BLUETOOTH PICONET APPLICATIONS

CHAPTER 1: INTRODUCTION

The Bluetooth wireless technology provides for a low-cost, low-power, short-range radio link for mobile devices and for Local Area Network (LAN) access points. It offers fast and reliable digital transmissions of both voice and data over the globally available, unlicensed, 2.4 GHz Industrial, Scientific and Medical (ISM) band. The Bluetooth technology comprises hardware, software and interoperability requirements. It has been adopted not only by all major players in the telecom, computer and home entertainment industry, but also in such diverse areas as the automotive industry and the health care sector. An impressively extensive amount of research has been conducted on Bluetooth, by both industry and academics, but integrating Bluetooth with a target application remains a complex task, potentially involving a full redesign of the application, making the universal adoption of Bluetooth problematic.

The major goal behind this project lies in the implementation of a complete Bluetooth solution to enable wireless data transfer between a number of Bluetooth-enabled laptop PCs, simultaneously permitting access to the Internet through a LAN - as envisioned in a typical office conference scenario. Time permitting; interoperability will be explored, by quantifying the effect of Bluetooth on a fully implemented IEEE 802.11 wireless LAN (WLAN).

Bluetooth applications are currently an active research area with industry trying to create new markets for Bluetooth-enabled products and with academics keen to explore the possibility of this wireless solution becoming the dominant technology in short-range wireless communications. This work fits-in with the current research trends and is intended to demonstrate the applicability of Bluetooth and its advantages in a typical office scenario. Current solutions require the use of wires to permit total connectivity or the need for expensive WLAN access hardware. Bluetooth, however, is ideal for the office environment and the ability of the technology to create *ad hoc* networks *on the fly* will reduce connection delays, alleviate expensive hardware and wires, while permitting total connectivity at a minimum of cost.

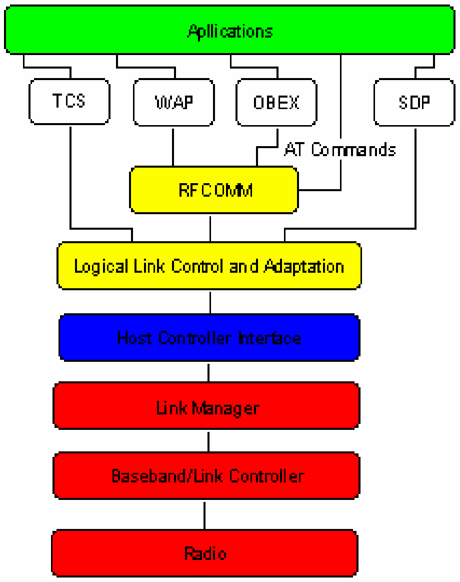
CHAPTER 2: TECHNICAL OVERVIEW

2.1 Bluetooth Overview

Bluetooth is a low cost, low power, short-range radio technology. It was originally developed as a cable replacement to connect devices such as mobile phone handsets, headsets and laptops. However enabling standardised wireless communications between any electrical devices has extended this. Bluetooth has created the notion of a Personal Area Network (PAN), a close range wireless network known as an ad-hoc network [2].

An ad-hoc (or spontaneous) network is where devices are part of the network only when in close proximity to the rest of the network. This means that the network topology and membership can be constantly changing. The Service Discovery Protocol (SDP) allows Bluetooth devices to discover what services are available or to find a Bluetooth device that supports a specific service (see Appendix I). The main advantages of Bluetooth can be related to its robustness, low complexity and capability to handle both data and voice transmission at the same time.

A key feature of the Bluetooth specification is that it aims to allow devices from lots of different manufacturers to work with one another. Bluetooth does not just define a radio system; it also defines a software stack (Bluetooth Protocol Stack) to enable applications to find other Bluetooth devices in the area, discover what services they offer and use those services that they require. The Bluetooth protocol stack is shown below in figure 2-1.



Protocol Stack	*Description*
Telephony Control Protocol Specification (TCS)	Provides telephony services
Service Discovery Protocol (SDP)	Lets Bluetooth devices discover what services other Bluetooth devices support
WAP and OBEX	Provides interfaces to the higher layer parts of other Communication Protocols
RFCOMM	Provides an RS232 like serial interface
Logical Link Controller and Adaptation (L2CAP)	Multiplexes data from higher layers, and converts between different packet sizes
Host Controller Interface (HCI)	Handles communication between a separate host and a Bluetooth module
Link Manager (LM)	Controls and configures links to other devices

Figure 2-1: The Bluetooth Protocol Stack.

Table 2 -1: Description of Protocol Stack layers.

2.2 How Bluetooth Operates

The Bluetooth system operates in the 2.4GHz Industrial, Scientific and Medical (ISM) band, which is globally available and license free. To make Bluetooth as robust as possible the operating band is divided into 1MHz-spaced channels, each signalling data at 1Mb/s. This is achieved by using GFSK (Gaussian Frequency Shift Keying) modulation scheme.

After each packet, both devices retune their radios to a different frequency, effectively hopping from radio channel to radio channel; this is known as FHSS (Frequency Hopping Spread Spectrum). Therefore if a transmission is compromised by interference on one channel, the retransmission will always be on a different (hopefully clear) channel. Each Bluetooth time slot lasts 625µs (625 microseconds) and generally devices hop once per packet, giving a hop rate of 1600 hops/second.

If devices are to hop to new frequencies after each packet, there has to be some sort of agreement between the devices. This is where the concept of Master and Slave appears; the Master is the Bluetooth device that sets the frequency hoping sequence. The Slave synchronises to the Masters in time and frequency by following the Master's frequency hoping sequence. Every Bluetooth device has a unique Bluetooth device address and a 28-bit Bluetooth clock. The baseband part of the Bluetooth System uses a special algorithm, which calculates the frequency hop sequence from the masters clock and device address. In addition to controlling the frequency hop sequence, the Master controls when Slaves are to transmit using Time Division Multiplexing (TDM).

When there is just one Master and one Slave the system is called a **Point to Point** connection. When many Slaves are connected to one Master, the system is called a **Point to Multipoint**. Both these types are referred to as a **Piconet** (shown in figure 2-2) and all follow the frequency hopping sequence of the Master. The Slaves in the Piconet only have links to the Master and no direct links between Slaves.

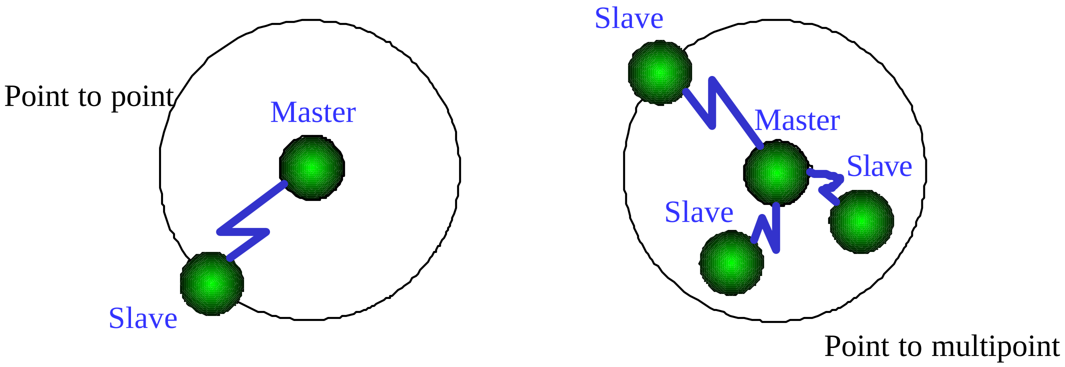


Figure 2-2: Point to point and point to multipoint piconets [2]

The specification limits the number of Slaves in a Piconet to seven, however larger networks can be set up using many Piconets to communicate with one another, and this is known as a **Scatternet**. In a Scatternet a device can be a Master in one Piconet and a Slave in another, or a Slave in both, but never a Master of more than one Piconet.

Further details on Bluetooth technology is included in Appendix I.

CHAPTER 3: RELATED WORK

3.1 Applications

[SSN] provide an explanatory study of the application visions of Bluetooth and give an overview of the business opportunities of the technology. As detailed in [KMSB], Bluetooth applications are currently hot research topics, with academics trying to implement the technology in all aspects of wireless networking. The research areas can be broadly classified as follows.

3.1.1 Automotive applications

Nüsser *et al.* [NP] are mainly concerned with the applicability of a Bluetooth-based wireless access system as an integral part of a car-based communication and infotainment platform. A potential in-car network architecture is described, restricting the use of Bluetooth to safety-uncritical applications. Even though a detailed framework is presented, the authors do not actually implement, test, or even simulate their ideas. Similarly, Murphy *et al.* [MWF] determine the suitability of Bluetooth for use in a mobile environment and in particular the case of data transfer between two fast-moving vehicles. The simulation results are promising, provided some minor modifications are made to the Bluetooth baseband protocol. However, a hardware implementation is not presented, which would provide insights into the direct applicability of Bluetooth. The robustness of Bluetooth performance in an automotive environment is discussed in [BS], while the automotive industry envisions such potential for wireless networks, that the Bluetooth Special Interest Group has created the Bluetooth Car Profile Working Group [1], [JKKG] with the sole task of building profiles to cover conceivable wireless scenarios in a car environment.

3.1.2 Medical applications

Pärkkä *et al.* [PGTLK] use Bluetooth for data transfer between measurement devices, a home server and a number of mobile terminals. The authors' goal is the provision of an easy-to-use self-monitoring system for weight control. A prototype of the system is presented, but no usability tests are performed to assess the applicability of the concept as a useful personal health management unit. Andreasson *et al.* [AEFCJ] present a Bluetooth system for remote patient monitoring, while [OSD] develop a similar wearable system for the monitoring of daily healthcare. Both systems utilise a point-to-point Bluetooth link, limiting the systems' capabilities to a relatively small number of measured parameters. An overview of future application areas in a tele-radiology environment is given by [Hu], but no insights are given as to exactly how Bluetooth will be used in the telemedicine industry. This last point is covered by Khoór *et al.* [KNFK], where Bluetooth was used to communicate short- and long-term digitised electrocardiograms (ECGs) along with other relevant clinical data, for the management of patients, to a web server via a GSM phone modem. The prototype was thoroughly tested and its usability proved through extensive real-life testing in hospital wards.

3.1.3 Home networking applications

A home piconet application is presented by Keller *et al.* [KPM], where Bluetooth is used as part of a system designed for the integrated control of different home devices. This system architecture seems promising, but the lack of a hardware design leaves unanswered the question of Bluetooth suitability in a home environment, especially where interference from other sources such as microwave ovens, baby monitors etc. is a limiting factor. The solution to home automation using Bluetooth, presented by Sriskanthan *et al.* [STK] requires further hardware testing to prove its functionality. Here, the authors propose a system, containing a single remote, mobile host controller and several client modules (home appliances). Bluetooth is used for inter-device communication and a protocol is devised to accommodate for the uniqueness of the home automation scenario. A limited point-to-point practical set-up is used to demonstrate the functionality of the protocol.

3.1.4 Multimedia applications

Khan *et al.* [KWR] present a Bluetooth-based design for multimedia communication. The proposed short-range multimedia radio link design is subsequently tested in a laboratory environment with extensive bit error rate (BER) measurements taken under different application scenarios. The results appear inconclusive, since no measurements are performed to test the applicability of Bluetooth to the streaming of different types of multimedia (e.g. MPEG 4, MPEG 2, etc.). This issue is examined by Kapoor *et al.* [KKGJ], who evaluate the efficacy of the Bluetooth technology in supporting ad hoc indoor multimedia communications. Through extensive simulations the authors conclude that Bluetooth performs very well in mixed data and real time traffic scenarios. A quick investigation into the opportunities available to Bluetooth for applications in the multimedia and mass media sectors is presented in [O].

3.1.5 Other application areas

An embedded Bluetooth CCD camera used in a video surveillance application scenario was developed in [Y], while Bigioi *et al.* [BISC] develop a complete solution to linking a Bluetooth camera to the Internet. Both these are point-to-point Bluetooth applications and do not concentrate on the added complexity of a piconet. Other Bluetooth application areas have included a feasibility study into the use of Bluetooth in an industrial environment [BW] and robotics [YV], where both concluded that Bluetooth has the potential to offer a wireless transmission link even in relatively harsh environments.

During CeBIT 2001 a team of experts [K] managed to show the usability of Bluetooth in a large room conference scenario. Over 350 Bluetooth devices where used in a 25,000 m^2 area, 18 m high, with 130 base stations needed to ensure total connectivity. The proposed architecture was, thus, tested in a live scenario with respect to scalability and mobility. Even though the experiment was successful and a number of problems solved, the author fails to give quantitative measurements of connection time, bit error rate and total cost. However, the experiment proved beyond doubt the applicability of Bluetooth to large-scale real life scenarios.

In contrast to our work, none of the above-mentioned works have physically implemented a Bluetooth piconet in a conference scenario, simultaneously providing access to the Internet. No published research exists on this application area and with companies reluctant to share information, it is hard to say to what extent this scenario has actually been implemented.

3.2 Quantification of interference between Bluetooth and IEEE 802.11

While a number of research projects have attempted to model and subsequently simulate interference issues between Bluetooth PANs and 802.11 WLANs (e.g. [CR], [FG], [GDS], [H], [SAE], and [Z]), not a lot of published work exists on evaluating interference through practical experimentation. Punnoose *et al.* [PTS] perform experiments in order to quantify the effect of Bluetooth on IEEE 802.11b. Similarly, Chandrashekhar *et al.* [CCMSP] conduct actual radio interference experiments in a typical office environment. Both conclude that Bluetooth has an important detrimental effect on IEEE 802.11, but again more detailed experiments are required in order to fully quantify the interference problems. As opposed to our work, however, both [PTS] and [CCMSP] only consider the effect of a single Bluetooth link on a single IEEE 802.11 link. In this project, we will attempt to quantify the effects of a Bluetooth piconet on an IEEE 802.11 WLAN.

CHAPTER 4: TOOLS & TECHNIQUES

There are two main areas to consider when developing and implementing a Bluetooth system:

- Hardware; radio, baseband etc.
- Software; to drive and control the hardware as defined in the Bluetooth specification.

4.1 Bluetooth Hardware

There are many companies developing "Bluetooth Chips" offering numerous solutions. These range from separate radio and baseband to modules incorporating the radio, baseband, micro-controller and flash memory in a single-chip solution. Each has its own advantages and disadvantages, and depend on the system architecture i.e. hosted system or embedded solution.

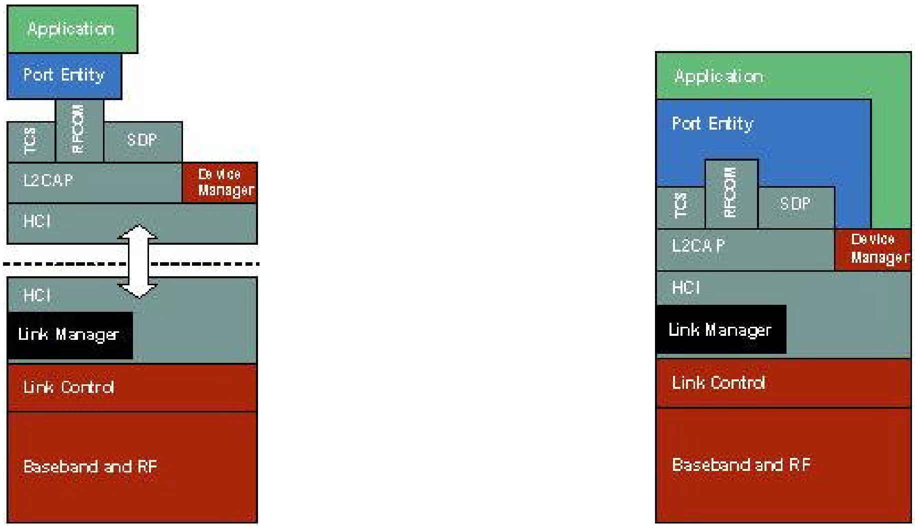


Figure 4-1: System Implementations; (a) with host e.g. laptop (b) fully - embedded solution

In the hosted model (see figure 4-1) the lower protocol stack layers reside on the Bluetooth device and the upper layers on a host e.g. PC. The hosted model is better suited for applications where a powerful host processor and memory are already available. For example, PCMCIA cards, USB, Serial-port dongles that attach to a PC.

In the fully embedded model the complete protocol stack and application reside on the Bluetooth device, this is used when there is no need for complex processing and the whole stack can run on a single micro-processor within the Bluetooth chip setup. A Bluetooth headset is an example of an embedded device.

4.2 Bluetooth Protocol Stack Software

BlueStack is a software implementation in 'C' of the Bluetooth protocol stack developed by Mezoe™. It has been designed to be applicable to a wide range of Bluetooth applications. BlueStack is supplied with the following elements (Referring back to figure 2-1 it can be seen that only the higher protocol stack layers are implemented in 'C' and are part of BlueStack.):

- Device Manager

- Service Discovery Protocol
- L2CAP
- RFCOMM
- Scheduler services

The BlueStack operating environment services and the BlueStack APIs (Application Programming Interfaces) can be accessed via the Interface Library functions.

For BlueStack, the Bluetooth protocol model has been expanded to include the Device Manager; this is part of the management entity. The Device Manager is a collection of functions that relate to the management aspects of Bluetooth. These functions include resource management, which is responsible for ensuring that the available resources (typically at the air interface) are not over-allocated to the different applications, which can all make service requests independently of each other [6]. The Device Manager API is provided to allow more sophisticated control of the Bluetooth stack. For example, Bluetooth Inquiries should be handled via this API.

Appendix II discusses software that allows us to see the messages being passed up and down BlueStack, this is very useful in measuring bit error rates and latency.

4.3 Our Bluetooth System

This project will use Bluetooth modules provided by Cambridge Silicon Radio (CSR) and BlueStack. The modules comprise of the radio, baseband, built in micro-controller and flash-memory. The lower layers of the protocol stack Radio, Baseband, Link Controller and Link Manager come pre-installed on the modules. Using Bluestack the higher protocol layers L2CAP, RFCOMM and SDP will run on a host (laptop or PC) and be managed by the application (explained in the next chapter). Our application requires a great deal of processing and hence we will be using laptops/PCs as hosts.

4.4 Cost Description

Below is the cost of the resources that will be used in this project:

Quantity	*Description*	*Cost per unit (£)*	*Total Cost (£)*
5 x	Bluetooth Module: Radio and Baseband IC Only	12	60
5 x	Flash Memory, Oscillator	8	40
1 x	BlueStack API (Non-commercial use)	FREE	FREE
		TOTAL COST	**£100**

CHAPTER 5: PROJECT DESIGN

5.1 Project Aim

Create a Bluetooth piconet application, which sets up and manages the piconet as well as offer file transfer, whiteboard applications and LAN connectivity. Quantitavely test and evaluate the efficiency and robustness of our Bluetooth piconet system by comparing to simulations and testing under different load conditions. Only recently have laptops and PCs been manufactured with Bluetooth built-in and therefore it is necessary to incorporate existing products with Bluetooth technology. All hardware attachments will be provided by our industrial sponsors: CSR.

If time permits the project will be extended to test the interference effects of IEEE 802.11b on Bluetooth. By introducing our Bluetooth network into a fully operating IEEE 802.11b environment, the effect of interference can be experimentally measured and the possibilities of coexistence investigated.

5.2 Project Objective

The major goal behind this project lies in the implementation of a complete Bluetooth solution to enable wireless data transfer between a number of Bluetooth-enabled laptop PCs, simultaneously permitting access to the Internet through a LAN - as envisioned in a typical office conference scenario (figure 5-1).

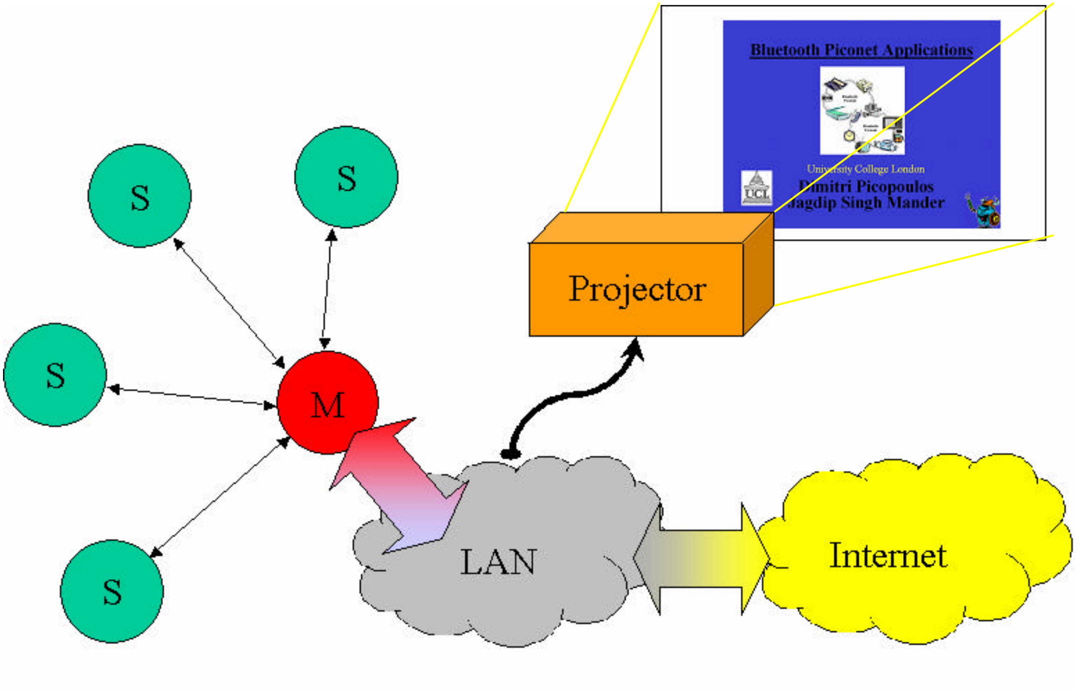


Figure 5-1: Proposed system – Bluetooth Piconet integrating with existing LAN.

Current solutions require the use of wires to permit total connectivity or the need for relatively expensive WLAN access hardware. Bluetooth, however, is ideal for the office environment and the ability of the technology to create *ad hoc* networks *on the fly* will reduce connection delays, alleviate expensive hardware and wires, while permitting total connectivity at a minimum of cost.

Other advantages include less disruption for system administrators as they need to only configure the Master PC to the LAN and as the other devices will connect through the Master to access the internet/intranet there will be no need to assign IP addresses or increased security issues. The system will be simple to use and will incorporate many existing applications, business card transfer, file transfer and whiteboard into a single user interface. It will add value to these existing applications by automatically sending the business card (using PUSH technology) and automatically connecting to the other users. As explained in chapter 2 a piconet can incorporate a maximum of 7 *active* slaves. If more than 7 devices are to be connected the extra devices will still connect into the piconet, but will be placed into low power mode (*parked*). When a *parked* device wants to actively participate in the network it does so by exchanging roles with the least *active* device.

5.3 Methods of Testing and Evaluation

Success of our Bluetooth piconet can be evaluated by measuring and investigating the following attributes:

- Total Connection Time
    - Discovering devices in the area
    - Connecting the devices to form the piconet
- Quality of Service (QoS)
    - Data Rate
    - Bit Error Rate
    - Latency
- Discuss/Investigate sources of interference
    - Introduce our piconet into areas of interference and measure the total connection time and QoS
    - Investigate co-existence with IEEE 802.11b (time permitting)
    - Discuss other sources of noise in the ISM band

Appendix II discusses software that allows us to see the messages being passed up and down BlueStack, this is very useful in measuring bit error rates and latency.

CHAPTER 6: PROJECT MANAGEMENT

The section describes how the project is divided and assigned to each member, sets milestones and deliverables and includes a Gantt chart that shows the interrelation of the two.

6.1 Important Project Dates

This MEng project is 1.5 Units (37.5%) of the Final 4th year. The University working year is 23 weeks, 37.5% of this about 8.5 weeks which is 60 days including weekends per person. There are two members in the project group, Jagdip Mander (JM) and Dimitri Picopoulos (DP) and each have different university commitments and schedules. It is important to define periods during the year when either/both members will be unavailable due to further commitments, such as modular courses at university. These have been entered into the project plan.

The initial project work was started over the summer but the time chart shows work from the 30th September onwards. This project proposal is due on the 28th October, a progress report is due on 17th January and the Final Project Report will be submitted around 28th March. There will also be a presentation given to the university and industrial supervisors as well as academic staff at the end of the year. These deadlines have been entered into the project plan.

6.2 Project Tasks

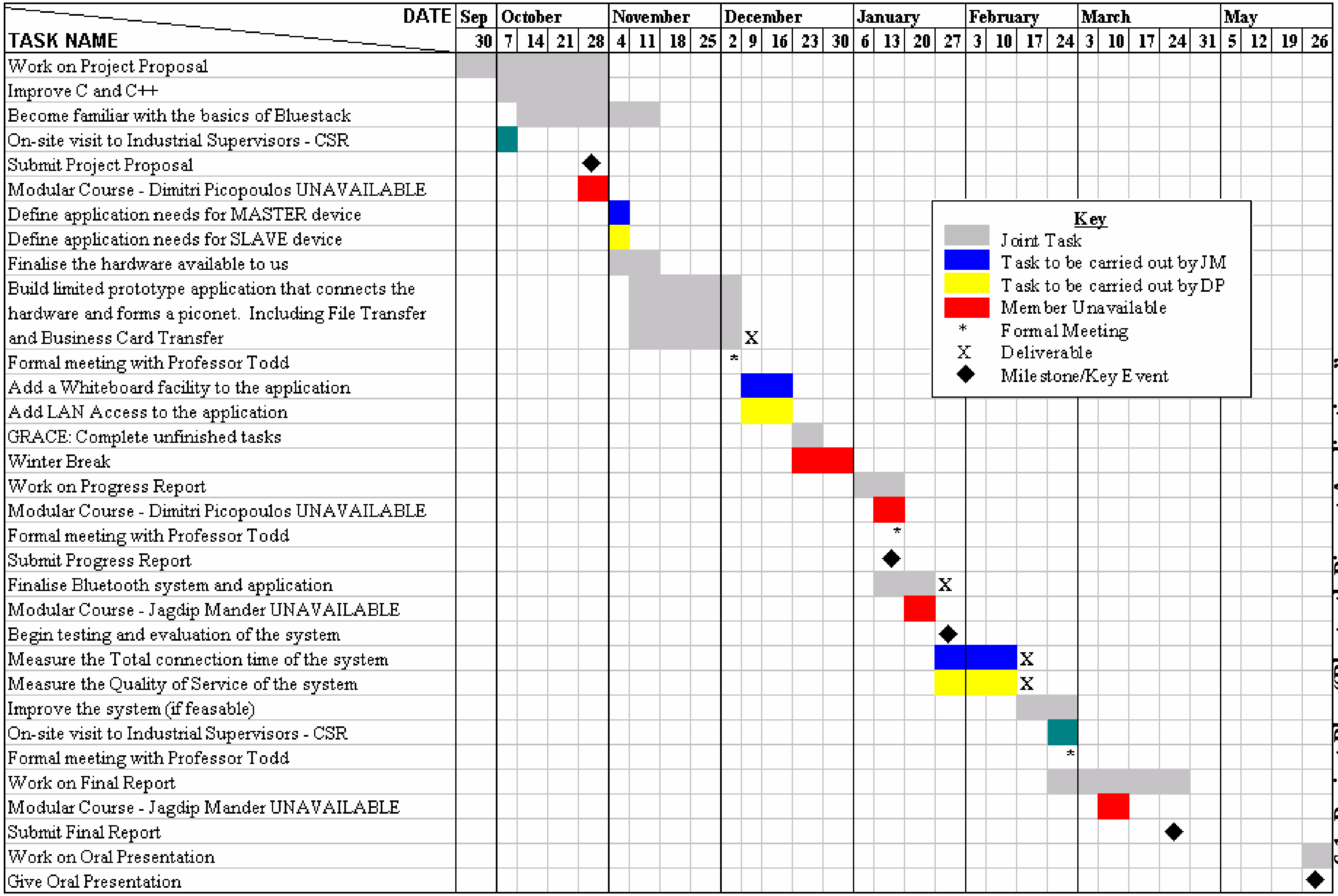
The aim of this project is to design, build and test a Bluetooth piconet application. This involves deciding on hardware and protocol stack software that will be suited to our needs. CSR's Bluetooth modules and protocol software package will be used and they will also be the industrial sponsors of this project.

The initial tasks will be carried out by both members of the team; Learning about BlueStack, defining the needs of the application and learning about the hardware. Additionally setting up the piconet and getting a simple application to run over it will be worked on together simultaneously. JM will concentrate on working with the Master device and DP will work on the Slave devices. Building a prototype system by 9th December depends on the hardware being ready to use and work with. This is shown in figure 6-1, the project plan.

Once this has been achieved the next step is to create a more advanced application incorporating the Whiteboard (JM) and LAN (DP) applications. This will take us up to the New Year and a short break has been planned during this time as well as a spare time-slot to finish earlier tasks that are not fully completed.

By the end of January the piconet application should be completed and be ready for testing. JM will measure the Total Connection Time of the system and evaluate the results. DP will measure the Quality of Service i.e. Data Rate, Latency etc and present the findings. According to this plan enough time will be available to complete these tasks and write a report describing the work and results in its entirety by 28th March.

The next chapter looks at the risks in a project this size and includes a contingency plan.



DATE / TASK NAME	Sep	October				November				December					January				February				March					May			
	30	7	14	21	28	4	11	18	25	2	9	16	23	30	6	13	20	27	3	10	17	24	3	10	17	24	31	5	12	19	26
Work on Project Proposal	▒																														
Improve C and C++		▒	▒	▒	▒																										
Become familiar with the basics of Bluestack			▒	▒	▒	▒	▒																								
On-site visit to Industrial Supervisors - CSR		🟩																													
Submit Project Proposal					◆																										
Modular Course - Dimitri Picopoulos UNAVAILABLE					🟥																										
Define application needs for MASTER device						🟦																									
Define application needs for SLAVE device						🟨																									
Finalise the hardware available to us						▒	▒																								
Build limited prototype application that connects the hardware and forms a piconet. Including File Transfer and Business Card Transfer							▒	▒	▒	▒	X																				
Formal meeting with Professor Todd										*																					
Add a Whiteboard facility to the application											🟦	🟦																			
Add LAN Access to the application											🟨	🟨																			
GRACE: Complete unfinished tasks													▒																		
Winter Break													🟥	🟥																	
Work on Progress Report															▒	▒															
Modular Course - Dimitri Picopoulos UNAVAILABLE																🟥															
Formal meeting with Professor Todd																*															
Submit Progress Report																◆															
Finalise Bluetooth system and application																▒	▒	X													
Modular Course - Jagdip Mander UNAVAILABLE																	🟥														
Begin testing and evaluation of the system																		◆													
Measure the Total connection time of the system																		🟦	🟦	🟦	X										
Measure the Quality of Service of the system																		🟨	🟨	🟨	X										
Improve the system (if feasable)																					▒	▒									
On-site visit to Industrial Supervisors - CSR																						🟩									
Formal meeting with Professor Todd																						*									
Work on Final Report																						▒	▒	▒	▒	▒					
Modular Course - Jagdip Mander UNAVAILABLE																								🟥							
Submit Final Report																										◆					
Work on Oral Presentation																															▒
Give Oral Presentation																															◆

Key

▒	Joint Task
🟦	Task to be carried out by JM
🟨	Task to be carried out by DP
🟥	Member Unavailable
*	Formal Meeting
X	Deliverable
◆	Milestone/Key Event

Figure 6-1: Project Plan “Bluetooth Piconet Applications”.

CHAPTER 7: RISK ASSESSMENT

7.1 Technical risk assessment

Drawing up contingency plans before starting to work on the project to face potential technical risks is important. We have highlighted the following risks:

1. CSR does not provide us with hardware, or equipment is faulty or gets damaged. Back-up equipment does exist through surplus from Jagdip Mander's work on Bluetooth during his 3rd year as an undergraduate. Currently this amounts to five Bluetooth modules, which connect to PCs through the serial port. Further hardware could be constructed if necessary.
2. Problems are encountered when manipulating the BlueStack software. The industrial sponsors have promised expert advice and help with this particular software package if problems prove too difficult to solve.
3. Illness. The project plan has included spare time slots to complete work if one of the members were unavailable for any reason. If either member is unavailable for a long period of time (i.e. 3 weeks) an emergency meeting will be called with Professor Todd to discuss what can be done.

7.2 Safety assessment

This project is principally software-based with a minimum of risk to personal safety. Regular 10-minute breaks will be intended to ease eyestrain and reduce the effects of radiation from display monitors. Similarly, maintaining a correct sitting posture will reduce the risk of back and spine problems due to prolonged hours being sited in front of computer screens.

Device handling will be required when setting up the Bluetooth piconet for experimentation and demonstrations. The equipment to be provided by CSR is all on a loan basis, hence the uttermost attention has to be paid to avoid damaging it. Careful device handling is essential to both personal safety and equipment survival.

Likely risks have been considered and it can be confirmed that they are covered by the following Codes of Practice:

1. SMOKING, EATING AND DRINKING, http://www.ee.ucl.ac.uk/~omed/smoke.html, Department Safety Page.
2. AFTER HOURS AND LONE WORKING, http://www.ee.ucl.ac.uk/~omed/A2.html, Department Safety Page.
3. ACCIDENT REPORTING, http://www.ee.ucl.ac.uk/~omed/A1.html, Department Safety Page.
4. FIRE, http://www.ee.ucl.ac.uk/~omed/fire.html, Department Safety Page.

CHAPTER 8: CONCLUSION

This project focuses on Bluetooth, a promising new wireless technology designed for short-range ad hoc connections. The work is intended to highlight the suitability of the technology in a typical office environment. In particular, we envision a conference scenario, where a number of laptop PCs wish to form a network providing total connectivity in a minimum amount of time and hassle. Information, such as business cards and files, can be broadcast to all participating nodes or to a limited number of devices. A whiteboard is set up for all to use and access to the Internet through a LAN is available. This Bluetooth piconet will be implemented in hardware using Bluetooth-enabled laptop PCs and extensive testing will be undertaken in order to prove Bluetooth applicability. BlueStack will be used to provide for link formation, while the GUI will be coded using C++. Careful time planning has shown that this project can be performed within the time constraints imposed by the UCL Electronic Engineering department. We are confident that this project will prove a success and enable us to physically implement a Bluetooth piconet application, putting both our hardware and software skills to the ultimate test: the fabrication of an application to meet a real-life situation.

REFERENCES

[AEFCJ] Jens Andreasson, Mikael Ekström, Ali Fard, Javier Garcia Castano, Tord Johnson. *P1 – 30: Remote System for Patient Monitoring Using Bluetooth.* In Proceedings of 1st IEEE International Conference on Sensors, Orlando, Florida, June 2002, pages 304 – 307.

[BISC] Petronel Bigioi, Adrian Ionas, George Susanu, Peter Corcoran. *Connectivity Solution to Link a Bluetooth Camera to the Internet.* In Proceedings of IEEE Consumer Electronics, Volume 47, Issue 9, August 2001, pages 294 – 299.

[BS] Hugh Burchett, Andrew Stirling. *Implementing Bluetooth Connectivity in the Automotive Environment.* ERA Conference: Vehicle Electronic Systems 2000, June 2000.

[BW] Urban Bilstrup, Per – Arne Wiberg. *Bluetooth in Industrial Environment.* In Proceedings of IEEE International Workshop on Factory Communication Systems 2000, 2000, pages 239 – 246.

[CCMSP] M.V.S Chandrashekhar, Pedro Choi, Keith Maver, Robert Sieber, Kaveh Pahlavan. *Evaluation of Interference Between IEEE 802.11b and Bluetooth in a Typical Office Environment.* In Proceedings of IEEE PIMRC' 2001, San Diego, Sep. 30 – Oct 3 2001, pages 71 – 75.

[CR] Carla F. Chiasserini, Ramesh R. Rao. *Performance of IEEE 802.11 WLANs in a Bluetooth Environment.* In Proceedings of IEEE Wireless Communications and Networking Conference, WCNC 2000, Chicago IL, September 2000, pages 94 – 99

[FG] Michael Fainberg, David Goodman. *Analysis of the Interference Between IEEE 802.11b and Bluetooth Systems.* In Proceedings of IEEE VTC 2001, October 2001, pages 967 – 971.

[GDS] N. Golmie, R. E. Van Dyck, A. Soltanian. *Interference of Bluetooth and IEEE 802.11: Simulation Modeling and Performance Evaluation.* In Proceedings of 4th ACM International Workshop on Modelling, Analysis and Simulation of Wireless and Mobile Systems 2001, Rome, Italy, July 2001, pages 11 – 18.

[H] Ivan Howitt. *IEEE 802.11 and Bluetooth Coexistence Analysis Methodology.* In Proceedings of IEEE 53rd Vehicular Technology Conference '01, May 2001, pages 1114 – 1118.

[Hu] Angela M. Hura. *Bluetooth – Enabled Teleradiology: Applications and Complications.* In Journal of Digital Imaging: the Official Journal of the Society for Computer Applications in Radiology, Volume 15, Supplement 1, 2002, pages 221 – 223.

[JKKG] Per Johansson, Manthos Kazantzidis, Rohit Kapoor, Mario Gerla. *Bluetooth: An Enabler for Personal Area Networking.* In Proceedings of IEEE Network Special Issue on Personal Area Networks, September – October 2001, pages 28 – 37.

[K] Rolf Kraemer. *Bluetooth Based Wireless Internet Applications for Indoor Hot Spots: Experience of a Successful Experiment During CeBIT 2001.* In Proceedings of 26th Annual IEEE Conference on Local Computer Networks 2001 (LCN 2001), 2001, pages 518 – 524.

[KKGJ] Rohit Kapoor, Manthos Kazantzidis, Mario Gerla, Per Johansson. *Multimedia Support Over Bluetooth Piconets.* In Proceedings of 1st Workshop on Wireless Mobile Internet 2001, Rome, Italy, 2001, pages 50 – 55.

[KMSB] David Kammer, Gordon McNutt, Brian Senese, Jennifer Bray. *Bluetooth Application Developer's Guide: The Short Range Interconnect Solution.* Syngress, 2002.

[KNFK] S. Khoór, J. Nieberl, K. Fügedi, E. Kail. *Telemedicine ECG – Telemetry with Bluetooth technology.* In Proceedings of Computers in Cardiology 2001, 2001, pages 585 – 588.

[KPM] Tomasz Keller, Rajmund Paczkowski, Józef Modelski. *Using Bluetooth in a System for Integrated Control of Home Digital Network Devices.* In Proceedings of 14th International Conference on Microwaves, Radar and Wireless Communications 2002 (MIKON – 2002), Volume 1, 2002, pages 199 – 202.

[KWR] J. Y. Khan, J. Wall, M. A. Rashid. *Bluetooth – Based Wireless Personal Area Network for Multimedia Communication.* In Proceedings of 1st IEEE International Workshop on Electronic Design, Test and Applications (DELTA '02), 2002, pages 47 – 51.

[MWF] Patrick Murphy, Erik Welsh, J. Patrick Frantz. *Using Bluetooth for Short – Term Ad Hoc Connection Between Moving Vehicles: A Feasibility Study.* In Proceedings of 55th IEEE Vehicular Technology Conference 2002 (VTC Spring '02), Volume 1, 2002, pages 414 – 418.

[NP] René Nüsser, Rodolfo Mann Pelz. *Bluetooth – Based Wireless Connectivity in an Automotive Environment.* In Proceedings of 52nd IEEE Vehicular Technology Conference 2000 (VTC Fall '00), Volume 4 , 2000, pages 1935 – 1942.

[O] Ville Ollikainen. *Bluetooth Applications in New Media Technology.* White paper, available [Online] http://www.cs.hut.fi/Opinnot/Tik-86.174/ [October 2002].

[OSD] Kazushige Ouchi, Takuji Suzuki, Miwako Doi. *LifeMinder: A Wearable Healthcare Support System Using User's Context.* In Proceedings of 22nd International Conference on Distributed Computing Systems Workshops 2002, 2002, pages 791 – 792.

[PGTLK] Juha Pärkkä, Mark van Gils, Tino Tuomisto, Raimo Lappalainen, Ilkka Korhonen. *A Wireless Wellness Monitor for Personal Weight Management.* In Proceedings of 2000 IEEE EMBS International Conference on Information Technology Applications in Biomedicine, 2000, pages 83 – 88.

[PTS] Ratish J. Punnoose, Richard S. Tseng, Daniel D. Stancil. *Experimental Results for Interference between Bluetooth and IEEE 802.11b DSSS Systems.* In Proceedings of IEEE Vehicular Society Conference, October 2001.

[SAE] Steven Selby, Afshin Amini, Cory Edelman. *Simulating Interference Issues between Bluetooth PANs and 802.11b and 802.11g WLANs.* White paper, available [Online]: http://www.itwireless.com [October 2002].

[SSN] Liisa – Maija Sainio, Taina Sikiö, Jukka Niiranen. *Application Visions and Business Opportunities of Bluetooth – A Wireless Technology for Local Data Transfer.* White paper, available [Online]: http://www.tbrc.fi/publications/data/.

[STK] N. Sriskanthan, F. Tan, A. Karande. *Bluetooth Based Home Automation System.* In Proceedings of Microprocessors and Microsystems, Volume 26, Issue 6, 10 August 2002, pages 281 – 289.

[Y] Ko Sung – Yuan. *The Embedded Bluetooth CCD Camera.* In Proceedings of IEEE Region 10 International Conference on Electrical and Electronic Technology 2001 (TENCON '01), Volume 1, 2001, pages 81 – 84.

[YV] Jian Y. Yan, Ljubo Vlacic. *Suitability of Bluetooth Technology in Communication Between Autonomous Mobile Robots.* In Proceedings of Microelectronic Engineering Research Conference 2001, Brisbane, Australia, 2001.

[Z] Jim Zyren. *Reliability of IEEE 802.11 Hi Rate DSSS WLANs in a High Density Bluetooth Environment.* White paper, Bluetooth '99, June 1999.

BIBLIOGRAPHY

[1] Bluetooth SIG. *Specification of the Bluetooth System – Wireless Connections Made Easy*. Available [Online]: http://www.bluetooth.com [October 2002].
[2] Jennifer Bray, Charles F. Sturman. *Bluetooth: Connect Without Cables.* Prentice Hall, 2001.
[3] Jaap C. Haartsen, Sven Mattisson. *Bluetooth – A New Low – Power Radio Interface Providing Short – Range Connectivity.* In Proceedings of the IEEE, Volume 88, Number 10, October 2000, pages 1651 – 1661.
[4] Brent A. Miller, Chatschik Bisdikian. *Bluetooth Revealed: The Insider's Guide to an Open Specification for Global Wireless Communications.* Prentice Hall, 2000.
[5] Nathan J. Muller. *Bluetooth Demystified.* McGraw – Hill, 2000.
[6] Mezoe. *BlueStack User Manual.* Available [Online]: http://www.mezoe.com [November 2001]
[7] Mezoe. *A Brief Introduction to BlueStack*. Available [Online]: http://www.mezoe.com [March 2002]

ACRONYMS

ACL:	Asynchronous Connectionless
AM:	Active Member
AM_ADDR:	Active Member Address
API:	Application Programming Interface
BD_ADDR:	Bluetooth Device Address
BER:	Bit Error Rate
CCD:	Charge-Coupled Device
CCL:	Cambridge Consultants Limited
CD:	Compact Disc
CLK:	The Master clock
CRC:	Cyclic Redundancy Checksum
CSR:	Cambridge Silicon Radio
DH:	Data High (Rate)
DM:	Data Medium (Rate)
ECG:	Electrocardiogram
FEC:	Forward Error Correction
FHS:	Frequency Hop Synchronisation (packet)
FHSS:	Frequency Hop Spread Spectrum
GFSK:	Gaussian Frequency Shift Keying
GIAC:	General Inquiry Access Code
GSM:	Global System for Mobile Communications
HCI:	Host Controller Interface
IAC:	Inquiry Access Code
IEEE:	Institute of Electrical and Electronic Engineers
IP:	Internet Protocol
IrDA:	Infrared Data Association
ISM:	Industrial Scientific and Medical (band)
L2CAP:	Logical Link Control and Adaptation Protocol
LAN:	Local Area Network
LC:	Link Controller
LM:	Link Manager
MAC:	Medium Access Control
MPEG-2:	Motion Picture Expert Group-2
MP3:	MPEG-1 Audio Layer-3
MPEG-4:	Motion Picture Expert Group-4
OBEX:	OBject EXchange (protocol)
PAN:	Personal Area Network
PC:	Personal Computer
PCMCIA:	Personal Computer Memory Card International Association
PDA:	Personal Digital Assistant
QoS:	Quality of Service
SCO:	Synchronous Connection Oriented
SDP:	Service Discovery Protocol
SIG:	(Bluetooth) Special Interest Group
TCS:	Telephony Control (Protocol) Specification
TDM:	Time Division Multiplexing
USB:	Universal Serial Bus
WAP:	Wireless Application Protocol
WLAN:	Wireless Local Area Network

APPENDIX I: DETAILED BLUETOOTH BACKGROUND

To underline the importance and potential revenue of Bluetooth technology, the group of companies working together to promote and define the Bluetooth specification include; Ericsson, Intel, IBM, Toshiba and Nokia. These corporations are the core promoters and original members of the Bluetooth Special Interest Group (SIG).Currently the SIG has expanded to the size of 3000 companies. The Bluetooth Specification is an open, global specification defining the complete system from the baseband to the application level and is defined by the SIG.

There is a great need for a wireless technology that allows devices to communicate to one another without cables. Using Bluetooth, electronic devices will no longer have to plug into, install, enable or configure to connect and use any other device. Through a ubiquitous standardised communications subsystem, devices will communicate seamlessly [2].

The main advantages of Bluetooth can be related to its robustness, low complexity, low power and low cost. It must be noted that the currently widely available Infrared (IrDA) technology is based on the fact that a connection of units will only be established if the units lie within line-of-sight, thereby being the main limitation of infrared links [2]. On the other hand, Bluetooths main strength lies in its capability to handle both data and voice transmissions at the same time, thereby allowing solutions such as a mobile hands-free headset for voice calls, print to fax capability and synchronised PDA (Personal Digital Assistant).

How Bluetooth Operates

Every Bluetooth device has a unique Bluetooth device address and a 28-bit Bluetooth clock. The baseband part of the Bluetooth System uses a special algorithm, which calculates the frequency hop sequence from the masters clock and device address which can be found from the FHS packet (Frequency Hop Synchronisation).

The specification limits the number of Slaves in a Piconet to seven, however larger networks can be set up using many Piconets to communicate to one another, this is known as a **Scatternet** In a Scatternet a device can be a Master in one Piconet and a Slave in another as shown in figure A-1.
(Note: that a device cannot be a Master in two Piconet as by definition it would be one Piconet)

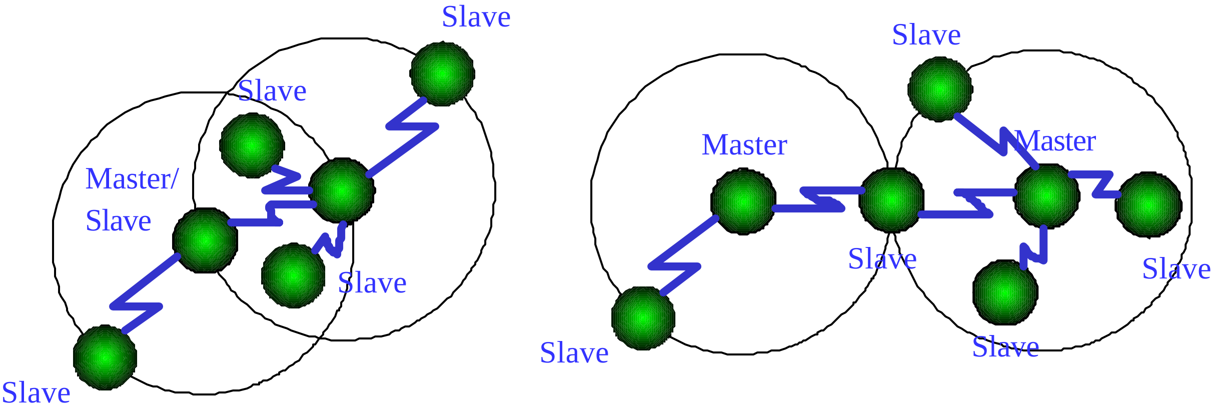


Figure A-1: Scatternets [2]

In a Scatternet, the main problem is interference from other Bluetooth devices in other Piconets, which are not aware of other Piconets in the area and so collisions will occur. The problem will escalate as more Piconets are in a given transmission area and the number of retransmission will increase, causing data rates to fall. If voice is being transmitted lost voice packets will be ignored, resulting in a poor quality voice channel.

The Bluetooth specification defines 3 power classes for radio transmitters [2] with an output power of:

Power Class	*Maximum Output Power /mW*	*Range /m*
Class 1	100	100+
Class 2	2.5	10
Class 3	1	1

Table A-1: Bluetooth Radio power classes.

The output power defines the range that the device is able to cover. It is important to realise that the range figures are for typical use. In a space of area where there is very little interference, a Class 1 device has been successfully tested at over a mile [2]. But in a crowded office the Bluetooth signal will be blocked and absorbed, resulting in a lower coverage area.

Voice and Data Links

Bluetooth allows both time critical data communication such as that required for voice or audio, as well as high speed, time insensitive packet data communication. To carry such data two different types of links are defined between any two devices. These are **SCO** (Synchronous Connection Oriented) links for voice communication and **ACL** (Asynchronous Connectionless) links for data communication [1].

ACL data packets are constructed from a 72-bit access code, a 54-bit packet header and a 16-bit CRC code, in addition to the payload data. There are a variety of packet types allowing different amounts of data to be sent. DM and DH stand for Data Medium and Data High respectively. The DH packets achieve a higher rate by using less error correction in the packet, thus leaving more room for data. The single slot packets DM1 and DH1 carry less data than the 3-slot packets DM3 and DH3 and similarly for 5-slot packets DM5 and DH5. Table A-2 shows the difference in capacity of the different packet types.

Packet Type	Max. Payload /bytes	FEC	Max. Symmetric Data Rate /kbps	Asymmetric Data Rate Forward /kbps	Asymmetric Data Rate Reverse /kbps
DM1	17	2/3	108.8	108.8	108.8
DH1	27	None	172.8	172.8	172.8
DM3	121	2/3	258.1	387.2	54.4
DH3	183	None	390.4	585.6	86.4
DM5	224	2/3	286.7	477.8	36.3
DH5	339	None	433.9	723.2	57.6

Table A-2: DM and DH packets [4]

A symmetric channel uses the same packet types in both directions. This channel type would be used when the Master and Slave need to send data at about the same rate. Often data will need to be transferred faster in one direction in one direction. For example if a PDA is browsing the web via a sever, there will be a high data rate from the server to the to the PDA as Web pages are being downloaded. In the reverse direction, only a few bytes will be needed to specify the next link to browse. For these purpose asymmetric channels should be used, figure A-2 illustrates the difference between symmetric and asymmetric channels.

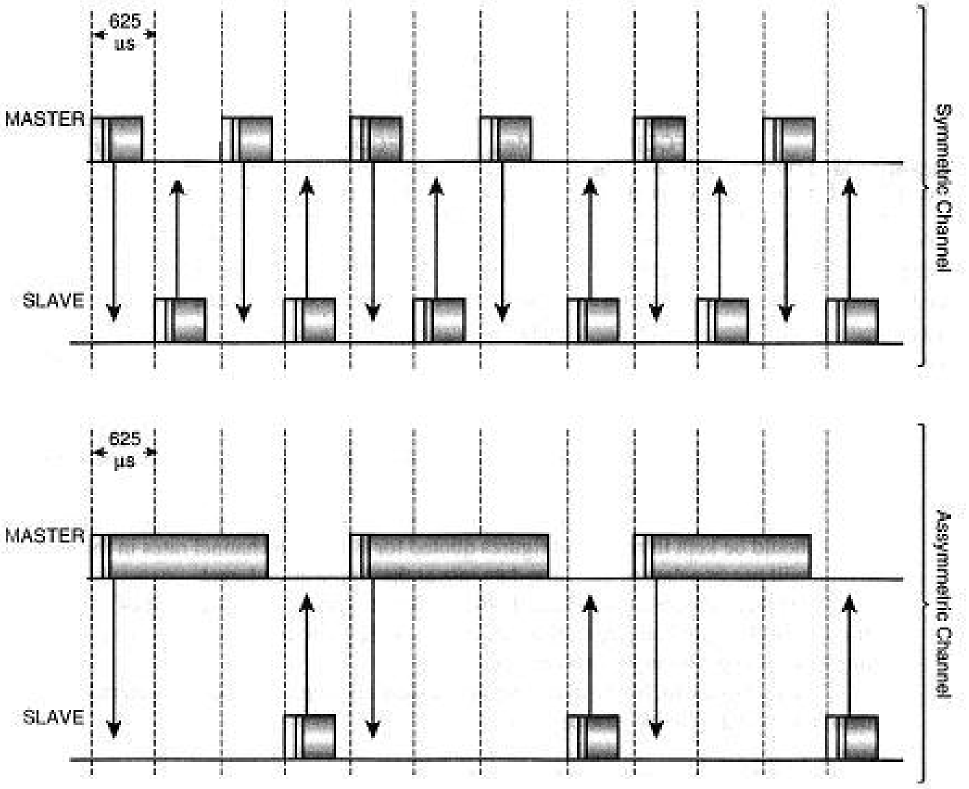


Figure A-2: Symmetric and Asymmetric channels [2]

The data rates given in Table A-2 are for the maximum number of bytes of data, which can be transferred on air. Higher protocol layers such as L2CAP and RFCOMM will use some channel capacity with headers and framing information. This is an important factor that must be taken into account when calculating data rates for applications (at the application level, the maximum data rate could be around 650 kb/s [1]). Specifically the choice of packet size will affect how efficiently packets are transferred; this can mean that asymmetric channels may not be appropriate. For example when transporting small RFCOMM packets (<27 bytes) the efficiency will decrease if packet type DH5 is used over DM1. Another consideration when choosing packet types is the likelihood of corruption. If many packets are being re-transmitted, then it is better to use short packets, as these are more likely to get through without errors.

The SCO links work at 64 kb/s and it is possible to have up to three voice links at once, or to mix voice and data [2]. The quality of these voice channels is similar to that of a GSM mobile cellular phone call.

This means that SCO links are not suitable to deliver audio quality required for music listening. A possible solution would be to use a compression technology such as MP3 which requires a transmission channel that supports a bit rate of 128 kb/s and to set up an ACL link with DH1 packet transmission. Providing the time criticality of the audio was maintained near CD-quality audio could be carried.

Bluetooth Device Address

Each Bluetooth device has a 48 bit IEEE MAC address known as the Bluetooth Device Address (BD_ADDR). It is used in Serial bit processing operation and in particularly derivation of the access code used to identify data packets. The BD_ADDR is split as follows:

- BD_ADDR[47:32] – NAP[15:0] (Non-significant address part)
  Used to initialise the encryption engine stream.
- BD_ADDR[31:24] – UAP[7:0] Upper address part)
  Used to calculate the frequency hopping sequence.
- BD_ADDR[23:0] – LAP[23:0] (Lower address part)
  Used by sync word generation and frequency hopping.

Low Power Modes

The following low power modes have been included in the Bluetooth specification [1]:

- **Standby**: no data transferred and the radio is switched off.
- **Hold**: device ceases to support ACL traffic for a defined period of time, to free bandwidth for paging etc. The device keeps its active member (AM) address and synchronises to the Master once its hold time has expired.
- **Sniff**: slave device listens for its AM address for a pre-defined slot time. It accepts traffic on its link for the time-slot only, then powers down until its next sniff slot.
- **Park**: Slave device is only active on a link during periodic beacon slots. Otherwise it gives up its AM address. During the beacon slot it can request to be un-parked and return to normal activity or can be ordered to do so by the Master.

The normal power mode is **Active**, on entry to the connection state the Slave switches to the Masters clock; the Master transmits a POLL to verify the link.

Inquiry & Paging Mechanisms

Due to the ad-hoc nature of a Bluetooth network where highly mobile devices may come into and go out of range of other communicating devices, the network topology and membership can be constantly changing. The Inquiry and Inquiry Scan mechanisms manage the process of device discovery. Figure A-3 shows a more detailed explanation of the message sequence for Inquiry and Inquiry scanning.

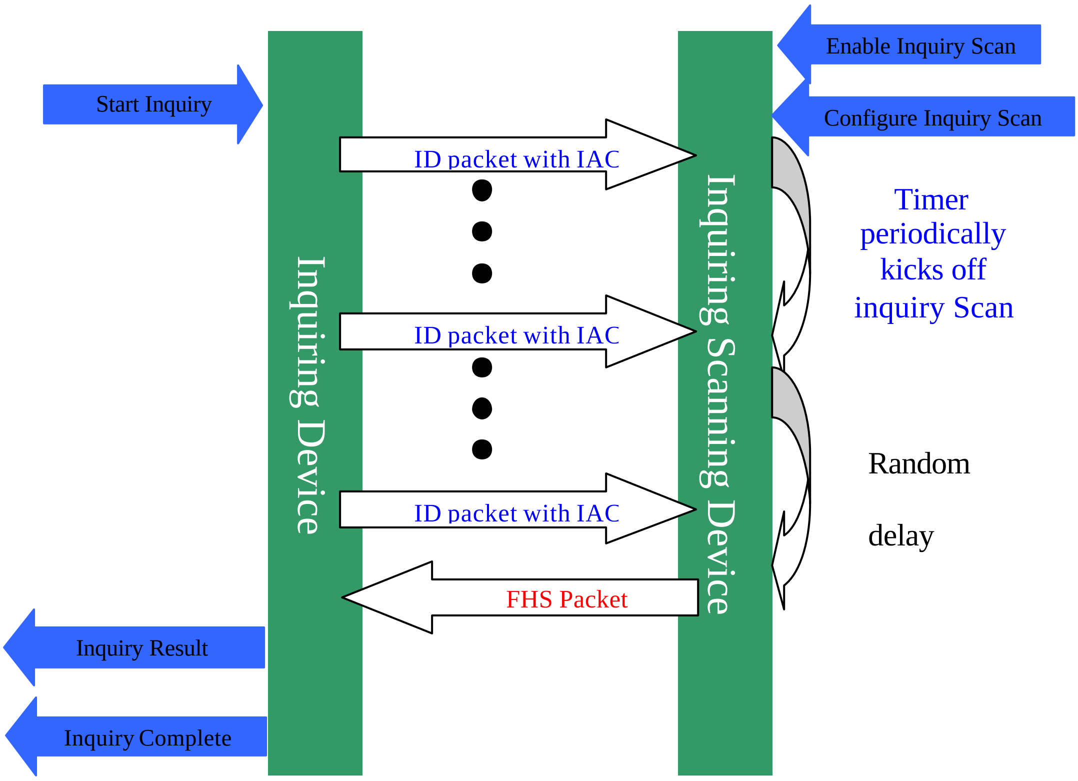


Figure A-3: Message sequence chart for Inquiry and Inquiry Scan [2].

The Inquiring device transmits ID packets, consisting of an Inquiry Access code (IAC). Usually the General Inquiry Access code (GIAC) is used, which is common between all devices performing inquiry procedures. Additionally there is a Limited IAC, designed to be used by a pair of devices for a short period of time. This would be by application level selection on both devices and provide a quick way of discovering a “known” device [2].

When an Inquiry scanning device receives an Inquiry it could respond immediately. This would result in many scanning devices all responding together the moment an inquiry was sent. The inquirer would not receive any responses, as they would all interfere with each other. Therefore a random stand off period is used, instead of responding immediately it waits for a random period of time and then re-enters the scanning state, listening once more for another ID packet. This time if it receives the ID packet it will respond with the FHS packet. In this way several devices can respond to an inquiry without interfering with each other.

It is permitted to avoid Inquiry altogether if the address of the device to connect to is already known, this is accomplished through Paging.

The link between a Master and a Slave are established by one device initiating the connection by addressing the request directly to the other device effectively saying: “Will you connect with me?” This is referred to as Paging. The other device must be listening for such requests this is referred to as Page Scanning. At the conclusion of the paging process, the pager becomes the Master and the page scanner becomes the Slave. Though in some cases this may not be suitable and can be changed by using a Master/Slave switch.

The messages exchanged between two devices during the connection establishment are shown in figure A-4 using the mandatory paging procedure.

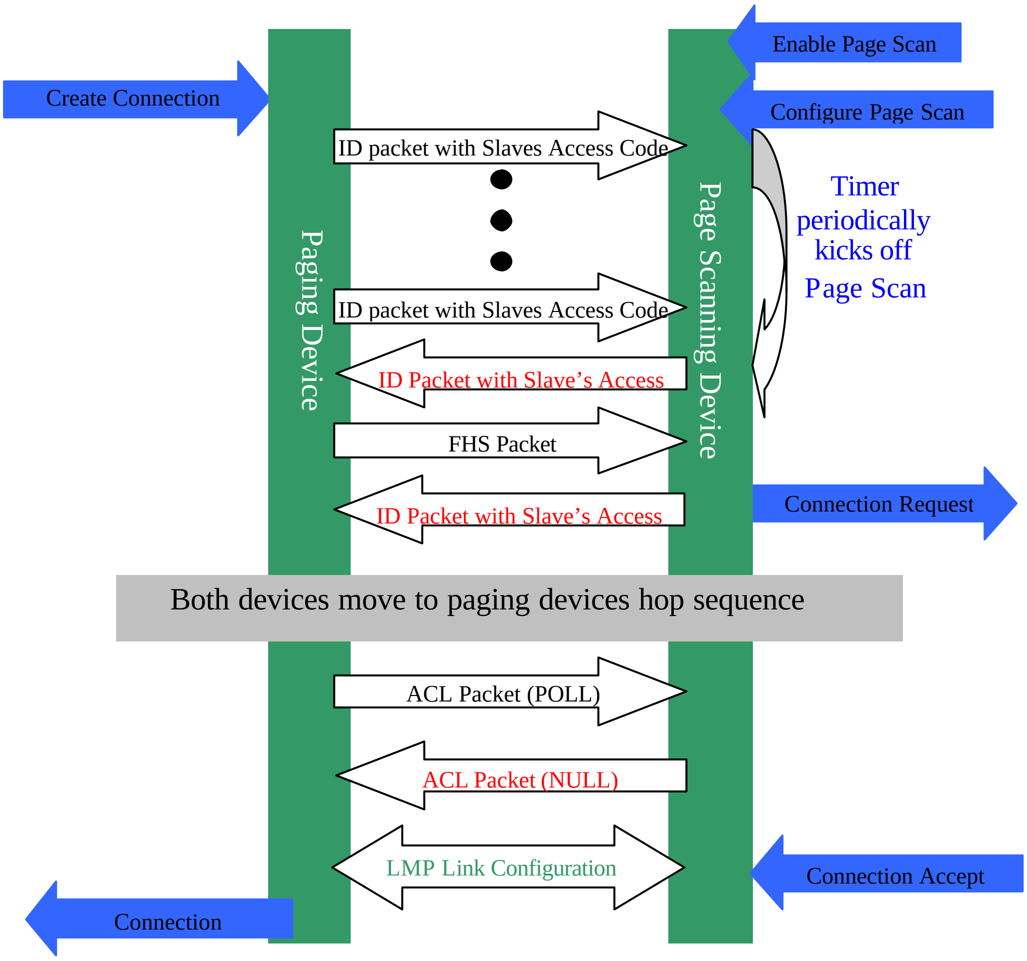

Create Connection
Enable Page Scan
Configure Page Scan
ID packet with Slaves Access Code
Paging Device
Page Scanning Device
Timer periodically kicks off Page Scan
ID packet with Slaves Access Code
ID Packet with Slave’s Access
FHS Packet
ID Packet with Slave’s Access
Connection Request
Both devices move to paging devices hop sequence
ACL Packet (POLL)
ACL Packet (NULL)
LMP Link Configuration
Connection Accept
Connection

Figure A-4: Message sequence chart for paging and page scanning [2].

The request to create a connection causes the device to enter paging mode, where it sends out a series of paging packets. The paging packet consists of ID packets based on the paged device’s address. The device in Page Scan mode periodically scans for a specified duration and at a specified interval. If the page-scanning device is scanning at the same time as the pager transmits its ID packet it will trigger and receive the ID packet. It will respond with another ID, again using its own address. Because there is only one device with that ID, only that device will respond and therefore there is no need for a random back-off delay unlike inquiry scanning. The page scanner is ready to receive the pagers FHS packet and responds with another ID packet. The page scanner can now extract the Bluetooth clock and active member address (CLK, AM_ADDR) to use in the new connection. Using these parameters the page scanner is able to calculate the pagers hop sequence, as both devices move into the connection state, the pager becomes the Master and the page scanner the Slave.

As with inquiry, the paging device hops twice per slot to maximise the chance of hitting the page scanner's frequency [3].

The master device sends a POLL packet to its slaves to check that the frequency hop sequence is correctly synchronised. The slave responds with a NULL packet. A NULL packet consists of only the access code and packet header and is used to pass acknowledgement to a device following reception of a packet, as well as for synchronisation as explained above. The NULL packet itself does not require acknowledgement. The POLL packet has the same structure as the NULL packet but must be acknowledged. Typically used by the master device to check for slaves in its piconet, which must respond if present [2&3].

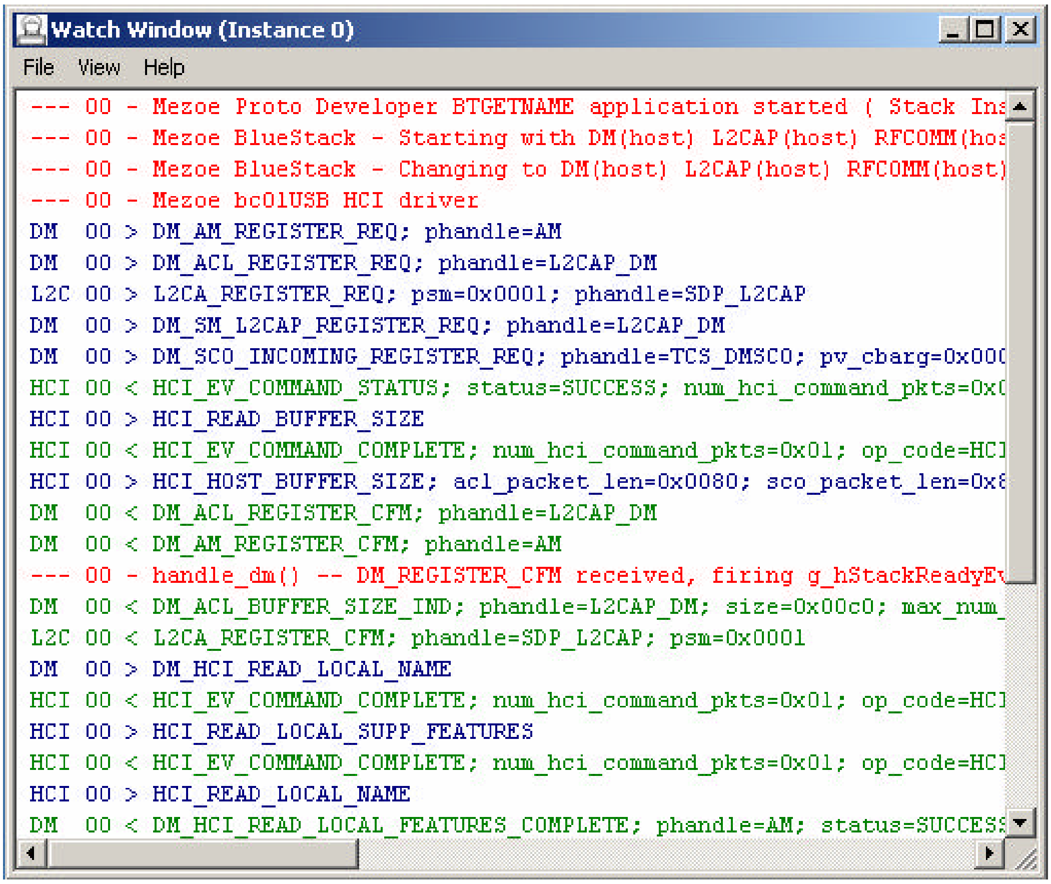
The active member address (AM_ADDR) is an address allocated by the Master to each active Slave in a piconet. The address is used to identify a particular slave a packet is intended for.

Discoverability and Connectability Modes

In order for a connection to be created both ends of the link have to be willing to connect. Some devices may be set so that they will not scan for inquiries, in that case they cannot be discovered by other devices and will be invisible. Similarly devices can be set not to perform page scans and therefore cannot be connected to. Applications can choose whether to make devices **connectable** or **discoverable** or both. Once in a connection a device is free to disconnect without warning at any time.

APPENDIX II: VIEWING MESSAGES BEING PASSED IN THE PROTOCOL STACK

An important and extremely useful support application is the Watch Window [7]. The Watch Window is a Windows application that displays detailed information on upper-level stack activity. Information is colour-coded for easy interpretation of the data. Commands in GREEN are messages being passed up the stack and commands in BLUE are messages going down the stack. RED text is for comments and information.



```
--- 00 - Mezoe Proto Developer BTGETNAME application started ( Stack Ins
--- 00 - Mezoe BlueStack - Starting with DM(host) L2CAP(host) RFCOMM(hos
--- 00 - Mezoe BlueStack - Changing to DM(host) L2CAP(host) RFCOMM(host)
--- 00 - Mezoe bc01USB HCI driver
DM  00 > DM_AM_REGISTER_REQ; phandle=AM
DM  00 > DM_ACL_REGISTER_REQ; phandle=L2CAP_DM
L2C 00 > L2CA_REGISTER_REQ; psm=0x0001; phandle=SDP_L2CAP
DM  00 > DM_SM_L2CAP_REGISTER_REQ; phandle=L2CAP_DM
DM  00 > DM_SCO_INCOMING_REGISTER_REQ; phandle=TCS_DMSCO; pv_cbarg=0x000
HCI 00 < HCI_EV_COMMAND_STATUS; status=SUCCESS; num_hci_command_pkts=0x0
HCI 00 > HCI_READ_BUFFER_SIZE
HCI 00 < HCI_EV_COMMAND_COMPLETE; num_hci_command_pkts=0x01; op_code=HCI
HCI 00 > HCI_HOST_BUFFER_SIZE; acl_packet_len=0x0080; sco_packet_len=0x8
DM  00 < DM_ACL_REGISTER_CFM; phandle=L2CAP_DM
DM  00 < DM_AM_REGISTER_CFM; phandle=AM
--- 00 - handle_dm() -- DM_REGISTER_CFM received, firing g_hStackReadyEv
DM  00 < DM_ACL_BUFFER_SIZE_IND; phandle=L2CAP_DM; size=0x00c0; max_num_
L2C 00 < L2CA_REGISTER_CFM; phandle=SDP_L2CAP; psm=0x0001
DM  00 > DM_HCI_READ_LOCAL_NAME
HCI 00 < HCI_EV_COMMAND_COMPLETE; num_hci_command_pkts=0x01; op_code=HCI
HCI 00 > HCI_READ_LOCAL_SUPP_FEATURES
HCI 00 < HCI_EV_COMMAND_COMPLETE; num_hci_command_pkts=0x01; op_code=HCI
HCI 00 > HCI_READ_LOCAL_NAME
DM  00 < DM_HCI_READ_LOCAL_FEATURES_COMPLETE; phandle=AM; status=SUCCESS
```

Figure A-5: Screen-shot showing the Watch Window running.

These messages used in conjunction with the Bluetooth Specification [1] can be used to measure and analyse the data rate of transfer between devices, the BER and packet re transmission.