่งโฆษณาแบบเลือกไม่กดรับตามจำนวนเครื่อง

ครั้งที่	1 เครื่อง(วินาที)	จำนวน 5 เครื่อง (วินาที)	จำนวน 10 เครื่อง (วินาที)
1	45	173	333
2	44	171	331
3	44	173	333
4	44	172	332
5	44	172	332
เวลาหน่วงเฉลี่ย	44.2	172.2	332.2

สรุปผลการทดลอง

จากผลการทดลองในตารางที่ 1 - 4 สรุปผลได้ว่าเวลารวมเฉลี่ยในแต่ละตารางมีค่ามากขึ้นตามลำดับ ตามจำนวนอุปกรณ์บลูทูธ ที่โปรแกรมสามารถค้นหาอุปกรณ์ได้ดังนี้

1. เวลารวมเฉลี่ยในการค้นหาอุปกรณ์ที่ใช้ในการโฆษณาตามจำนวนเครื่อง (ตารางที่ 1) จะมีค่ามากขึ้นตามลำดับอุปกรณ์ที่ถูกค้นพบทั้งหมด
2. เวลารวมเฉลี่ยในการส่งโฆษณาแบบเลือกกดรับทันทีไม่มีการ Pairing devices (ตารางที่ 3) จะใช้เวลารวมเฉลี่ยน้อยกว่าในการส่งโฆษณาแบบเลือกกดรับทันทีแต่มีการ Pairing devices ก่อนการรับโฆษณา (ตารางที่ 2) เนื่องจากอุปกรณ์บลูทูธต้องใช้เวลาในการยืนยัน Pass key กันก่อน
3. เวลารวมเฉลี่ยในการส่งโฆษณาแบบเลือกไม่กดรับตามจำนวนเครื่อง (ตารางที่ 4) จะใช้เวลารวมเฉลี่ยมากที่สุดและมากขึ้นเรื่อยๆตามลำดับอุปกรณ์ที่ถูกค้นพบทั้งหมด

บทที่ 5
บทสรุป

โครงงานนี้ได้พัฒนาโปรแกรมส่งโฆษณาผ่านบลูทูธอัตโนมัติ ด้วยภาษา Java API for Bluetooth (JSR-82) with BlueCove library พัฒนาบน NetBeans IDE6.5

เมื่อพัฒนาโปรแกรมส่งโฆษณาผ่านบลูทูธอัตโนมัติ โดยสามารถส่งโฆษณาได้ ดังที่แสดงผลการทดลองในบทที่ 4 ไปแล้วนั้น ในบทนี้จะทำการสรุปผลการทดลอง วิเคราะห์ผลการทดลอง ผลการทดสอบ และแสดงปัญหาที่พบของโครงงานนี้ พร้อมทั้งเสนอแนวทางในการแก้ไขปัญหาในเบื้องต้น

5.1 สรุปการทดลอง

จากผลการทดลองในบทที่ 4 สรุปได้ว่าเวลาที่ใช้ในการทำงานของโปรแกรมโดยรวมเฉลี่ยมีค่ามากขึ้นตามลำดับตามจำนวนอุปกรณ์ที่ถูกค้นพบ นอกจากนี้เวลารวมเฉลี่ยของโปรแกรมจะเปลี่ยนแปลงไปตามแต่ละวิธีที่ใช้ในการยืนยันการรับโฆษณา เช่น การกดรับทันทีไม่มีการ Pairing devices, การกดรับแบบทำการ Pairing devices ก่อนการกดรับโฆษณา และไม่กดรับโฆษณาเลยไม่ว่ากรณีใดๆ

5.2 วิเคราะห์ผลการทดลอง

จากผลการทดลองในบทที่ 4 สามารถวิเคราะห์ได้ว่าประสิทธิภาพโดยรวมของโปรแกรมขึ้นอยู่กับปัจจัยต่างๆดังนี้ อุปกรณ์บลูทูธ (Local Devices) ในแต่ละเวอร์ชันมี Class ต่างกัน, วิธีในการยืนยันกดรับโฆษณาต่างกัน, จำนวนเครื่องที่ถูกค้นพบในแต่ละรอบการทำงาน, อุปกรณ์บลูทูธ (Remote Devices) ในแต่ละรุ่นต่างกัน ซึ่งปัจจัยต่างๆจะมีผลต่อเวลาในการทำงานของโปรแกรมและทำให้ประสิทธิภาพโดยรวมเปลี่ยนแปลงไปด้วย

5.3 ปัญหาและแนวทางแก้ไข

ปัญหาที่พบในการปฏิบัติโครงงานโปรแกรมไม่สามารถค้นหาไฟล์ทรานเฟอร์เซอร์วิสของอุปกรณ์บลูทูธได้ครบทุกอุปกรณ์ เนื่องจาก Java API JSR 82 ที่มีให้ใช้งานไม่สามารถค้นหาไฟล์ทรานเฟอร์เซอร์วิสบนอุปกรณ์บลูทูธบางชนิดได้ทำให้โปรแกรมบลูทูธที่พัฒนาขึ้นไม่สามารถสร้างการเชื่อมต่อของอุปกรณ์ชนิดนั้นได้

แนวทางแก้ไขเลือกใช้ Java API JSR 82 รุ่นใหม่ที่รองรับการทำงานของโปรแกรม

เอกสารอ้างอิง

[1] Bluecove[Online].Available from : URL http:\\www.bluecove.org 2008

[2] วีระศักดิ์ ซึงถาวร. **Java Programming Volume I**. กรุงเทพมหานคร : ซีเอ็ดยูเคชั่น. 2549.

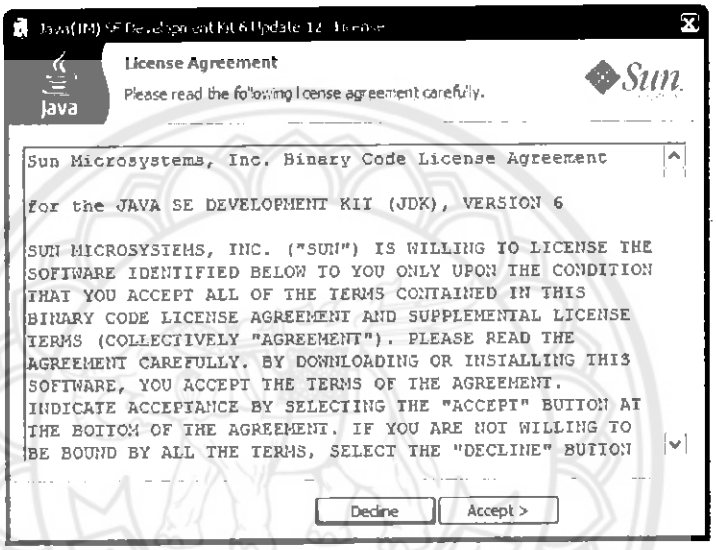
[3] ทศพล ธนะทิพานนท์ และ วรเศรษฐ สุวรรณิก. เขียนโปรแกรม JAVA เบื้องต้น. กรุงเทพมหานคร : ซีเอ็ดยูเคชั่น. 2549.

[4] วรเศรษฐ สุวรรณิก. **Java GUI using NetBeans.** กรุงเทพมหานคร : วรรณิก. 2551.

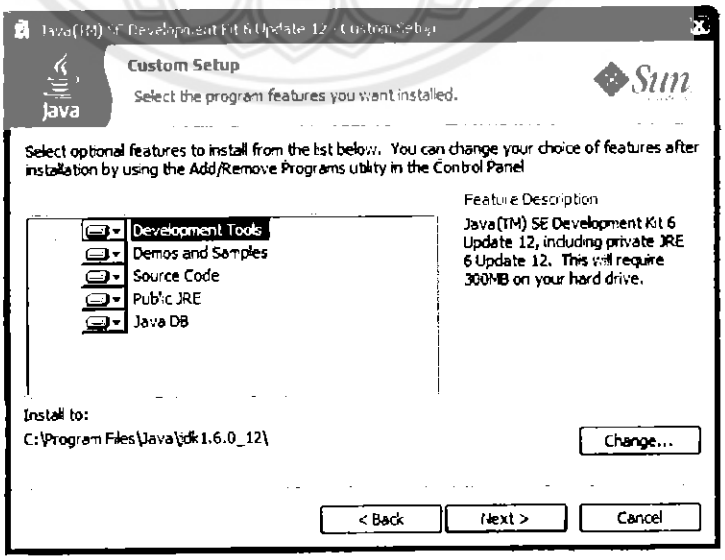
ภาคผนวก

ขั้นตอนการติดตั้งโปรแกรม

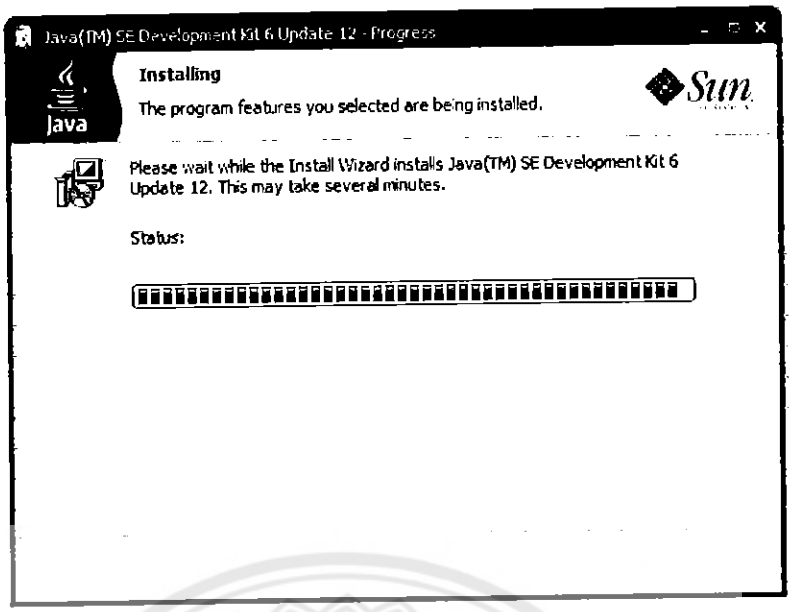
โปรแกรม JDK (J2SE Develop kit 6) jdk-6u12-windows-i586-p.exe ในที่นี่เป็นเวอร์ชัน 1.6.0.120 ทำการดับเบิ้ลคลิกไอคอนของไฟล์ แล้วโปรแกรมจะสำรวจคอมพิวเตอร์ เพื่อเตรียมการติดตั้ง และ install shield wizard จะแสดง dialog ให้เราอ่านข้อตกลงทางกฎหมาย



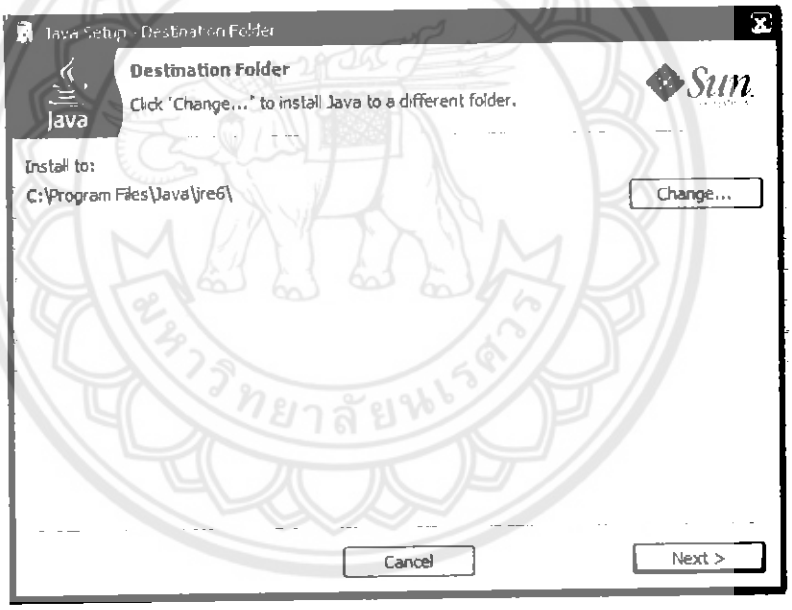
ทำการเลือก Accept แล้วกดปุ่ม Next จะปรากฏ Custom Setup Dialog ให้เลือกว่าจะติดตั้งอะไรบ้าง ในที่นี่เลือกอันแรกเพียงอันเดียว



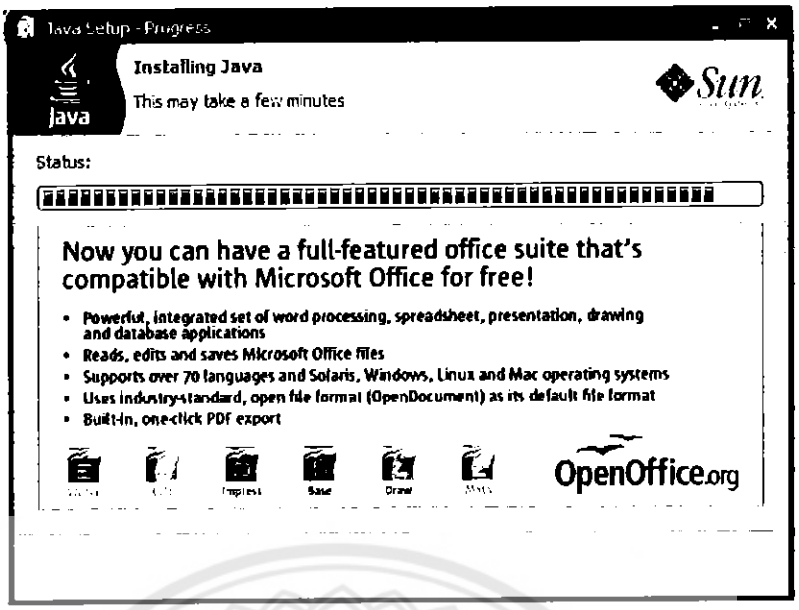
เมื่อเลือกแล้วทำการกด Next จะทำการก๊อบปี้ไฟล์ไปติดตั้งบนฮาร์ดดิสก์



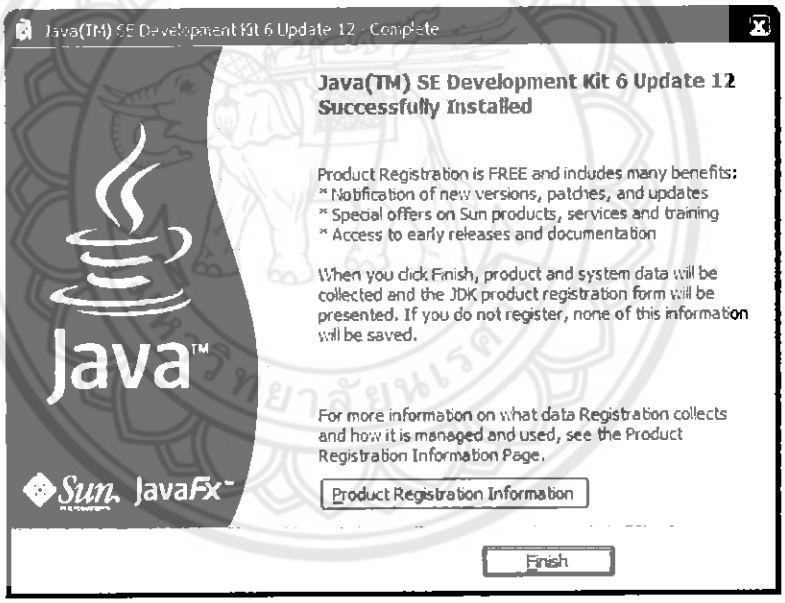
ระหว่างที่ติดตั้ง โปรแกรมจะถามเราว่าจะติดตั้ง JRE หรือไม่



ในที่นี่เราต้องการติดตั้ง ให้เลือกกด Next

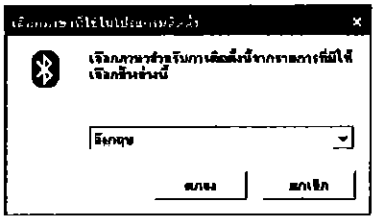


โปรแกรมก็จะทำการติด JRE จนเสร็จ แล้วจะกลับไปติดต่อ JDK ต่อไปจนเสร็จ

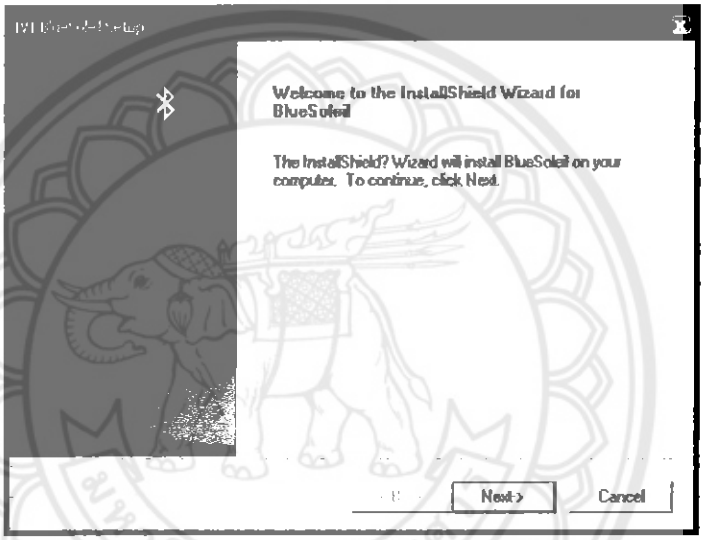


กดปุ่ม Finish เพื่อสิ้นสุดการติดตั้ง JDK

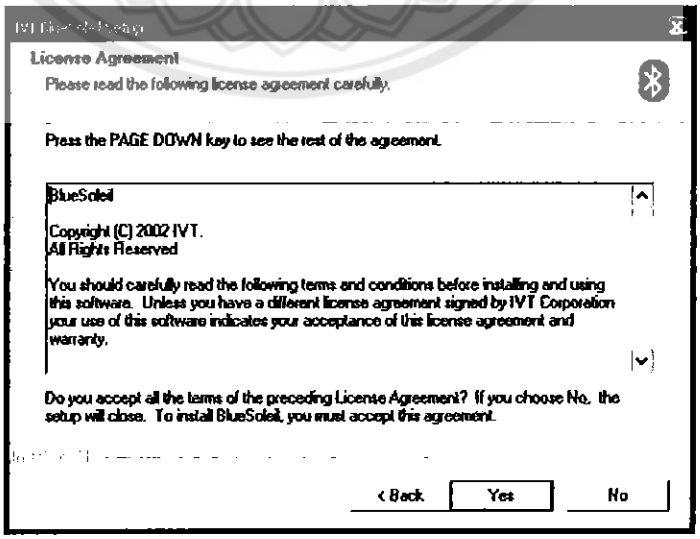
ขั้นตอนการติดตั้งโปรแกรม Blue Soleil1.6 ทำการติดตั้ง Driver Bluetooth



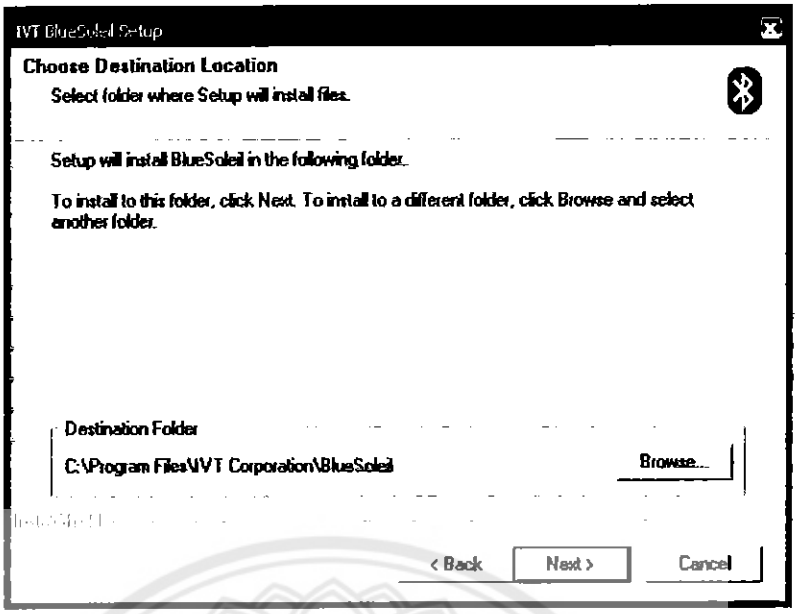
ทำการเลือกภาษาที่จะติดตั้ง ในที่นี่เลือกภาษาอังกฤษ



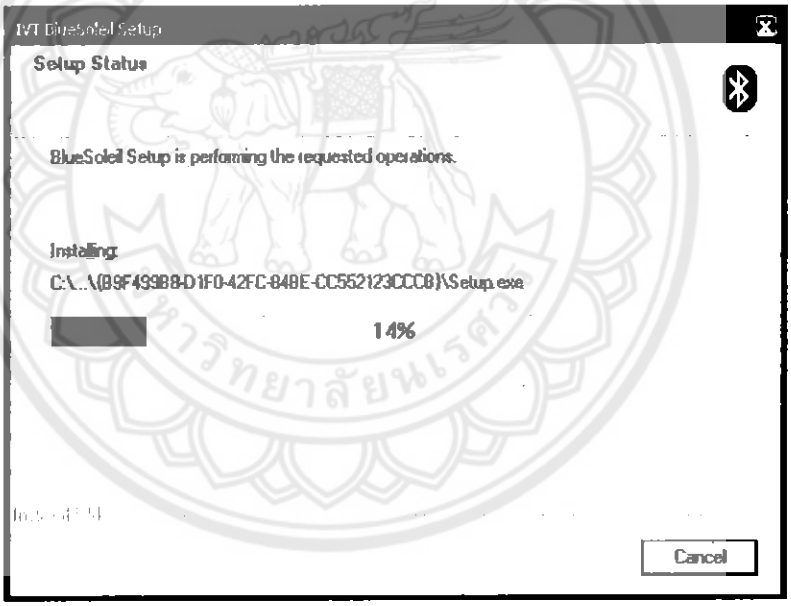
คลิก Next เพื่อไปยังหน้าถัดไป



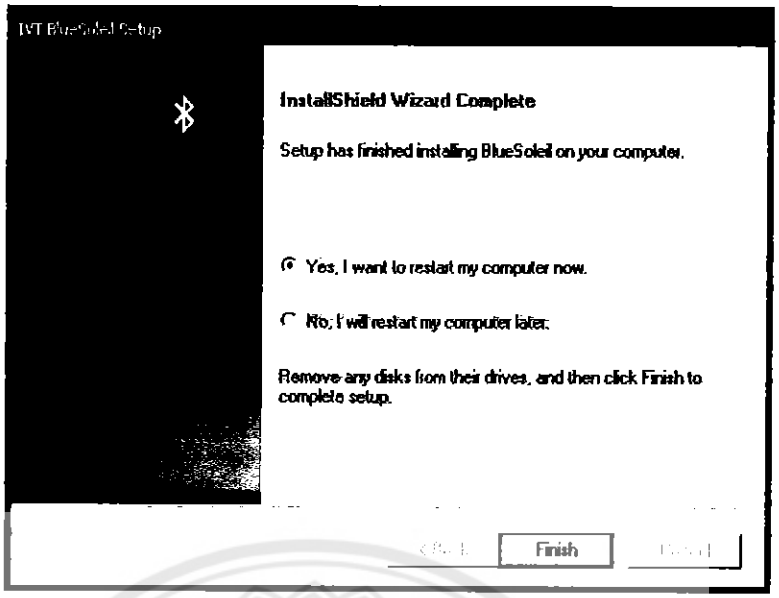
หน้านี้จะพบ License ของโปรแกรม เราควรจะอ่าน แล้วกด Next เพื่อไปยังหน้าถัดไป



ทำการเลือก Directory ที่เก็บไฟล์ Setup เสร็จแล้วกด Next

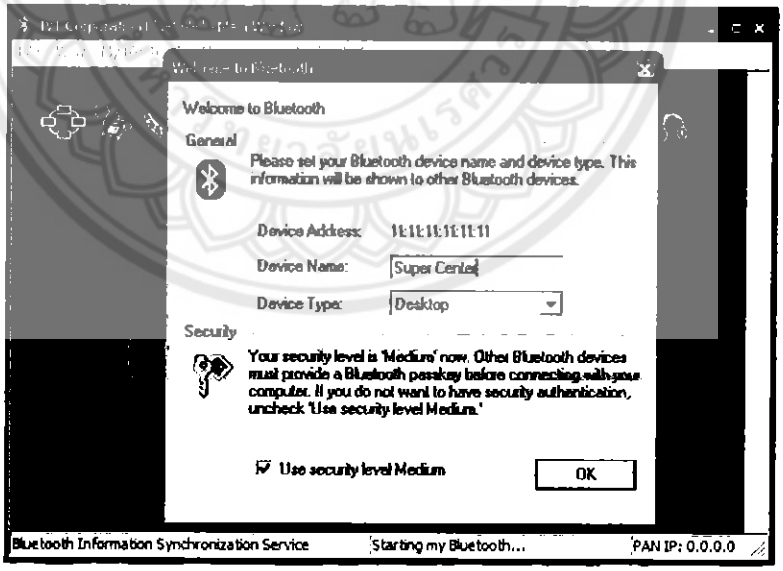


โปรแกรมจะทำการก๊อบปี้ไฟล์ติดตั้งลง Directory ที่เราเลือกไว้

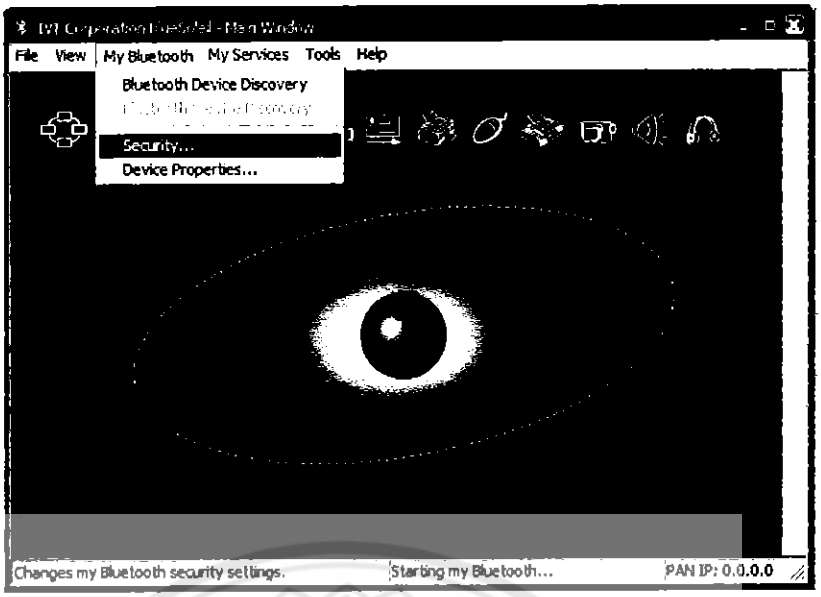


คลิก Finish เพื่อทำการเสร็จสิ้นการติดตั้ง แล้วคอมพิวเตอร์จะทำการ Restart

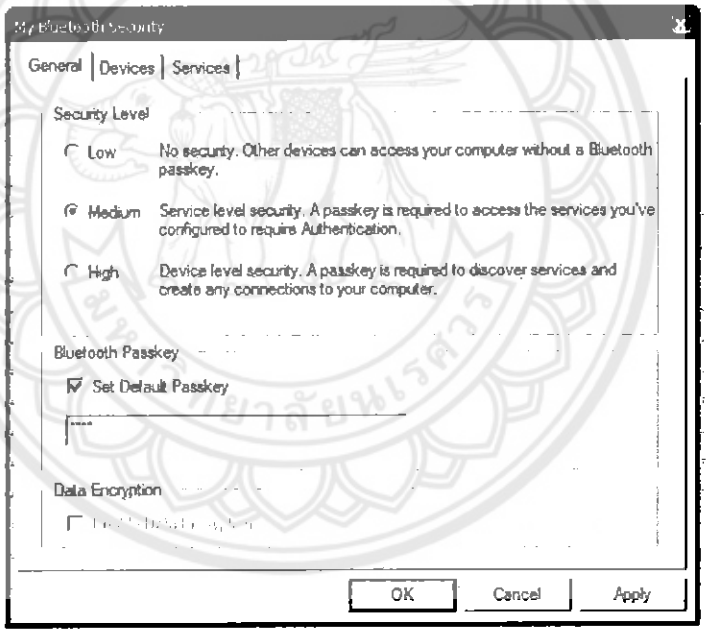
เมื่อคอมพิวเตอร์เปิดขึ้นมาอีกครั้งทำการดับเบิ้ลคลิกไอคอน Blue Soil เพื่อเปิดโปรแกรมขึ้นมา



กด OK เพื่อเข้าโปรแกรม



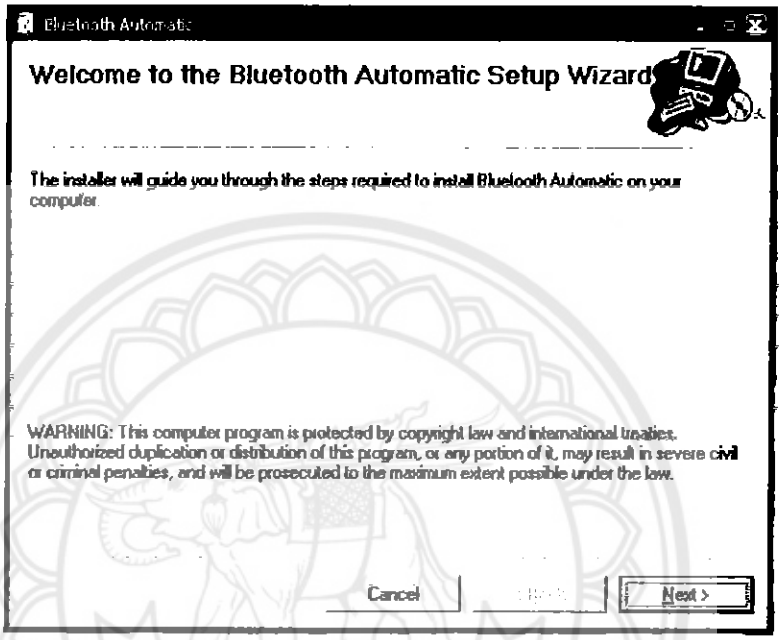
ทำการเลือก Tab My Bluetooth แล้วไปที่ Security เพื่อปรับแต่งค่า Pass Key



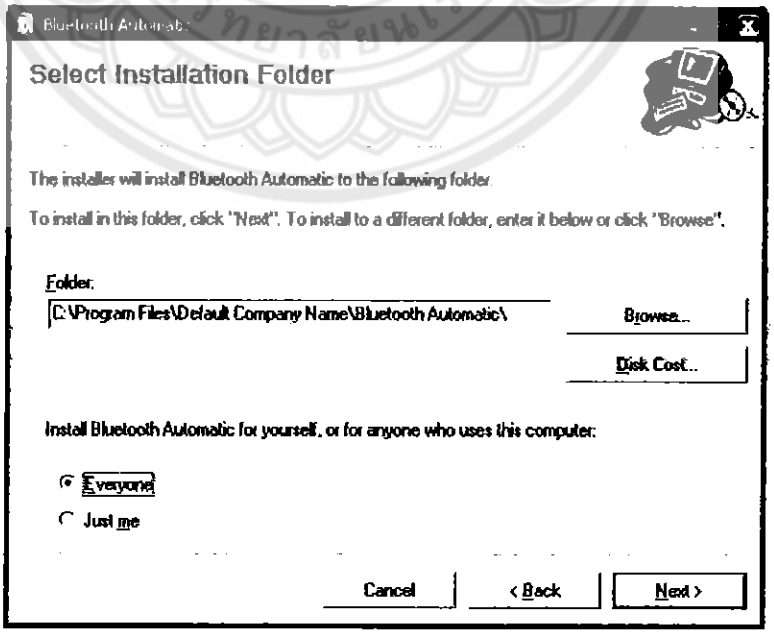
ไปที่ Tab General แล้วไปคลิกที่ Set Default Passkey ในที่นี่ปรับค่า Passkey เท่ากับ 1334

การติดตั้งไฟล์โปรแกรม Bluetooth Autometic Advertise Transfer

ทำการดับเบิ้ลคลิกไอคอน Bluetooth Autometic.exe แล้วจะมีหน้าต่างขึ้นมาให้ทำการ Setup

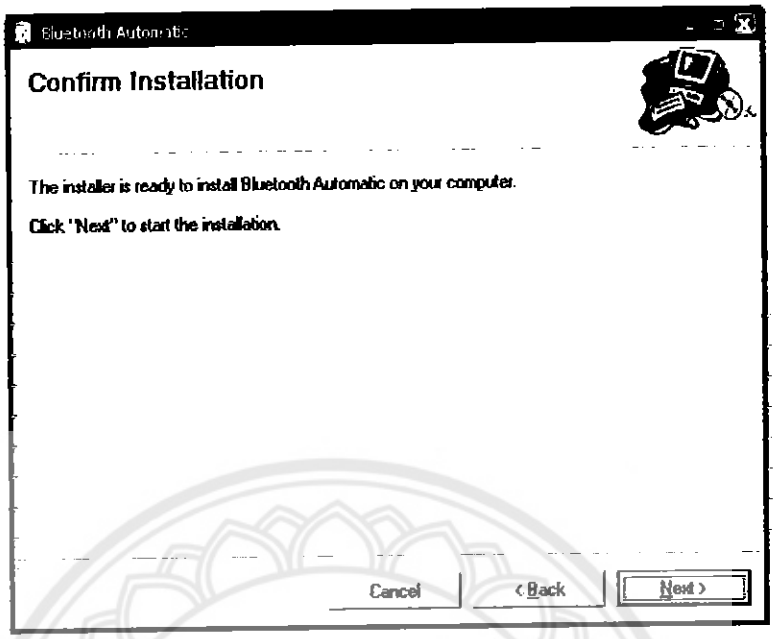


กด Next เพื่อไปยังหน้าถัดไป

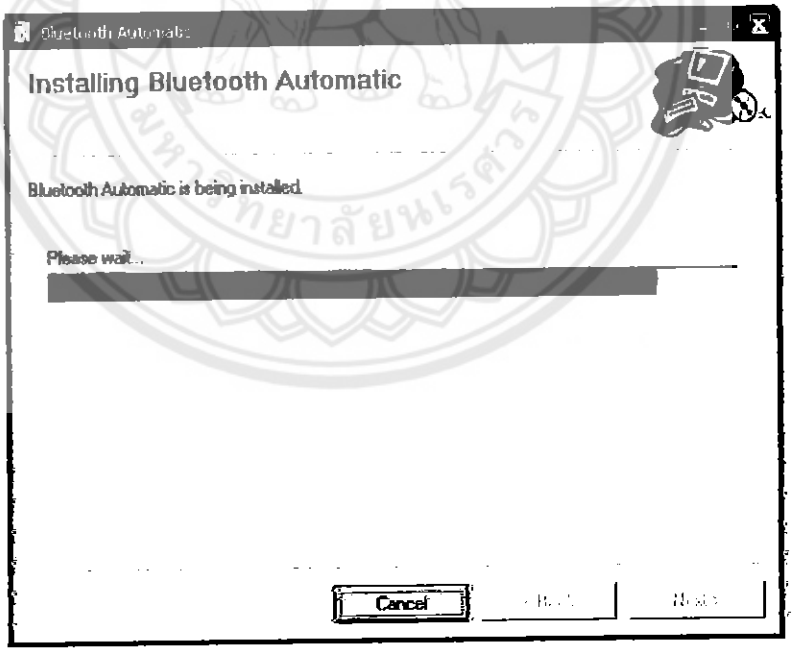


ทำการเลือก Directory ที่เก็บของไฟล์ setup แล้วทำการเลือก Choice เป็น Everyone

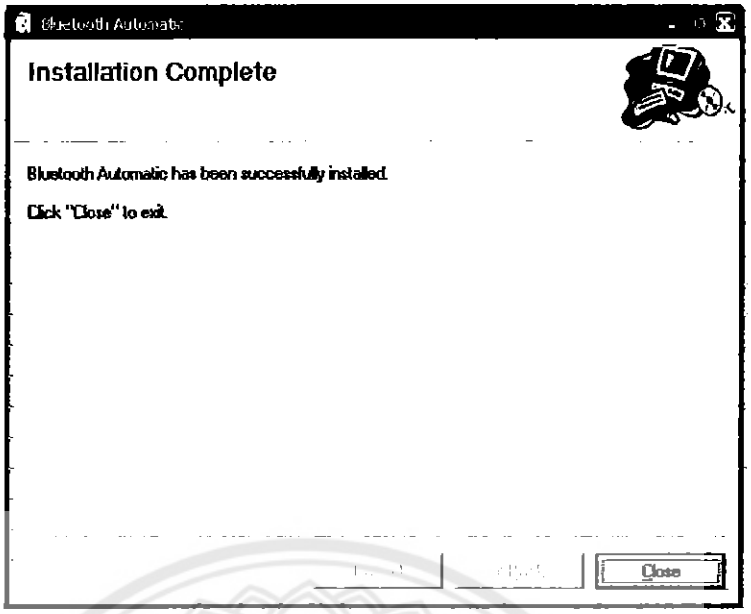
แล้วกด Next



โปรแกรมจะถามอีกครั้งเพื่อยืนยันความแน่ใจของผู้ลงโปรแกรม กด Next เพื่อทำการลงโปรแกรม



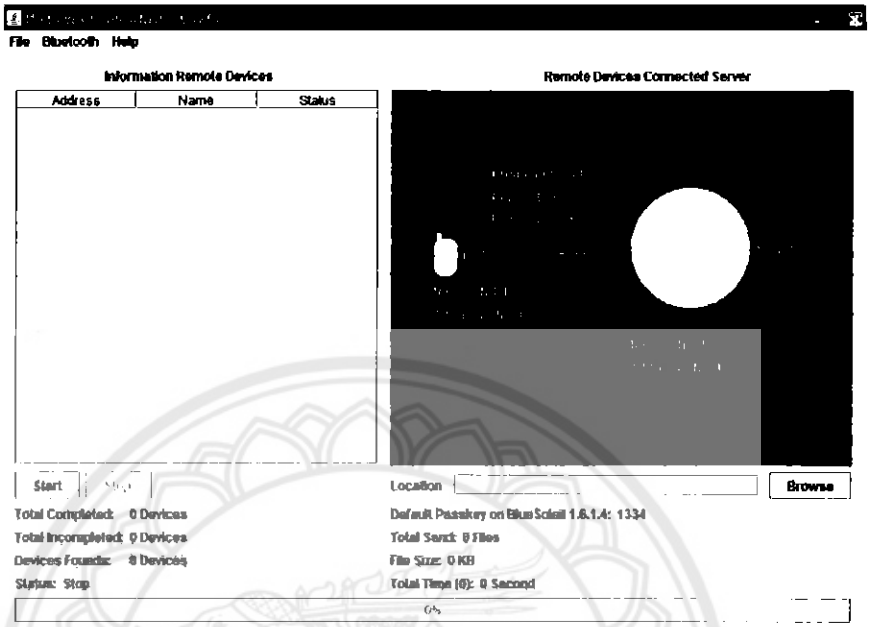
โปรแกรมทำการก๊อบปี้ไฟล์ติดตั้งลง Directory ที่เราเลือกไว้



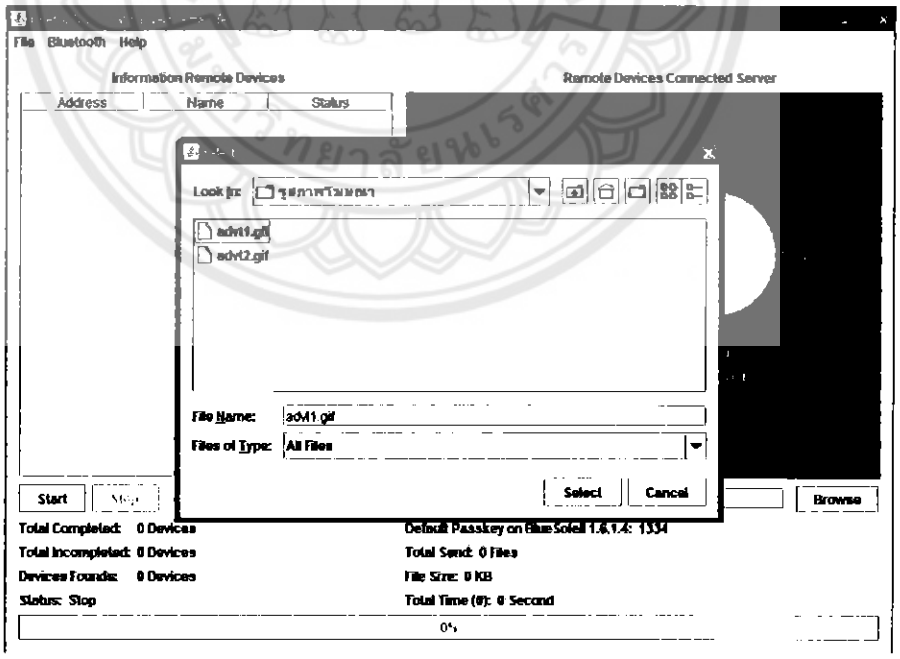
กด Close เพื่อทำการเสร็จสิ้นการติดตั้ง

ไฟล์ภาพหรือไฟล์โมษณา ควรไม่เกิน 2 เมกะไบต์ และมีขนาดกว้างคูณยาวประมาณ 150 x 150 pixel จะทำให้ภาพออกมาพอดีกับหน้าจอมือถือ

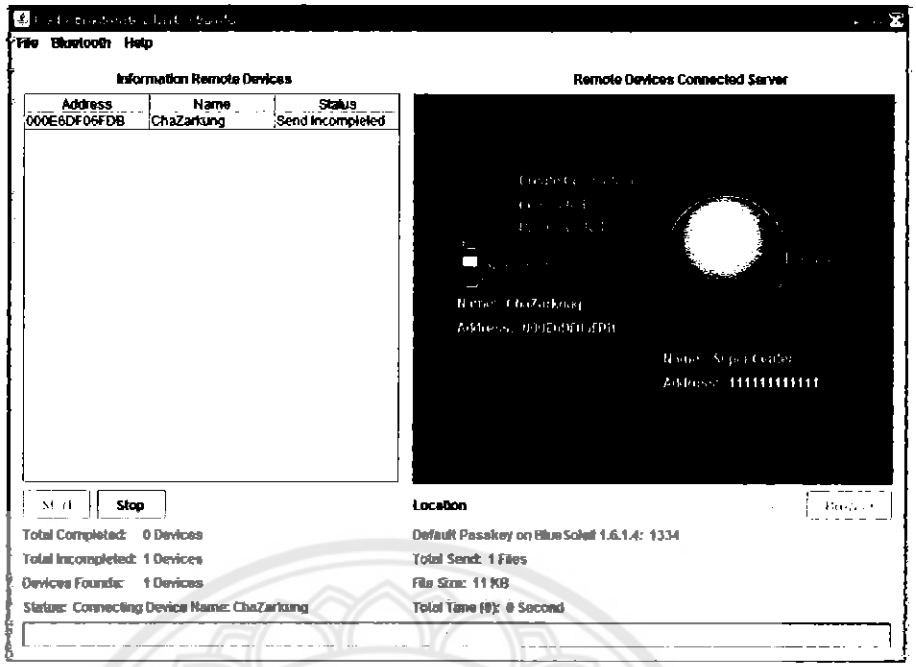
การใช้งานโปรแกรม Bluetooth Autometic Advertise Transfer



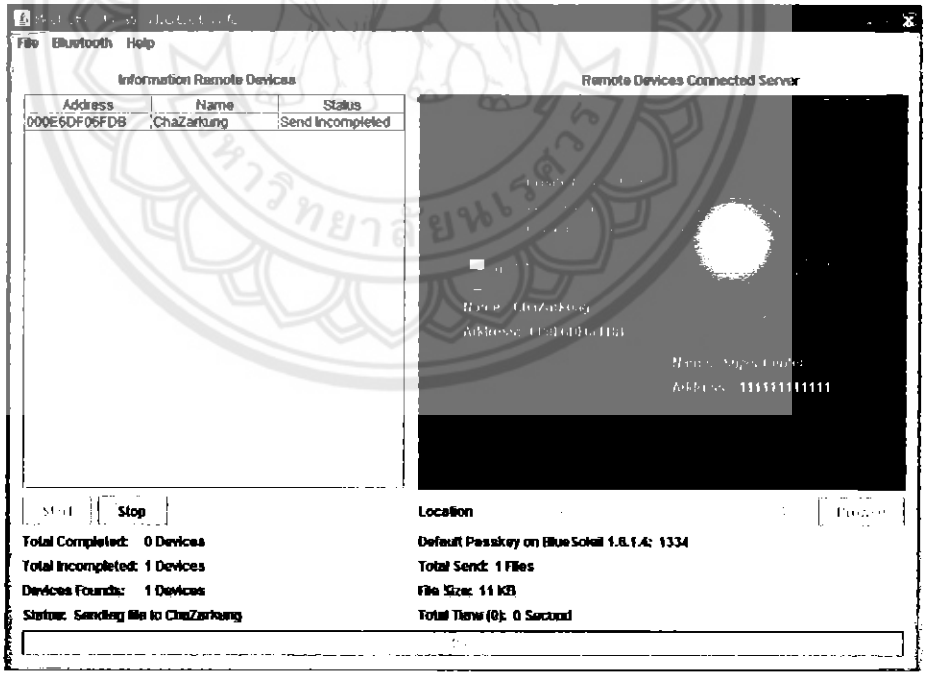
ทำการดับเบิ้ลคลิกไอคอน Bluetooth Autometic.exe แล้วจะมีหน้าต่างโปรแกรมเป็นดังนี้



กดปุ่ม Browse เพื่อเลือกไฟล์โฆษณาที่ต้องการใช้ส่งให้กับ Remote Devices



จากนั้นกดปุ่ม Start เพื่อเริ่มการทำงานโปรแกรม Bluetooth Autometic Advertise Transfer



จากนั้นโปรแกรมจะทำสร้างการเชื่อมต่อและส่งไฟล์โฆษณาให้กับ Remote Devices

ประวัติผู้เขียนโครงงาน

ชื่อ นางสาวกาญจนา ธะนีวรรณ์

ภูมิลำเนา 43 หมู่ 1 ต.ท่าด้วง อ.หนองไผ่ จ.เพชรบูรณ์ 67140

ประวิติการศึกษา

- จบมัธยมศึกษาจากโรงเรียนผู้ใหญ่พระสุธรรมยานเถระวิทยา
- ปัจจุบันกำลังศึกษาอยู่ระดับปริญญาตรีชั้นปีที่ 4
  สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
  มหาวิทยาลัยนเรศวร

E-mail: bee_engi@hotmail.com

ชื่อ นายบรรชา จำปาศักดิ์

ภูมิลำเนา 20 หมู่ 14 ต.หินดาต อ.ป่างศิลาทอง จ.กำแพงเพชร 62120

ประวิติการศึกษา

- จบมัธยมศึกษาจากโรงเรียนนครสวรรค์
- ปัจจุบันกำลังศึกษาอยู่ระดับปริญญาตรีชั้นปีที่ 4
  สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
  มหาวิทยาลัยนเรศวร

E-mail: zhegikung_cpe@hotmail.com

ชื่อ นายเอกลักษณ์ รังสิกรรพุม

ภูมิลำเนา 101 หมู่4 ต.สะเดียง อ.เมือง จ.เพชรบูรณ์ 67000

ประวิติการศึกษา

- จบมัธยมศึกษาจากโรงเรียนเพชรพิทยาคม
- ปัจจุบันกำลังศึกษาอยู่ระดับปริญญาตรีชั้นปีที่ 4
  สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
  มหาวิทยาลัยนเรศวร

E-mail: t-pointshoot@hotmail.com