Multi-channel GPRS-based mobile telemedicine system with bluetooth and J2ME interfaces

This item was submitted to Loughborough University's Institutional Repository by the/an author.

Additional Information:

- A Doctoral Thesis. Submitted in partial fulfilment of the requirements for the award of Doctor of Philosophy of Loughborough University.

Metadata Record: https://dspace.lboro.ac.uk/2134/8019

Publisher: © M.F.A. Rasid

Please cite the published version.
This item is held in Loughborough University’s Institutional Repository (https://dspace.lboro.ac.uk/) and was harvested from the British Library’s EThOS service (http://www.ethos.bl.uk/). It is made available under the following Creative Commons Licence conditions.

For the full text of this licence, please go to:
http://creativecommons.org/licenses/by-nc-nd/2.5/
MULTI-CHANNEL GPRS-BASED MOBILE
TELEMEDICINE SYSTEM WITH
BLUETOOTH AND J2ME INTERFACES

By

Mohd Fadlee A Rasid, BSc

A doctoral thesis
submitted in partial fulfilment of the requirements for the award of
Doctor of Philosophy of Loughborough University

October 2005

Department of Electronic and Electrical Engineering
Loughborough University

© M. F. A. RASID, 2005
To my wife Nora and my son Zareef, to whom I dedicate this thesis.......

ABSTRACT

One of the emerging issues in m-Health is how best to exploit the mobile communications technologies that are now almost globally available. This thesis describes a multi-channel m-Health system with a Bluetooth interface based on the General Packet Radio Service (GPRS). The challenge here is to produce a system to transmit a patient’s biomedical signals directly to a hospital using a mobile phone on a commercial GPRS network.

As greater patient mobility gradually becomes a trend in remote monitoring, the integration of medical sensors with global connectivity seems to be the next step in providing telemedicine services. The system samples signals from sensors on the patient, then transmits the incoming digital data over a Bluetooth link to a GPRS mobile phone.

The system is equipped with patient user interface programs for the patient to perform the data acquisition process from the sensors. There are two programs available, one being the patient interface on a laptop while the other is the patient interface on a mobile phone. The later interface program is developed based on Java 2 Micro Edition (J2ME) MIDlet suite application.

The system is integrated with client-server application programs to allow the monitoring and management of medical data. An application server is responsible for handling the telemedicine session and controlling the client connection request from a remote patient. All the medical data transmitted during a telemedicine session are stored in a database together with the patient information and telemedicine session details for further assessment. These data are available to clinicians as and when required, by accessing the database via browser programs.

The prototype system allowed real-world mobile tests to be carried out and provide valuable insights into real user experience with m-Health systems.
ACKNOWLEDGEMENTS

I would like to express my heartfelt and deepest gratitude to my supervisor, Professor Bryan Woodward for his continued guidance, patience, inspiration, motivation, encouragement and enthusiastic support over my years at Loughborough University.

I am grateful to the Universiti Putra Malaysia for providing sponsorship and opportunity for me to pursue an extremely interesting and challenging PhD research.

I should like to thank members of the Department of Electronic and Electrical Engineering as a whole, with special thanks towards all members of the Applied Signal Processing research group for making my three years in the department a wonderful and memorable research experience.

My appreciation goes to Lawrence Gore and Paul Atkins, whom I had the opportunity of co-supervising their MEng Advanced Project and whose works have partly contributed to this research.

Further, I should like to express my sincere gratitude to my dearest mother for her prayers and to my two brothers for looking after her throughout the years I have been away.

Last but not least, I would like to express my highest appreciation and heartfelt gratitude to my lovely wife, Nora and our first baby, Zareef, who had experienced the first three wonderful years of his childhood in Loughborough. I am thankful for their great patience and invaluable emotional support throughout my PhD years at Loughborough University.
# TABLE OF CONTENTS

**Thesis Access Form**

**Certificate of Originality**

**Abstract**

**Acknowledgements**

**Table of Contents**

**Table of Figures**

**Table of Tables**

**Glossary**

## Chapter One: Introduction 1

1.1 General Overview of m-Health Concept 1

1.2 Aim and Objectives of Research 4

1.3 Design Considerations 4

1.4 Chapter Synopsis 7

1.4.1 Chapter One: Introduction 7

1.4.2 Chapter Two: Review of m-Health Systems 7

1.4.3 Chapter Three: Mobile Bio-monitoring: Design Concepts 8

1.4.4 Chapter Four: System Hardware Design 8

1.4.5 Chapter Five: System Software and Protocols 8

1.4.6 Chapter Six: Application and Database Server 9

1.4.7 Chapter Seven: Performance Analysis and Technical Results 9

1.4.8 Chapter Eight: Conclusions and Future Research 9

1.5 Summary 9

1.6 References 10

## Chapter Two: Review of m-Health Systems 12

2.1 Wireless Technologies in m-Health 12

2.2 Wireless Medical Sensor Technologies 18

2.3 Summary 20

2.4 References 21
Chapter Three: Mobile Bio-monitoring: Design Concepts 26

3.1 Physiological Signals for Bio-monitoring 26
   3.1.1 Electrocardiogram (ECG) Signals 27
   3.1.2 Blood Pressure 31
   3.1.3 Body Temperature 31
   3.1.4 Oxygen Saturation (Spo2) 32

3.2 Short-range Wireless Communication for Bio-monitoring 33
   3.2.1 Features and Specifications 34
   3.2.2 Protocols Architecture 37

3.3 Choice of Mobile Cellular Networks 40
   3.3.1 Introduction to General Packet Radio Service (GPRS) 40
   3.3.2 GPRS Network Architecture 41
   3.3.3 GPRS Protocols 42

3.4 Summary 46
3.5 References 47

Chapter Four: System Hardware Design 51

4.1 Overview of the m-Health System 51
4.2 m-Health Processor Module 54
   4.2.1 Atmel Microcontroller Board 56
   4.2.2 Bluetooth Communication Module 58
   4.2.3 Medical Sensing Nodes 60
   4.2.4 Power Consumption and Physical Dimensions 63

4.3 Mobile Phone 65
   4.3.1 Mobile Phone Specifications 66
   4.3.2 GPRS Network and Mobile Terminal Configuration 67

4.4 Summary 71
4.5 References 72
Chapter Five: System Software and Protocols

5.1 Overview of m-Health System Software and Protocols  
5.2 m-Health Processor Module Software  
  5.2.1 Main Software Program  
  5.2.2 Bluetooth Module Configuration Program  
  5.2.3 Bluetooth Link Establishment Protocols  
5.3 GPRS and IP Protocols  
  5.3.1 GPRS Data Flow  
  5.3.2 PDP Context Activation  
5.4 Mobile Phone Application Software  
  5.4.1 J2ME Development Process  
  5.4.2 Telemedicine MIDlet Suite  
  5.4.3 Mobile Phone Limitation and Known Issues  
5.5 Summary  
5.6 References

Chapter Six: Application and Database Servers

6.1 Overview of a Client-Server Model  
  6.1.1 Development Environment  
6.2 MySQL Telemedicine Database Server  
  6.2.1 Telemedicine Database Structure  
6.3 Telemedicine Application Server  
  6.3.1 Server Protocols  
  6.3.2 Server Graphical User Interface  
6.4 Patient User Interface Program  
  6.4.1 Patient Interface on a laptop for fixed-IP or GPRS network  
  6.4.2 Patient Interface on a mobile phone for GPRS network  
6.5 Doctor Browser Interface  
  6.5.1 Doctor Browser on a Laptop  
  6.5.2 Doctor Browser on a Mobile Phone  
6.6 Security Issues  
6.7 Summary  
6.8 References
# Table of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>A complete cardiac cycle of an ideal ECG signal</td>
<td>28</td>
</tr>
<tr>
<td>3.2</td>
<td>The Einthoven’s Triangle</td>
<td>28</td>
</tr>
<tr>
<td>3.3</td>
<td>Key benefits of short-range wireless technologies</td>
<td>34</td>
</tr>
<tr>
<td>3.4</td>
<td>Bluetooth packet format</td>
<td>35</td>
</tr>
<tr>
<td>3.5</td>
<td>Bluetooth protocols stack</td>
<td>37</td>
</tr>
<tr>
<td>3.6</td>
<td>Bluetooth host-based implementation</td>
<td>39</td>
</tr>
<tr>
<td>3.7</td>
<td>Bluetooth host-less implementation</td>
<td>39</td>
</tr>
<tr>
<td>3.8</td>
<td>GPRS network architecture</td>
<td>41</td>
</tr>
<tr>
<td>3.9</td>
<td>GPRS protocol structure</td>
<td>43</td>
</tr>
<tr>
<td>4.1</td>
<td>Multi-channel GPRS-based m-Health system</td>
<td>52</td>
</tr>
<tr>
<td>4.2</td>
<td>Schematic of m-Health processor module</td>
<td>54</td>
</tr>
<tr>
<td>4.3</td>
<td>Master and slave concept for Bluetooth link establishment</td>
<td>60</td>
</tr>
<tr>
<td>4.4</td>
<td>Electrodes positions for Lead I ECG reading on a patient</td>
<td>61</td>
</tr>
<tr>
<td>4.5</td>
<td>Set up of a temperature sensor device</td>
<td>62</td>
</tr>
<tr>
<td>4.6</td>
<td>SPO2 finger probe connection</td>
<td>63</td>
</tr>
<tr>
<td>4.7</td>
<td>m-Health processor module prototype</td>
<td>65</td>
</tr>
<tr>
<td>4.8</td>
<td>The structure of an IMSI number</td>
<td>70</td>
</tr>
<tr>
<td>5.1</td>
<td>Integrated software implementation of multi-channel GPRS-based m-Health system</td>
<td>75</td>
</tr>
<tr>
<td>5.2</td>
<td>Packet structure of m-Health processor module</td>
<td>77</td>
</tr>
<tr>
<td>5.3</td>
<td>Digitised data output format of m-Health processor module</td>
<td>78</td>
</tr>
<tr>
<td>5.4</td>
<td>Flowchart of the main software program</td>
<td>78</td>
</tr>
<tr>
<td>5.5</td>
<td>Private IP address blocks</td>
<td>82</td>
</tr>
<tr>
<td>5.6</td>
<td>Packet delivery flows in m-Health system</td>
<td>84</td>
</tr>
<tr>
<td>5.7</td>
<td>GPRS data flow process and relevant transformations at each protocol layer</td>
<td>85</td>
</tr>
<tr>
<td>5.8</td>
<td>Format of a GPRS RLC/MAC block for data transfer</td>
<td>87</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>5.9</td>
<td>The overview of J2ME building blocks</td>
<td>95</td>
</tr>
<tr>
<td>5.10</td>
<td>The Series 60 platform architecture</td>
<td>98</td>
</tr>
<tr>
<td>5.11</td>
<td>A MIDlet Life Cycle</td>
<td>100</td>
</tr>
<tr>
<td>5.12</td>
<td>Relationship between Java APIs for Bluetooth and J2ME</td>
<td>101</td>
</tr>
<tr>
<td></td>
<td>platform for Series 60 mobile phones</td>
<td></td>
</tr>
<tr>
<td>6.1</td>
<td>Client-server model of the m-Health system</td>
<td>110</td>
</tr>
<tr>
<td>6.2</td>
<td>Database connectivity concept</td>
<td>115</td>
</tr>
<tr>
<td>6.3</td>
<td>Diagram of the relationships between the telemedicine database tables</td>
<td>116</td>
</tr>
<tr>
<td>6.4</td>
<td>Architecture of the Telemedicine Application Server</td>
<td>120</td>
</tr>
<tr>
<td>6.5</td>
<td>Packet extraction process</td>
<td>121</td>
</tr>
<tr>
<td>6.6</td>
<td>Telemedicine session creation routine</td>
<td>122</td>
</tr>
<tr>
<td>6.7</td>
<td>Data updating process</td>
<td>123</td>
</tr>
<tr>
<td>6.8</td>
<td>Telemedicine Application Server Login interface</td>
<td>124</td>
</tr>
<tr>
<td>6.9</td>
<td>Active connection control interface</td>
<td>125</td>
</tr>
<tr>
<td>6.10</td>
<td>Session list interface</td>
<td>126</td>
</tr>
<tr>
<td>6.11</td>
<td>Patient registration interface</td>
<td>127</td>
</tr>
<tr>
<td>6.12</td>
<td>Use of a mobile phone as a modem</td>
<td>128</td>
</tr>
<tr>
<td>6.13</td>
<td>Patient user interface program on a laptop</td>
<td>129</td>
</tr>
<tr>
<td>6.14</td>
<td>Data flow process in the patient interface program</td>
<td>131</td>
</tr>
<tr>
<td>6.15</td>
<td>a) Mobile phone patient program- Main menu</td>
<td>133</td>
</tr>
<tr>
<td></td>
<td>b) Mobile phone patient program- Test mode</td>
<td></td>
</tr>
<tr>
<td>6.16</td>
<td>Overall procedures for mobile phone patient program</td>
<td>134</td>
</tr>
<tr>
<td>6.17</td>
<td>Packet payload format for mobile phone patient program</td>
<td>135</td>
</tr>
<tr>
<td>6.18</td>
<td>System architecture of the doctor browser</td>
<td>137</td>
</tr>
<tr>
<td></td>
<td>interface program</td>
<td></td>
</tr>
<tr>
<td>6.19</td>
<td>a) Doctor browser program Login interface</td>
<td>138</td>
</tr>
<tr>
<td></td>
<td>b) Patient list and sessions list interface</td>
<td>139</td>
</tr>
<tr>
<td></td>
<td>c) Doctor browser program data display interface</td>
<td>139</td>
</tr>
<tr>
<td>6.20</td>
<td>Flowchart of the doctor browser program on a laptop</td>
<td>140</td>
</tr>
<tr>
<td>6.21</td>
<td>Possible architectures for the mobile phone browser program</td>
<td>143</td>
</tr>
</tbody>
</table>
Figure 7.1 Set-up for transmitting ECG records from the MIT-BIH database using a patient interface program on a laptop

Figure 7.2 Patient interface program on the laptop for the MIT-BIH database

Figure 7.3 Set-up for transmitting real multi-channel medical data from a test patient using a mobile phone interface

Figure 7.4 Patient interface program on the mobile phone for the mobile tests

Figure 7.5 Mobile test with real multi-channel medical data from a test patient using the mobile phone interface

Figure 7.6 Set-up for transmitting real multi-channel medical data with a laptop as an intermediate platform

Figure 7.7 Test set-up was prepared and evaluated while driving in a car

Figure 7.8 Lead MLII ECG signals received and stored in the telemedicine database

Figure 7.9 Completion time for various data sizes under different mobile environments using a GPRS Class 8 mobile phone

Figure 7.10 Throughputs for various data sizes under different mobile environments using a GPRS Class 8 mobile phone

Figure 7.11 Completion time for various data sizes under different mobile environments using a GPRS Class 6 mobile phone

Figure 7.12 Throughputs for various data sizes under different mobile environments using a GPRS Class 6 mobile phone

Figure 7.13 The mobile phone interface display while transmitting medical signals during the mobile tests

Figure 7.14 Completion time for multi-channel medical data with both indoor and outdoor mobile tests environments

Figure 7.15 Throughputs for multi-channel medical data with both indoor and outdoor mobile tests environments

Figure 7.16 Display of the patient interface program while transmitting medical signals for the indoor environment
Figure 7.17  Completion time for the mobile tests using the laptop interface with the time of day variation

Figure 7.18  Throughputs for the mobile tests using the laptop interface with the time of day variation
# TABLE OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 2.1</td>
<td>EU funded projects under the Fifth Framework Programme</td>
<td>18</td>
</tr>
<tr>
<td>Table 3.1</td>
<td>Types of clinical information</td>
<td>26</td>
</tr>
<tr>
<td>Table 3.2</td>
<td>Several biomedical signal characteristics</td>
<td>27</td>
</tr>
<tr>
<td>Table 3.3</td>
<td>Effects on human body with a change in body temperature</td>
<td>32</td>
</tr>
<tr>
<td>Table 3.4</td>
<td>Short-range wireless technologies</td>
<td>33</td>
</tr>
<tr>
<td>Table 3.5</td>
<td>Bluetooth power classes</td>
<td>34</td>
</tr>
<tr>
<td>Table 3.6</td>
<td>SCO packets</td>
<td>36</td>
</tr>
<tr>
<td>Table 3.7</td>
<td>ACL packets</td>
<td>37</td>
</tr>
<tr>
<td>Table 3.8</td>
<td>Mobile technologies</td>
<td>40</td>
</tr>
<tr>
<td>Table 3.9</td>
<td>GPRS coding scheme</td>
<td>44</td>
</tr>
<tr>
<td>Table 3.10</td>
<td>GPRS multi-slot classes</td>
<td>45</td>
</tr>
<tr>
<td>Table 4.1</td>
<td>Input channel numbers and the type of medical signals</td>
<td>55</td>
</tr>
<tr>
<td>Table 4.2</td>
<td>Pin assignments of the Bluetooth module</td>
<td>59</td>
</tr>
<tr>
<td>Table 5.1</td>
<td>Description of the LED status in the Bluetooth module</td>
<td>80</td>
</tr>
<tr>
<td>Table 5.2</td>
<td>Size of overhead added at each protocol layer</td>
<td>86</td>
</tr>
<tr>
<td>Table 5.3</td>
<td>Size of GPRS RLC/MAC block for data transfer</td>
<td>87</td>
</tr>
<tr>
<td>Table 5.4</td>
<td>Minimal number of RLC/MAC blocks required for 1036 bytes of medical data per IP packet</td>
<td>88</td>
</tr>
<tr>
<td>Table 5.5</td>
<td>List of GPRS PDP types</td>
<td>91</td>
</tr>
<tr>
<td>Table 5.6</td>
<td>JSRs related to the m-Health system</td>
<td>97</td>
</tr>
<tr>
<td>Table 6.1</td>
<td>Terminology of relational database model</td>
<td>113</td>
</tr>
<tr>
<td>Table 6.2</td>
<td>Description of the Patient Information table</td>
<td>117</td>
</tr>
<tr>
<td>Table 6.3</td>
<td>Description of the Sessions table</td>
<td>117</td>
</tr>
<tr>
<td>Table 6.4</td>
<td>Description of the Signal Sets table</td>
<td>117</td>
</tr>
<tr>
<td>Table 6.5</td>
<td>Security and encryption of wireless technologies</td>
<td>144</td>
</tr>
<tr>
<td>Table 7.1</td>
<td>GPRS reliability classes</td>
<td>154</td>
</tr>
<tr>
<td>Table 7.2</td>
<td>GPRS delays classes</td>
<td>155</td>
</tr>
</tbody>
</table>
Table 7.3 Data size variations used in the mobile tests 158
Table 7.4 Mobile phone models used in the mobile tests 159
Table 7.5 List of parameters variation in each mobile environment 160
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3GPP</td>
<td>3rd Generation Partnership Project</td>
</tr>
<tr>
<td>ACL</td>
<td>Asynchronous Connection-Less</td>
</tr>
<tr>
<td>ADC</td>
<td>Analogue-to-Digital Converter</td>
</tr>
<tr>
<td>ADO</td>
<td>ActiveX Data Object</td>
</tr>
<tr>
<td>ADSL</td>
<td>Asymmetric Digital Subscriber Line</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>APN</td>
<td>Access Point Name</td>
</tr>
<tr>
<td>ARQ</td>
<td>Automatic Repeat Request</td>
</tr>
<tr>
<td>ASIC</td>
<td>Application Specific Integrated Circuit</td>
</tr>
<tr>
<td>ASP</td>
<td>Active Server Pages</td>
</tr>
<tr>
<td>ATM</td>
<td>Asynchronous Transfer Mode</td>
</tr>
<tr>
<td>BER</td>
<td>Bit Error Rate</td>
</tr>
<tr>
<td>BSC</td>
<td>Base Station Controller</td>
</tr>
<tr>
<td>BTS</td>
<td>Base Transceiver Station</td>
</tr>
<tr>
<td>CDC</td>
<td>Connected Device Configuration</td>
</tr>
<tr>
<td>CGI</td>
<td>Common Gateway Interface</td>
</tr>
<tr>
<td>CLDC</td>
<td>Connected Limited Device Configuration</td>
</tr>
<tr>
<td>CMRR</td>
<td>Common Mode Rejection Ratio</td>
</tr>
<tr>
<td>CRC</td>
<td>Cyclic Redundancy Check</td>
</tr>
<tr>
<td>CT</td>
<td>Computerised Tomography</td>
</tr>
<tr>
<td>DBMS</td>
<td>Database Management System</td>
</tr>
<tr>
<td>DTM</td>
<td>Dual Transfer Mode</td>
</tr>
<tr>
<td>ECG</td>
<td>Electrocardiogram</td>
</tr>
<tr>
<td>EDGE</td>
<td>Enhanced Data Rate for Global Evolution</td>
</tr>
<tr>
<td>EEG</td>
<td>Electroencephalogram</td>
</tr>
<tr>
<td>EMG</td>
<td>Electromyogram</td>
</tr>
<tr>
<td>ETSI</td>
<td>European Telecommunications Standards Institute</td>
</tr>
<tr>
<td>FEC</td>
<td>Forward Error Correction</td>
</tr>
<tr>
<td>FLIP</td>
<td>Flexible In-System Programmer</td>
</tr>
<tr>
<td>GFSK</td>
<td>Gaussian Frequency Shift Keying</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>GGSN</td>
<td>Gateway GPRS Support Node</td>
</tr>
<tr>
<td>GPRS</td>
<td>General Packet Radio Service</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>GSM</td>
<td>Global System for Mobile Communications</td>
</tr>
<tr>
<td>GTP</td>
<td>GPRS Tunnelling Protocol</td>
</tr>
<tr>
<td>HCI</td>
<td>Host Controller Interface</td>
</tr>
<tr>
<td>HLR</td>
<td>Home Location Register</td>
</tr>
<tr>
<td>HTML</td>
<td>HyperText Markup Language</td>
</tr>
<tr>
<td>IANA</td>
<td>Internet Assigned Numbers Authority</td>
</tr>
<tr>
<td>IETF</td>
<td>Internet Engineering Task Force</td>
</tr>
<tr>
<td>IMEI</td>
<td>International Mobile Equipment Identity</td>
</tr>
<tr>
<td>IMSI</td>
<td>International Mobile Subscriber Identity</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>ISDN</td>
<td>Integrated Services Digital Network</td>
</tr>
<tr>
<td>ISM</td>
<td>Industrial, Scientific and Medical</td>
</tr>
<tr>
<td>ISP</td>
<td>In-System Programming</td>
</tr>
<tr>
<td>J2EE</td>
<td>Java 2 Platform Enterprise Edition</td>
</tr>
<tr>
<td>J2ME</td>
<td>Java 2 Platform Micro Edition</td>
</tr>
<tr>
<td>J2RE</td>
<td>Java 2 RunTime Environment</td>
</tr>
<tr>
<td>J2SE</td>
<td>Java 2 Platform Standard Edition</td>
</tr>
<tr>
<td>JAR</td>
<td>Java Archive</td>
</tr>
<tr>
<td>JDBC</td>
<td>Java Database Connectivity</td>
</tr>
<tr>
<td>JSR</td>
<td>Java Specification Requests</td>
</tr>
<tr>
<td>kbps</td>
<td>kilobits per second</td>
</tr>
<tr>
<td>Kbytes</td>
<td>Kilobytes</td>
</tr>
<tr>
<td>L2CAP</td>
<td>Logical Link Control and Adaptation Protocol</td>
</tr>
<tr>
<td>LLC</td>
<td>Logical Link Control</td>
</tr>
<tr>
<td>LMDS</td>
<td>Local Multi-point Distribution Service</td>
</tr>
<tr>
<td>LMP</td>
<td>Link Manager Protocol</td>
</tr>
<tr>
<td>LSB</td>
<td>Least Significant Bit</td>
</tr>
<tr>
<td>MAN</td>
<td>Metropolitan Area Network</td>
</tr>
</tbody>
</table>
MCC Mobile Country Code
MIDP Mobile Information Device Profile
MMS Multimedia Messaging Service
MNC Mobile Network Code
MS Mobile Station
MSB Most Significant Bit
MSIN Mobile Subscriber Identification Number
NAT Network Address Translation
NIBP Non-Invasive Blood Pressure
ODBC Open Database Connectivity
SIM Subscriber Identity Module
SoC System-on-Chip
SPO2 Oxygen Saturation
PACCH Packet Access Control Channel
PACS Picture Archiving and Communications System
PAN Personal Area Network
PCB Printed Circuit Board
PCS Personal Communication System
PDA Personal Digital Assistant
PDN Packet Data Network
PDP Packet Data Protocol
PDTCH Packet Data Traffic Channel
PERL Practical Extraction and Report Language
PHP PHP: Hypertext Preprocessor
PPG Photoplethysmography
PSTN Public Switched Telephony Network
P-TMSI Packet Temporary Mobile Subscriber Identity
QoS Quality of Service
RAM Random Access Memory
RF Radio Frequency
RLC/MAC Radio Link Control/Medium Access Control
SCO Synchronous Connection-Oriented
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDK</td>
<td>Software Development Kit</td>
</tr>
<tr>
<td>SDP</td>
<td>Service Discovery Protocol</td>
</tr>
<tr>
<td>SGSN</td>
<td>Serving GPRS Support Node</td>
</tr>
<tr>
<td>SMS</td>
<td>Short Message Services</td>
</tr>
<tr>
<td>SNDCP</td>
<td>Subnetwork Dependent Convergence Protocol</td>
</tr>
<tr>
<td>SQL</td>
<td>Structured Query Language</td>
</tr>
<tr>
<td>SRAM</td>
<td>Static Random Access Memory</td>
</tr>
<tr>
<td>TCH</td>
<td>Traffic Channel</td>
</tr>
<tr>
<td>TCP/IP</td>
<td>Transmission Control Protocol/Internet Protocol</td>
</tr>
<tr>
<td>TDD</td>
<td>Time-Division Duplexing</td>
</tr>
<tr>
<td>TDMA</td>
<td>Time Division Multiple Access</td>
</tr>
<tr>
<td>UART</td>
<td>Universal Asynchronous Receiver Transmitter</td>
</tr>
<tr>
<td>UDP</td>
<td>User Datagram Protocol</td>
</tr>
<tr>
<td>UMTS</td>
<td>Universal Mobile Telephone Service</td>
</tr>
<tr>
<td>WAN</td>
<td>Wide Area Network</td>
</tr>
<tr>
<td>WAP</td>
<td>Wireless Application Protocol</td>
</tr>
<tr>
<td>WBAN</td>
<td>Wireless Body Area Network</td>
</tr>
<tr>
<td>WEP</td>
<td>Wired Equivalent Privacy</td>
</tr>
<tr>
<td>WLAN</td>
<td>Wireless Local Area Network</td>
</tr>
</tbody>
</table>
CHAPTER ONE:

INTRODUCTION

1.1 General Overview of m-Health Concept

The traditional way of providing telemedicine services has been to transmit biomedical signals from a patient to a hospital using "landline" communication technology, such as the Public Switched Telephony Network (PSTN) and the Integrated Services Digital Network (ISDN) [1.1-1.4]. The transmission of signals representing parameters such as the electrocardiogram (ECG), blood pressure, temperature and oxygen saturation at low bandwidths and x-ray images and ultrasound scans at high bandwidths has been achieved in radiology, cardiology, pathology and surgery [1.1-1.5]. The idea of providing medical information and health care services to a remote patient has made telemedicine emerge as a fast and rapidly expanding area of research. Telemedicine has considerable potential in enhancing the quality and accessibility of medical services, both in urban and rural areas and in developed and developing countries, because it allows patients to be monitored in a truly remote way, perhaps while on the move and from almost anywhere in the world.
The limitations of conventional telemedicine are self-evident, since applications are limited to communications between fixed locations, often with desktop applications. Therefore, the integration of mobile cellular networks with telemedicine applications is considered the next important step in creating more mobile and cost effective health care services. Collaboration between technologists and clinicians is beginning to demonstrate the practicality and effectiveness of this integration for health care provision [1.6]. The advancement of wireless communication technology has made a significant impact on telemedicine, varying from fixed platform services to wireless and mobile applications. In addition, parallel advances in pervasive, miniaturized and wearable systems have led to the m-Health concept, the 21st century evolution of traditional telemedicine. In general, m-Health can be defined as “mobile computing, medical sensor and communications technologies for healthcare” [1.7].

Although the common medical scenario of face-to-face consultations between a clinician and a patient will never be replaced, there are medical cases that can be managed more efficiently by using m-Health systems. Potential mobile applications include remote routine check-ups, emergency and rescue situations, and sports science physiological measurements. Medical services can now be delivered to any location within the coverage area of cellular networks, such as the Global System for Mobile communications (GSM). The option of monitoring patients effectively and reliably outside a hospital presents a great opportunity to lower the burden on national health care resources. A patient from a rural area could be given a routine check by mobile phone without having to commute to a hospital on a regular basis. Routine inspections and monitoring can be done while the patient is at home, travelling, at work or at leisure, thereby relieving resources for more demanding cases in hospitals. This can save time and money for both clinicians and patients, especially by obviating the need for patients to travel, often for short visits after long waits.
To give an idea of the numbers of patients who might be monitored, in the United Kingdom alone there are presently some 1.5 million people who suffer from angina and 500,000 who have some measure of heart failure. Heart disease is the main cause of death in the UK, accounting for over 232,000 deaths in 2003 [1.8].

In order to exploit the benefits discussed, there is a need for a new m-Health system that is capable of transmitting a patient’s biomedical signals to a hospital, and which provides flexibility and convenience to the patient. Integration of application-specific wireless sensors to the existing communication infrastructure is the focus of such a system. The system, which is compact and connected to wide area networks via General Packet Radio Service (GPRS) mobile phones, is equipped with Bluetooth interface for wireless sensors. This thesis presents the design and practical implementations of an integrated m-Health system with an interface processor for the transmission of multi-channel biomedical signals over a Bluetooth link to GPRS-based mobile cellular networks. This design allows uplink transmission of physiological signals from a patient directly to a hospital using a mobile phone on a nationally deployed commercial GPRS network.
1.2 Aim and Objectives of Research

The aim of this research is to develop an embedded Bluetooth m-Health system for transmissions of multi-channel biomedical signals via one or more mobile cellular networks. The objectives of the research are as follows:

- To design and implement a modular m-Health processor unit to sample, process and transmit biomedical signals
- To design a Bluetooth communication interface between the m-Health processor and a mobile phone
- To transmit sampled biomedical signals from a patient to a hospital using a cellular network and investigate the performance of the system under different mobile environments

1.3 Design Considerations

As greater patient mobility gradually becomes a trend in remote monitoring, the integration of wireless sensors with global connectivity seems to be the next step in providing telemedicine services.

The end-product of the research is a processor to sample, process and transmit up to four channels of biomedical signals via cellular networks. It allows the transmission of data for any or all of the four most common parameters used in medical monitoring, which are the electrocardiogram (ECG, from which the heart rate can be derived), blood pressure, temperature and oxygen saturation (SpO₂).

To achieve this aim requires an embedded Bluetooth interface between the processor and a mobile phone so that the signals can be transmitted transparently from a patient to a hospital, or to a clinician on the move. The
The processor unit has directly-wired inputs from sensors on the patient and a wireless connection to a mobile phone.

The rapid advancement in availability, performance and bandwidth of mobile network technology means the design of the processor unit has to be adaptable to such changes. The hardware is capable of multiplexing more medical signals in the processor unit so that more digital data can be handled with high-bandwidth Third Generation (3G) cellular networks. The flexibility of this design also allows the system to be used in non-medical applications, such as in environmental monitoring. The software is coded such that any future upgrades will require only minimal changes.

Another design consideration is the connection between the processor unit and the mobile phone. To provide flexibility of use for the patient, the processor unit is implemented with an embedded short-range, low-power Bluetooth communication module. With this connection, the alignment between the processor unit and the mobile telephone is not as critical as it was with the infrared link used in a previous design [1.9-1.11]. The connection between the patient's mobile phone and the hospital uses the uplink connection of commercially available GPRS network in the UK.

In addition, the Bluetooth-enabled processor unit is also capable of utilising an Internet Protocol (IP) connection to a hospital if the unit is interfaced with IP-connected devices via its Bluetooth connection, such as a laptop or a personal digital assistant (PDA). This feature is in line with the future 4G vision to have all IP-based heterogeneous networks with any system, anytime and anywhere [1.7].

The design of the system also includes a remotely accessible database server. The integration of the database server allows large amounts of medical data to be securely stored after the data acquisition and transmission. The availability of multi-channel biomedical information provides a better insight
into the patient's medical condition. The patient's medical data from different sessions could be accessed by clinicians from any location by a direct or wireless connection to the database server.

The practical implementation of the research helped to investigate the performance of the system, as part of future m-Health applications. The research is vital in demonstrating and evaluating the concept as well as assessing the effectiveness of using mobile networks for the transmission of multi-channel biomedical signals. It is envisaged that personalised next-generation medical services will include the integration of Intelligent sensors in Wireless Body Area Network (WBAN) with communication technology gateways such as Internet Protocol (IP) and Wide Area Network (WAN) [1.7]. With that, the research contributes to the analysis of fundamental concepts that lead to the emergence of future wearable, intelligent and personalised healthcare systems.

Numerous challenges in deployment of m-Health are expected, since the economical and social issues are prominent in the introduction of a new technology. The price imposed on patients and cost savings on healthcare providers will definitely be the main debate, as well as liability, ethical and privacy issues. However, the core and key contribution of the research is to materialise what is scientifically and technologically achievable, which has been the main focus of the project.

By utilising short-range and cellular network technologies, a more convenient, flexible and timely accessible m-Health system can be made possible. It was reported in GSM World news that in December 2004, there were around 1.7 billion cellular network subscribers in the world [1.12] and market analysts predict that there will be approximately 720 million Bluetooth devices in operation by the year 2008 [1.13]. By implementing a system with technologies that are strongly adopted worldwide, this m-Health system is poised to play a large part in the future of personalised medical services,
potentially made available to everyone from all walks of life. With the notion of "From 'Cell Phone' to 'Remote Control on Life'", it is interesting to see how wireless technologies will affect the way people live over the next few decades [1.14].

1.4 Chapter Synopsis

The research incorporates knowledge in both computer and communication systems for medical applications. Due to the diverse nature of the research, the thesis is organised to focus on design and implementation methodology. The Design Concepts are first introduced, followed by the System Hardware, System Software and Application, and Database Server implementation. The practical nature of the project places an emphasis on system performance under actual communication network and with real medical data.

1.4.1 Chapter One: Introduction

Chapter One presents a general introduction of the m-Health concept and establishes the design considerations of the project.

1.4.2 Chapter Two: Review of m-Health Systems

Chapter Two presents a comprehensive review of the research topic, from evolution to the future challenges of m-Health. The emphasis is placed on existing systems with satellite links, 'short-range' networks and links, and mobile cellular networks. The review also highlights current development in wireless medical sensor technologies and relates to the sensors adopted for this research.
1.4.3 Chapter Three: Mobile Bio-monitoring: Design Concepts

Chapter Three focuses on the concepts that led to the fundamental guidelines for the research and on the rationale for choosing the components of the system. Characteristics of biomedical signals, features and specifications of Bluetooth technology, and exploitation of GPRS networks are discussed.

1.4.4 Chapter Four: System Hardware Design

Chapter Four describes the overall system and how the system works. The hardware design of the processor module is discussed in detail. For a practical implementation of the system, a commercially available mobile phone has been used and necessary configuration on the phone is explained.

1.4.5 Chapter Five: System Software and Protocols

Chapter Five provides the software and protocols implementation of the system. The chapter discusses the overview of software functionality and highlights its modular features. The assembly program for the processor module, which includes the communication interface with the Bluetooth module, is presented in a diagrammatic form. The use of Java 2 Platform Micro Edition (J2ME) MIDlet suite for the patient interface program on the mobile phone that handles the GPRS connection protocols is also presented. The chapter also describes the GPRS IP packet format and protocols adopted for medical data transmission.
1.4.6 Chapter Six: Application and Database Server

Chapter Six explains the integration of an application and a database server into the overall m-Health system. The protocols and procedures adopted in designing both servers are presented in detail. A relational database design used is justified and the patient database entries are also explained. The server interfaces and the corresponding functions are described. The system architecture for the mobile phone browser program is highlighted. The security features implemented that provide protection for the clinical data integrity and the medical confidentiality of the patient are also explained.

1.4.7 Chapter Seven: Performance Analysis and Technical Results

Chapter Seven presents the practical performance of the system with a GPRS network under different mobile conditions. The details of the mobile test procedures carried out and the test parameters used for the analysis are described. Results and findings are summarised at the end of the chapter.

1.4.8 Chapter Eight: Conclusions and Future Research

Chapter Eight concludes all the work accomplished in the research and highlights the potential use of the system in other areas of research. Future recommendations discuss some possible research opportunities and collaborations to address different aspects of the system.

1.5 Summary

This introduction chapter briefly introduces m-Health and highlights the importance of the concept. The chapter also outlines the aim and objectives of the research, together with the design considerations in implementing the project. A synopsis of each chapter is also provided to give an insight into the content of the thesis.
1.6 References


CHAPTER TWO:

REVIEW OF m-HEALTH SYSTEMS

2.1 Wireless Technologies in m-Health

The concept of bio-telemetry has been explored extensively in the last few decades. Most of biomedical signals such as ECG, temperature and blood pressure have been successfully transmitted from human and animals. An important step in the early development of bio-telemetry was the recording and transmission of an ECG over a distance of 1.5km by Einthoven in 1903 via the public telephone system [2.1]. Radio-telemetry transmission later became the main technique for bio-telemetry since initiated by S. R. Winters in 1921. Developments in the early days were limited to a short-range single-channel radio transceiver. The use of bio-telemetry equipment in mobile clinical emergency systems was also envisaged but was rarely used with portable devices. Significant improvements in bio-telemetry then prompted the introduction of remote monitoring in telemedicine.

The capability of patient monitoring from rural health centres and isolated locations since then has grown rapidly as part of m-Health applications. Advancements of wireless communication technology have made
developments in m-Health systems more interesting in the past few years, thus contributing to the vital "medical connectivity" area of emerging m-Health applications. With m-Health, the classic bio-telemetry concept then evolved to m-diagnostic, monitoring and on the move applications between patients and healthcare providers [2.2, 2.3]. In general, m-Health systems can be categorised in terms of the technologies used, namely (i) satellite links, (ii) 'short-range' networks and links, and (iii) mobile cellular networks (GSM, GPRS and 3G). Many recent studies have addressed these technologies, and how their integration into m-Health applications helps towards an enhanced quality, greater performance and a more flexible system.

The use of satellite communication allows telemedicine applications to be facilitated in extreme and isolated locations. A telemedicine system with a satellite link can be made available in almost any part of the world, even in the most remote and hostile environments, such as on Mount Everest [2.4, 2.5]. A satellite link has been used in this ambitious and sophisticated telemedicine project to monitor the location, heart rate, skin temperature, core body temperature and activity level of climbers on Mount Everest during expeditions, and claimed to have been the first to achieve continuous real-time monitoring of vital signs on an ambulatory person in truly remote or hazardous conditions.

Satellite telephones have also been used to transmit echocardiograph images [2.6] as well as ECG tracing and blood pressure measurements during a high altitude mountaineering expedition to Mount Logan [2.7]. Another project involving a satellite link has been the development of a web-based Picture Archiving and Communications System (PACS) [2.8]. This experimental system was designed to allow doctors to search for and retrieve compressed medical images from remote servers. A web-based PACS would enable small hospitals in remote areas to access a central PACS easily and retrieve the relevant image data reliably.
Satellite technology has potential applications in various other situations, such as navigating ships and aircraft in flight [2.9], remote civilian and military operations [2.10], space operations [2.11, 2.12] and in emergency scenes or disaster situations [2.13]. Unfortunately, such a system requires bulky and expensive equipment, as well as dedicated satellite links and operation by skilled personnel.

Studies of mobile cellular networks include the performance, using software simulation, of telemedicine links based on the IS-54 and GSM protocols [2.29, 2.30]. Different data traffic conditions were simulated, ranging from ideal to distorted, and the results showed successful multi-channel transmission of medical data with low bit error rates, depending on the mobile environment tested. Modelling of a photoplethysmography (PPG) mobile telemedicine system with reduced motion artefacts PPG sensors also includes the feasibility of mobile systems for mobile patient monitoring [2.31]. The main outcome highlighted by these simulations was the need to investigate the effect of patient mobility on the reception quality of the received signals.

Further simulations have been carried out on wireless telemedicine networks using 3G specifications. These simulations included a next-generation mobile telemedicine test-bed, which showed potential improvements over present network specifications in terms of the quality of service [2.32]. The performance of a 3G network has also been simulated for multimedia applications of a mobile tele-echography robotic system, using a channel simulator with the Universal Mobile Telephone Service (UMTS) Proxy [2.33].

Some applications of m-Health systems using cellular networks have concentrated on the transmission of medical data using down-link channels [2.34, 2.35]. Typically, medical information is transmitted from a hospital to an on-call clinician, normally for a second opinion or for specialist assessment. Direct transmission of electrocardiograms to a mobile phone for management of a patient with acute myocardial infarction has been achieved using a Nokia
9000 communicator to transmit a high-resolution ECG to a cardiologist [2.34]. While the system appeared to be valuable in large city hospitals at night and weekends, it may be even more useful in smaller suburban or district hospitals, particularly in rural and remote areas, where no doctor is immediately available.

Another emerging application in using down-link channels is the use of wireless application protocol (WAP) devices to allow users to browse or to view a static display of medical information while on the move [2.36-2.38]. The WAP application utilizes access to a WAP content server and downloads the medical information for patient monitoring in a store-and-forward mode. However, a WAP-based application requires a client-specific browsing capability and it is difficult to produce a generic and device-independent application [2.37]. The use of short-message-services (SMS) on the other hand is limited to sending alarm messages to alert either a patient or a clinician of a possible medical situation [2.39, 2.40].

Further applications have involved the transmission of biomedical signals during emergency situations, such as from a moving ambulance, using the GSM network [2.41-2.45]. These are examples of using the up-link channel of a cellular network. The principle is to transmit critical biomedical signals to a hospital trauma centre for early diagnosis and evaluation before the patient arrives. The disadvantage of the procedure is that the associated equipment is complex, comprising specialised medical equipment, a laptop computer and perhaps a digital camera, and multiple cellular channels are needed to transmit the biomedical signals and video images.

There is probably a greater need to send signals from a patient in a non-emergency situation, such as during post-operative care in which patients transmit their biomedical signals to a hospital for monitoring applications. A portable ECG monitor has been used to record ECG signals, which are later transmitted as a high-pitched whine via a mobile phone to a monitoring
centre [2.46]. Another project has involved the transfer of still images, taken with a digital camera in a patient’s home, to a hospital server via an e-mail attachment [2.47]. This technique may be useful for an initial observation but the patient’s original biomedical signals are not available for observation. A radiology consultation has also been performed with a GSM network for sending computerized tomography (CT) images from a portable computer to a hospital image network [2.48].

In terms of mobile technology, the adoption of GPRS networks in m-Health systems is relatively new [2.35, 2.49] but it has the major advantage of enabling transmission of both data and speech. The use of GPRS networks is also envisaged in future applications with wearable medical sensors [2.50, 2.51].

Previous research on the transmission of multi-parameter biomedical signals is bounded by the use of a PC-based system [2.52]. The system is constrained by the use medical devices with cable outputs to data acquisition boards and two laptop PCs equipped with communication modems. This limitation motivates further research focusing on a portable system that utilises the mobile uplink channel with the capability of sending multi-channel biomedical signals. Since most work done in the use of mobile networks has centred on simulated environments, there is a need for a practical system that can be examined and evaluated with commercial and widely available GPRS networks.

Several research projects have been concerned with the implementation of telemedicine systems using a wireless local area network (WLAN) and a short-range radio frequency (RF) transceiver [2.14-2.20]. The main focus was to allow wireless transmission of vital signs from the patient’s medical equipment to a central system or to a clinician’s PDA [2.18]. The MedL-AN system has demonstrated the suitability of having a wireless environment within a hospital Accident and Emergency ward [2.19]. A combination of both WLAN
and an RF link was also evaluated in an emergency room situation, which allows greater manoeuvrability within the room [2.20]. All of these systems were intended for implementation in a hospital and therefore have limited range capability. This type of system cannot normally be used at home or when on the move, as is possible with an implementation based on a wide area cellular network.

The idea of using Bluetooth technology in m-Health applications has also been explored in several projects, including the simultaneous deployment of Bluetooth and WLAN technology in a hospital environment, with data rates up to about 100 kbps [2.21]. The feasibility of implementing Bluetooth technology with a digital ECG recorder [2.22], a stethoscope [2.23], wireless ECG sensors [2.24] and miniaturised non-invasive bio-sensors [2.25] has also been studied. Bluetooth-enabled physical sensors have also been used in rescue work to monitor the physical conditions of both rescuers and injured victims [2.26]. The work done so far is limited to a short-range personal area network (PAN) and definitely creates a need for a system that is capable of integrating wide area network connectivity such as the GPRS network. In view of its growing availability and versatile application potential, Bluetooth is one of the technologies adopted here.

The use of ZigBee standard protocols in m-Health systems is rather new. The ZigBee protocols are aimed for a very low power consumption and built on top of the IEEE 802.15.4 standard [2.2, 2.27]. The data rate for ZigBee protocols is in the range of 20 to 250 kbps, with the transmission range of between 5 to 500m. There is only one medical sensor implementation with ZigBee protocols [2.28] known to date.

In addition, research activities in the area of m-health systems have also been supported by the European Union (EU). The research projects are funded under the consecutive four-year programmes, known as Framework
Programmes. Some of the EU-funded projects in m-Health systems under the Fifth Framework Programme (1998-2002) are presented in Table 2.1 [2.3].

<table>
<thead>
<tr>
<th>Projects</th>
<th>Telemedicine Area</th>
<th>Data Transmitted</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOBIDEV – Mobile devices for healthcare applications</td>
<td>Hospital</td>
<td>Electronic Patient Record (EPR)</td>
</tr>
<tr>
<td>C-Care – Continuous care</td>
<td>Hospital</td>
<td>EPR</td>
</tr>
<tr>
<td>WARD-IN-HAND – Mobile workflow support and information distribution in hospitals via handheld PCs</td>
<td>Hospital</td>
<td>EPR</td>
</tr>
<tr>
<td>HEALTHMATE – Personal intelligent health mobile systems for Telecare</td>
<td>Emergency Rural areas</td>
<td>EPR</td>
</tr>
<tr>
<td>CHS – Distance information technologies for home care</td>
<td>Remote monitoring</td>
<td>Diabetes ECG</td>
</tr>
<tr>
<td>EPI-MEDICS – Enhanced personal, intelligent and mobile system for early detection and interpretation of cardiological syndromes</td>
<td>Remote monitoring</td>
<td>ECG</td>
</tr>
</tbody>
</table>

Table 2.1 EU funded projects under the Fifth Framework Programme

2.2 Wireless Medical Sensor Technologies

Commercially available non-invasive medical sensors now vary from piezoelectrical materials, infrared sensors to opto-electronic sensors. As technology advances in short-range wireless and biotechnology, the adoption of wireless medical sensors integrated with a networking capability has been gradually introduced into mobile healthcare applications. The use of intelligent sensors with wireless connectivity can now produce a Wireless Body Area Network (WBAN) to acquire physiological signals from various body parts [2.53]. The short-range wireless protocols implemented ranging from RF transceivers to
Bluetooth and the newly introduced ZigBee standard. In view of its growing role in wireless sensors application, the Bluetooth protocol has been adopted in this research. The concept of a wireless sensor network not only has been investigated in health applications, but also in smart homes, for security, and for industrial automation, as well as the automotive industry [2.54].

As micro-and nano-technologies continue to evolve, miniaturized biomedical sensors can also be embodied with wearable carriers. The wearable carriers can possibly be a finger ring [2.55], a smart shirt [2.56], a bio-cloth T-shirt [2.50] or fibre and yarn knitted fabric [2.51]. All embodied sensors are integrated with a communication gateway for mobile wide area connection.

Telecommunication capability is also highlighted as one of the major issues in current and future challenges of smart wearable health systems [2.57]. The link between sensors and the link between a smart wearable system and a healthcare provider involves short and long range wireless and mobile communications. The portable system developed in this research should therefore contribute to a basis of current and future development in wearable systems. With the increase of an ageing population and chronic diseases, society becomes more health conscious and patients become "health consumers" looking for better health management [2.57]. People's perception is shifting towards patient-centred, rather than the classical, hospital-centred health services. It is expected that m-Health systems integrated with wireless sensors have great potential in transforming the current way of healthcare delivery and in leading towards the provision of personalised next-generation medical applications.
2.3 Summary

A review of the early concept of bio-telemetry leads to the emergence of applications in m-Health. The categorisation of existing m-Health systems is based on the technologies used, namely satellite links, 'short-range' networks and links, and mobile cellular networks. The satellite-based telemedicine systems can be made available to every part of the world, especially during disaster and emergency situations. The fact that satellite systems require bulky and expensive equipment, dedicated satellite links and operation by skilled personnel limits their wide acceptance. The use of 'short-range' networks so far has focused on improving wireless connectivity in a hospital and therefore cannot normally be used at home or when on the move. On the other hand, most of the research in the use of mobile cellular networks has centred on simulated environments and the practical implementations have concentrated on the transmission of medical data using mobile down-link channels.

Current development in medical sensor technologies has started to integrate wireless medical sensors with wide area communication technology gateways. The integrated m-Health system presented in this thesis highlights a unique system with an embedded Bluetooth interface capable of transmitting multi-channel biomedical data. The display and the uplink transmission of the data is possible with the use of a mobile phone, equipped with the J2ME MIDlet interface. The modular approach of the system design allows the system to be upgradeable and interoperable with other mobile networks. The final prototype contributes to the analysis and practical implementations of design concepts that lead to the novel idea of future wearable, intelligent and personalised healthcare.
2.4 References


3.1 Physiological Signals for Bio-monitoring

There is some variation in the amount and type of clinical data that can be transmitted in m-Health applications, as shown in Table 3.1 [3.1]. The types of signals adopted in the research are physiological signals applicable for monitoring and routine check-ups applications. Typical signals monitored are the electrocardiogram (ECG), temperature, blood pressure and oxygen saturation (SPO2), of which the ECG presents the biggest challenge because it has the most complex form. The characteristics of various biomedical signals are shown in Table 3.2 [3.2].

<table>
<thead>
<tr>
<th>Information Source</th>
<th>Type</th>
<th>Typical File Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECG Recording</td>
<td>Data</td>
<td>100 Kbytes</td>
</tr>
<tr>
<td>Electronic Stethoscope</td>
<td>Audio</td>
<td>100 Kbytes</td>
</tr>
<tr>
<td>X-Ray</td>
<td>Still Image</td>
<td>1 Mbytes</td>
</tr>
<tr>
<td>30s of Ultrasound Image</td>
<td>Moving Image</td>
<td>10 Mbytes</td>
</tr>
</tbody>
</table>

Table 3.1 Types of clinical information
CHAPTER THREE

<table>
<thead>
<tr>
<th>Signal</th>
<th>Frequency range</th>
<th>Signal Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electrocardiogram (ECG)</td>
<td>0.05 – 100 Hz</td>
<td>10 µV – 5 mV</td>
</tr>
<tr>
<td>Electroencephalogram (EEG)</td>
<td>0.5 – 60 Hz</td>
<td>15 – 100 mV</td>
</tr>
<tr>
<td>Heart Rate</td>
<td>45 – 200 beats/min</td>
<td>N/A</td>
</tr>
<tr>
<td>Breathing Rate</td>
<td>12 – 40 breaths/min</td>
<td>N/A</td>
</tr>
</tbody>
</table>
| Blood Pressure          | dc - 60 Hz      | 40 – 300 mm Hg (Arterial)  
                        |                 | 0 – 15 mm Hg (Venous)   |

Table 3.2 Several biomedical signal characteristics

3.1.1 Electrocardiogram (ECG) Signals

The ECG is one of the bioelectric signals produced by electrical changes associated with contraction of muscle and neural cells of the human body [3.3-3.5]. The electrical changes can be measured by placing three-lead or 12-lead ECG electrodes on the skin surface of the chest or limbs, depending on whether it is for monitoring or diagnostic cardiology purposes. Different electrode arrangements change the shape and intensity of the measured signal. Each lead gives unique information based on the lead’s orientation relative to the axis of the heart.

Figure 3.1 illustrates a complete cardiac cycle of an ideal ECG signal. The first ECG wave in a cardiac cycle is a P-wave corresponding to the Atrial Depolarisation that causes contraction of the atria. Then, the Atrial Repolarisation and the Ventricular Depolarisation cause ventricles to contract, hence producing a QRS complex. The ECG T-wave is caused by Ventricular Repolarisation, which is the return of the ventricles to their resting electrical state.
A three-lead ECG is derived from limb electrodes that form a triangle, known as Einthoven’s triangle [3.6]. The triangle is considered as equilateral for measurement purposes. Figure 3.2 illustrates the Einthoven’s Triangle.
The potential difference between the limbs can be viewed as a voltage source. Lead I looks at the heart in the horizontal axis from left arm to right arm and is the origin of the electrical axis of the heart. Lead II looks at the heart from left leg to right arm and Lead III looks at the heart from left leg to left arm.

ECG signals are commonly monitored in cardiology applications, using fixed recorders for intensive care patients and ambulatory monitors for mobile patients. In 'smart' hospitals and homes, digitized signals from a mobile patient are transmitted to the nearest wall-mounted receiver via a short-range radio link, then over a LAN to a central monitoring computer.

With the view of simplicity and adequacy for physicians to monitor the most important cardiac functions, heart rate, rhythm and in certain cases inadequate blood flow to heart muscle [3.4], the three-lead ECG will initially be implemented in our mobile telemedicine processor module. This type of monitoring is simple to apply and the gathering of useful ECG data is also very tolerant of misplaced leads. Keeping it simple is essential and a three-lead ECG is very appropriate in a remote situation, considering that the telemedicine processor module will be used by non-medically trained people.

The bandwidth requirement depends on the type of applications of the ECG signals [3.2]. The bandwidth of standard clinical 12-lead ECG applications is in the range of 0.05 to 100Hz. The monitoring applications use bandwidths in the range of 0.5 to 50Hz and heart rate meters (cardio tachometers) use a band-pass filter centred at 17Hz. The amplitude of an ECG signal is approximately 10 μV - 5 mV.

ECG recordings present different kinds of interference [3.7-3.9]. The main considerations are;
1. 50Hz electrical (power) line interference;
2. Baseline wander - low frequency noise interference caused by motion of the electrodes;
3. Electromyogram (EMG) artefacts - high frequency electric signals caused by muscle motion.

Among them, the EMG artifacts are the most difficult to be removed due to their random character and they share the same frequency band as the desired ECG signals obtained from the same pair of electrodes. Several methods have been proposed in order to remove artifacts from ECG signals to enhance the precision of ECG analysis [3.7-3.9]. In the proposed system, introducing a biomedical signal-conditioning module reduces these noise interference and motion artifacts. The module will consist of a gain amplifier, a low-pass and a high-pass filter. As for the 50Hz electrical (power) line interference, there are a few methods of reducing or eliminating the interference signal, such as using a notch filter, using a high common mode rejection ratio (CMRR) amplifier, applying an adaptive filter algorithm in the software program or, in a later design stage, using a battery as the power supply source [3.2, 3.9].

As mentioned earlier, ECG signals are used in monitoring and diagnostic cardiology applications such as ambulatory monitors, ECG recorders in intensive care units and telemetry applications. These applications are involved in a range of sampling rates and quantisation levels in digitising the ECG signals. For example, a high sampling rate of 1kHz and 12-bits precision is used in a real-time 12-lead ECG automatic analysis system for morphological diagnosis of a cardiac patient [3.5]. ECG signals are also oversampled to 1024Hz to improve the precision of the QT time interval measurement [3.10]. A sampling rate of 250Hz with 16-bit resolution has been used in ECG characterisation processing [3.12]. In ambulatory applications, however, a minimum sampling frequency for a fair
representation of ECG signals is 128Hz [3.11] and ambulatory ECG monitors often require a relatively low sampling frequency of 128Hz [3.10].

3.1.2 Blood Pressure

As the heart serves as a four-chambered pump for the circulatory system, the pumping function of the heart cycle produces diastole and systole phases. The diastole phase is the resting or the filling phase and the systole phase is during contracting or pumping of a heart cycle [3.6]. At each cardiac contraction, the aorta will pump the blood and the energy is transformed into pressure. The blood pressure measured during this contraction is called systolic and corresponds to the maximum pressure. At filling phase of the heart, the valves of the aorta will close themselves to prevent the blood to re-enter the heart. The residual pressure in the vessels is then called diastolic, which corresponds to the minimum pressure. Non-Invasive Blood Pressure (NIBP) is one of the most common ways of monitoring a patient's blood pressure by using an electronic monitor with a pressure sensor. The current unit of blood pressure is the millimetre of mercury (mmHg). The average blood pressure reading for a healthy person is in the range of 110 – 130 mmHg for Systolic pressure and between 75 – 85 mmHg for Diastolic pressure [3.13-3.15]. Currently, the general definition of high blood pressure (or arterial hypertension) value is 140 mmHg for the systolic blood pressure and 90 mmHg for the diastolic blood pressure.

3.1.3 Body Temperature

The core body temperature is used to provide information on the general health of a patient. The normal core body temperature is between 36.6 – 37.3 degrees Celsius. The most common illness indicated by core body temperature is a fever, with a core body temperature above 37.3 degrees
Celsius. The high core body temperature during fever is due to body reaction to fight infection or disease. Possible effects on human body due to significant changes in core body temperature are presented in Table 3.3 [3.16, 3.17].

<table>
<thead>
<tr>
<th>Temperature °C</th>
<th>Symptoms</th>
</tr>
</thead>
<tbody>
<tr>
<td>27</td>
<td>Sudden cardiac death in minutes</td>
</tr>
<tr>
<td>29</td>
<td>Loss of consciousness</td>
</tr>
<tr>
<td>30</td>
<td>No response to pain</td>
</tr>
<tr>
<td>33</td>
<td>Hallucinations, delusions – 50% lethality</td>
</tr>
<tr>
<td>37</td>
<td>Normal</td>
</tr>
<tr>
<td>42</td>
<td>Hyperthermia: Life-threatening damage, nervous system and/or cardiac failure</td>
</tr>
</tbody>
</table>

Table 3.3 Effects on human body with a change in body temperature

3.1.4 Oxygen Saturation (Spo2)

Pulse oximetry is a non-invasive measurement technique that enables measurement of blood oxygen saturation and heart rate. The intensity of light transmitted (or reflected) across the finger at two distinct wavelengths, in the red and infrared regions of spectrum is related to the oxygen saturation [3.18]. Dual wavelength illumination of arterial blood results in an absorption contrast, based on the proportion of haemoglobin that is chemically combined with oxygen. Oxygen saturation indicates the percentage of haemoglobin saturated with oxygen, which low saturations are given by the 10%-70% range. Oxygen saturation values obtained from pulse oximetry (SpO2) are one part of a complete assessment of the patient’s oxygenation status. Normal oxygen saturation values for a healthy individual are in the region of 97% to 99% [3.19]. The low saturation range is normally useful in foetus monitoring during labour and delivery.
3.2 Short-range Wireless Communication for Bio-monitoring

Bluetooth was invented in 1994 by L.M. Ericsson of Sweden and was named after Harald Blaatand "Bluetooth" II, the King of Denmark 940-981 A.D. Bluetooth is a universal, short-range, low-power radio protocol operating in the 2.4GHz unlicensed Industrial, Scientific and Medical (ISM) frequency band. Its maximum data rate and potential range is shown in Table 3.4, along with data for other short-range wireless technologies [3.20]. Some of the key benefits of short-range wireless technologies in general are also presented in Figure 3.3 [3.20]. The possibility of embedding Bluetooth transceiver modules into various devices [3.21] has inspired the use of this technology in the research. Bluetooth enables portable systems to have wireless connections for both data and voice applications with point-to-point or point-to-multipoint connections. Technical specifications of the protocol and its architecture are available in further details elsewhere [3.22].

<table>
<thead>
<tr>
<th>Technology</th>
<th>Data Rate</th>
<th>Frequency</th>
<th>Max. Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>IrDA (Infrared Data Association)</td>
<td>4 Mbps</td>
<td>IR Spectrum</td>
<td>2m</td>
</tr>
<tr>
<td>WLAN (Wireless Local Area Network)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IEEE 802.11a</td>
<td>54 Mbps</td>
<td>5 GHz</td>
<td>&lt;600m</td>
</tr>
<tr>
<td>IEEE 802.11b</td>
<td>11 Mbps</td>
<td>2.4 GHz</td>
<td></td>
</tr>
<tr>
<td>IEEE 802.11g</td>
<td>54 Mbps</td>
<td>2.4 GHz</td>
<td></td>
</tr>
<tr>
<td>Bluetooth IEEE 802.15</td>
<td>723 kbps</td>
<td>2.4 GHz</td>
<td>10-100m</td>
</tr>
<tr>
<td>(Personal Area Network)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ZigBee (IEEE 802.15.4 standard)</td>
<td>250 kbps</td>
<td>2.4 GHz</td>
<td></td>
</tr>
<tr>
<td></td>
<td>40 kbps</td>
<td>902-928 MHz</td>
<td>5-500m</td>
</tr>
<tr>
<td></td>
<td>20 kbps</td>
<td>868-870 MHz</td>
<td></td>
</tr>
<tr>
<td>DECT (Digital Enhanced Cordless Telecommunication)</td>
<td>736 kbps</td>
<td>1.88 GHz</td>
<td>300-2500m</td>
</tr>
<tr>
<td>HomeRF (Home Radio Frequency)</td>
<td>1 Mbps</td>
<td>2.4 GHz</td>
<td>&lt;40m</td>
</tr>
</tbody>
</table>

Table 3.4 Short-range wireless technologies
3.2.1 Features and Specifications

The Bluetooth system operates under the Master-Slave concept that forms a Piconet (two or more units) or a Scatternet (multiple piconets) operation. A Bluetooth system consists of a 2.4GHz radio unit, a baseband unit, a link manager and a host interface unit. The modulation technique is Gaussian Frequency Shift Keying (GFSK) at a rate of 1Msymbol/s and transmitted using one of the 79 channels with 1MHz channel spacing in the 2.402 to 2.480GHz band.

Bluetooth utilises a spread spectrum frequency hopping connection with a rate of 1600 hops per second. A Bluetooth radio transceiver is categorised into three power classes, as shown in Table 3.5.

<table>
<thead>
<tr>
<th>Power Class</th>
<th>Maximum Output Power</th>
<th>Typical Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100 mW (20 dBm)</td>
<td>100 m</td>
</tr>
<tr>
<td>2</td>
<td>2.5 mW (4 dBm)</td>
<td>20 m</td>
</tr>
<tr>
<td>3</td>
<td>1 mW (0 dBm)</td>
<td>10 m</td>
</tr>
</tbody>
</table>

Table 3.5 Power classes
A Bluetooth channel uses Time-Division Duplexing (TDD) for full duplex transmission, where the master and slave transmit alternately. Each channel is divided into 625µs time slots. The master starts its transmissions in even-numbered time slots while the slaves start to transmit in odd-numbered time slots. On the channel, information is exchanged through packets and each packet is transmitted at a different hopping frequency. A packet normally uses one time slot, but this can be extended to multi-slots packet.

Bluetooth link connection can be categorised into two main types, namely Synchronous Connection-Oriented (SCO) link and Asynchronous Connection-Less (ACL) link. The SCO link is a symmetric, point-to-point connection between the master and a specific slave with dedicated timeslot reservations. The SCO link is typically used to carry isochronous information such as voice transmission. The ACL link is a point-to-multipoint link between the master and all active slaves in a piconet. The ACL link has no prescribed time slot allocations and is intended to support data applications.

Information exchange in Bluetooth links are done via packets. Bluetooth packets are divided into three main portions, the access code, the header and the payload, as shown in Figure 3.4. The access code is used for packet identification and synchronisation in a piconet and normally consists of 72 bits. All packets in a piconet share the same access code. The header part contains link control information such as device address, type of packet, flow control, acknowledgement status, sequence number and is protected with a

<table>
<thead>
<tr>
<th>LSB 72</th>
<th>54</th>
<th>0-2745</th>
<th>MSB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access Code</td>
<td>Header</td>
<td>Payload</td>
<td></td>
</tr>
</tbody>
</table>

Figure 3.4 Bluetooth packet format
1/3-rate Forward Error Correction (FEC). The size of the header is fixed at 54 bits. The payload of Bluetooth packets contains user information and occupies a single or multiple time slots. The Least Significant Bit (LSB) shows the start of a packet and the Most Significant Bit (MSB) indicates the end of a Bluetooth packet.

Bluetooth packets are related to the types of physical links that they are connected to, either the SCO packets or ACL packets, as shown in Table 3.6 and Table 3.7, respectively. For SCO packets, HV stands for High-quality Voice and DV for combined data-voice. All SCO packets do not have a Cyclic Redundancy Check (CRC) except for the data field in a DV packet. ACL packets are distinguished by the amount of FEC provided in the payloads, which are protected with a CRC for error detection. High-rate packets are not protected by FEC, while medium rate packets are protected. The DH packet stands for Data-High rate and DM stands for Data-Medium rate. The subsequent number followed indicates the maximum number of time slots the packets may occupy.

<table>
<thead>
<tr>
<th>Type</th>
<th>FEC</th>
<th>Symmetric Max. Rate (kbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HV1</td>
<td>1/3</td>
<td>64.0</td>
</tr>
<tr>
<td>HV2</td>
<td>2/3</td>
<td>64.0</td>
</tr>
<tr>
<td>HV3</td>
<td>No</td>
<td>64.0</td>
</tr>
<tr>
<td>DV</td>
<td>2/3 for Data</td>
<td>64.0 + 57.6 Data</td>
</tr>
</tbody>
</table>

Table 3.6  SCO packets
<table>
<thead>
<tr>
<th>Type</th>
<th>FEC</th>
<th>Symmetric Max. Rate (kbps)</th>
<th>Asymmetric Rate (kbps)</th>
<th>Max. Rate (kbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DM1</td>
<td>2/3</td>
<td>108.8</td>
<td>108.8</td>
<td>108.8</td>
</tr>
<tr>
<td>DH1</td>
<td>No</td>
<td>172.8</td>
<td>172.8</td>
<td>172.8</td>
</tr>
<tr>
<td>DM3</td>
<td>2/3</td>
<td>258.1</td>
<td>387.2</td>
<td>54.4</td>
</tr>
<tr>
<td>DH3</td>
<td>No</td>
<td>390.4</td>
<td>585.6</td>
<td>86.4</td>
</tr>
<tr>
<td>DM5</td>
<td>2/3</td>
<td>286.7</td>
<td>477.8</td>
<td>36.3</td>
</tr>
<tr>
<td>DH5</td>
<td>No</td>
<td>433.9</td>
<td>723.2</td>
<td>57.6</td>
</tr>
<tr>
<td>AUX1</td>
<td>No</td>
<td>185.6</td>
<td>185.6</td>
<td>185.6</td>
</tr>
</tbody>
</table>

Table 3.7 ACL packets

3.2.2 Protocols Architecture

The basic Bluetooth protocol stack consists of Radio, Baseband, Link Manager Protocol (LMP), Host Controller Interface (HCI), Logical Link Control and Adaptation Protocol (L2CAP), Service Discovery Protocol (SDP), RFCOMM Protocol and Applications layer, as shown in Figure 3.5.

![Bluetooth protocols stack](image-url)
The Radio layer is the physical layer responsible for all spread spectrum frequency hopping communication and the GFSK modulation scheme. The Baseband layer is responsible for controlling the radio layer, providing hopping sequences, lower level encryption, packet handling, SCO and ACL link establishment. The LMP is involved in piconet management, link configuration and security functions.

The L2CAP is the protocol with which most applications would interact and provides multiplexing, packet segmentation and reassembly as well as quality of service checking. The SDP provides a way for the Application Layer to discover the availability and characteristics of Bluetooth services. The RFCOMM protocol is an emulation of serial ports over the L2CAP based on the European Telecommunications Standards Institute (ETSI) standard.

The implementation of a Bluetooth system is possible with two different configurations, namely host-based implementation and host-less implementation [3.22, 3.23].

The host-based implementation is where the Bluetooth module implements lower layer protocols – Radio, Baseband and LMP, and the upper layers reside in a software in the Host, usually a personal computer or a laptop, as shown in Figure 3.6.
The host-less implementation is where all the layers are implemented in the Bluetooth module, and since L2CAP is in the module itself, the HCI layer is no longer required, as shown in Figure 3.7. The host-less implementation is the one adopted in the system, in which the Bluetooth protocols are fully embedded into the Bluetooth module. Further details on the integration of the module into the system hardware are explained in section 4.2.2.

Figure 3.6 Bluetooth host-based implementation

Figure 3.7 Bluetooth host-less implementation
3.3 Choice of Mobile Cellular Networks

With the evolution of cellular networks from the second generation (2G), such as GSM, to General Packet Radio Service (GPRS) and Enhanced Data Rate for Global Evolution (EDGE), then to the Third Generation (3G), more services can be designed and modelled for next-generation mobile telemedicine applications. These advances in mobile technology generally reflect the increased availability, performance and communication bandwidth of mobile networks. The maximum theoretical data rates are presented in Table 3.8.

<table>
<thead>
<tr>
<th>Technology</th>
<th>Maximum Theoretical Data Rates</th>
<th>Frequency Spectrum</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSM</td>
<td>9.6 kbps</td>
<td>900/1800/1900 MHz</td>
</tr>
<tr>
<td>GPRS</td>
<td>171 kbps</td>
<td>900/1800/1900 MHz</td>
</tr>
<tr>
<td>EDGE</td>
<td>384 kbps</td>
<td>900/1800/1900 MHz</td>
</tr>
<tr>
<td>3G / UMTS</td>
<td>2 Mbps</td>
<td>1885 MHz to 2200 MHz</td>
</tr>
</tbody>
</table>

Table 3.8 Mobile technologies

3.3.1 Introduction to General Packet Radio Service (GPRS)

The second generation GSM networks' data communication is supported by circuit switched data bearer services with limited data rates of up to 9.6 kbps. The GPRS network represents an enhancement of the existing GSM network, by introducing packet data services with a packet-switched transmission within GSM radio architecture, for "always on" mobility. The GPRS network dynamically assigns timeslots on GSM radio channels to transmit packets of data with a maximum data of 171.2 kbps. The packet data services are possible with the introduction of new network elements to the existing GSM infrastructure, which will be explained in the next section. Although the service provision is continuous, subscribers are charged only when data transmission takes place.
The performance of data services in mobile cellular networks is normally associated with long connection set-up times, low available bandwidth, and an inefficient use of the radio resources. The standardisation of GPRS therefore focused strongly on overcoming these drawbacks and provides a platform to support existing packet-oriented protocols like X.25 and IP [3.24]. It has been reported that the majority of European operators have introduced the GPRS network [3.25] the situation today is that GPRS is widely available.

3.3.2 GPRS Network Architecture

The GPRS network architecture is shown in Figure 3.8. Two new elements, Serving GPRS Support Node (SGSN) and Gateway GPRS Support Node (GGSN) are introduced to the network as compared to the GSM architecture [3.26]. The SGSN is responsible for communication between mobile station (MS) and the GPRS network. The GGSN provides an interface to external packet data networks as well as to other GPRS network operators. Incoming packets are routed by the GGSN to the appropriate SGSN for a particular mobile station within its service area. Other network elements in the GPRS network architecture are Base Transceiver Station (BTS), Base Station Controller (BSC) and Home Location Register (HLR). The details of each of these network elements are available somewhere else [3.24, 3.26-3.28].
3.3.3 GPRS Protocols

A GPRS protocol structure is shown in Figure 3.9 [3.29]. The GPRS radio access protocols are involved in controlling the link between the mobile station and SGSN, and therefore are discussed in this section. Complete details of all the protocol structures in the GPRS network is available in the GPRS technical specifications [3.24, 3.30-3.33].

There are three main protocols that control the data transfer in GPRS packet mode transmission. The protocols are Subnetwork Dependent Convergence Protocol (SNDCP), Logical Link Control (LLC) protocol and Radio Link Control/Medium Access Control (RLC/MAC) protocol. These protocols are responsible for formatting raw user data into network packets for transmission over the GPRS network. As the IP network is used in most commercial GPRS networks, IP packets are considered in explaining GPRS data flows. GPRS access scheme is based on time division multiple access (TDMA) and is compatible with standard GSM structure. In the mobile station, user data is encapsulated into IP packets, which are then transformed into LLC frames by the SNDCP layer. Each LLC frame is further segmented into Radio Link Control (RLC) blocks at the RLC/MAC layer. The RLC blocks are then encoded and arranged into GSM time slots for transmission over the radio interface. A single RLC block requires four consecutive time slots for transmission. The relation between user medical data and GPRS packets transformation is presented in section 5.2.1.

The availability of radio resources in GPRS is optimised by introducing multi-slot capability, which allows concurrent access to radio resources in a more flexible way. The introduction of Packet Data Traffic Channel (PDTCH) increases the throughput of a single mobile terminal by allocating up to eight available time slots for a single user. A flexible resource allocation is also achieved by allowing available time slots to be dynamically allocated between up to eight users.
Figure 3.9  GPRS protocol structure
GPRS transmission is capable of providing a maximum theoretical speed of up to 171.2kbps with all eight timeslots offered to a single user. One GPRS timeslot can transmit different amounts of data depending on the coding schema (CS) adopted. There are four types of coding schema for GPRS transmission [3.26, 3.27]. All schemas have data and FEC except for coding schema 4 (CS4), as shown in Table 3.9 [3.34].

<table>
<thead>
<tr>
<th>Coding Scheme</th>
<th>FEC</th>
<th>User Bit Rate (kbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 1</td>
<td>1/2</td>
<td>9.05</td>
</tr>
<tr>
<td>CS 2</td>
<td>~2/3</td>
<td>13.4</td>
</tr>
<tr>
<td>CS 3</td>
<td>~3/4</td>
<td>15.6</td>
</tr>
<tr>
<td>CS 4</td>
<td>1 (No FEC)</td>
<td>21.4</td>
</tr>
</tbody>
</table>

Table 3.9 GPRS coding scheme

The European Telecommunications Standards Institute (ETSI) provides the specification for GPRS Mobile Station (MS) Classes that defines the behaviour of mobile phones regarding their operation in packet mode, circuit-switched mode or both [3.24, 3.35]. A Class A mobile phone supports simultaneous circuit-switched voice and packet-switched data services. Voice calls may be initiated and answered during a packet data transfer, and packet data transfers may be started and terminated during voice calls. A Class B mobile phone supports both circuit-switched and packet-switched services but the traffic exchange (either data or voice) must be only one at a time. For example, if the mobile phone is in a packet data transfer mode, the user may be paged for a voice call. If the mobile phone is in voice call, then the data transmission has to be suspended. A Class C mobile phone is only capable of non-simultaneous operation of either the circuit-switched or packet-switched services.
GPRS mobile phones are also classified based on the number of time slots they are capable of, known as multi-slot classes [3.36, 3.37]. The multi-slot classes determine the maximum data rates achievable for both downlink and uplink transmission. For example, a mobile phone with GPRS Class 10 specification is capable of four downlink and two uplink timeslots (referred to as 4+2) with a maximum of five active time slots. Active slots mean the total number of time slots the mobile phone can operate simultaneously for both uplink and downlink communications. The list of available GPRS multi-slot classes and the corresponding active slots is given by Table 3.10 [3.37].

<table>
<thead>
<tr>
<th>Multi-slot Class</th>
<th>Downlink Slots</th>
<th>Uplink Slots</th>
<th>Active Slots</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>11</td>
<td>4</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 3.10  GPRS multi-slot classes

The simultaneous operation of voice and data services in GPRS is known as Dual Transfer Mode (DTM). In theory, there are two channelisation methods defined for DTM, namely the single-timeslot method and multiple-timeslot
method [3.35, 3.38]. In the single-timeslot method, a half-rate traffic channel (TCH) for voice data is combined with a half-rate packet data traffic channel (PDTCH) in the same single timeslot. The multiple timeslot method comprises of a single timeslot for TCH, plus one or more adjacent PDTCH timeslots for packet data transfer. The capability of DTM and GPRS mobile class is very much depending on the mode supported by network operators, which determine the type of services, GPRS users able to experience.

3.4 Summary

The design concepts involved in implementing the m-Health system are presented in the chapter. The concept includes the consideration of types of clinical data for transmission and the combination of wireless technologies that lead to the concept of mobile bio-monitoring. Although the types of clinical data adopted in the research are the physiological signals applicable to monitoring and routine check-ups, the m-Health system is capable of having different signal inputs should the system be intended for different medical applications. The combination of wireless technologies offers a flexibility of use for the patient with the Bluetooth interface and exploits a wide area network connectivity with the cellular network. An overview of the Bluetooth technology and the GPRS network are described, focusing on the specifications that are directly related to the system implementation.
3.5 Reference


[3.30] 3GPP, "Digital cellular telecommunications system (Phase 2+); Mobile Station (MS) - Serving GPRS Support Node (SGSN); Subnetwork Dependent Convergence Protocol (SNDCP)," TS 44.065 version 6.3.0 Release 6, September 2004.

[3.31] 3GPP, "Digital cellular telecommunications system (Phase 2+); General Packet Radio Service (GPRS); Mobile Station (MS) - Base Station System (BSS) Interface; Radio Link Control/ Medium Access Control (RLC/MAC) protocol," TS 44.060 version 6.11.1 Release 6, February 2005.


[3.33] 3GPP, "Digital cellular telecommunications system (Phase 2+); General Packet Radio Service (GPRS); Base Station System (BSS) - Serving GPRS Support Node (SGSN); BSS GPRS Protocol," TS 48.018 version 6.7.0 Release 6, January 2005.


[3.37] 3GPP, "Digital cellular telecommunications system (Phase 2+); Multiplexing and multiple access on the radio path," TS 45.002 version 6.8.0 Release 6, January 2005.

[3.38] 3GPP, "Digital cellular telecommunications system (Phase 2+); Physical layer on the radio path; General description," TS 45.001 version 6.5.0 Release 6, November 2004.


CHAPTER FOUR:

SYSTEM HARDWARE DESIGN

4.1 Overview of the m-Health System

The main idea of the system is to allow a remote patient to capture clinical information and then transmit the physiological signals to a hospital via a cellular network. The received patient data is then displayed to a clinician for monitoring and assessment purposes. The clinician has the option either to contact the patient directly or to send a message to inform the patient about the outcome of the evaluation. The overall diagram of the mobile telemedicine system is given in Figure 4.1.

The physiological signal inputs from the patient are those normally used for monitoring purposes, such as Electrocardiogram (ECG, from which heart rate can be derived), body temperature, blood pressure, and oxygen saturation (SPO2). The input signals are then fed into an embedded Bluetooth telemedicine processor module to be digitised and encapsulated into Bluetooth packet format. Next, the digitised data are transmitted to a Bluetooth mobile phone and subsequently via the GPRS cellular network to a base station, then to a hospital via a packet data network.
Figure 4.1 Multi-channel GPRS-based m-Health system
The essence of the system presented here contains the following links, as shown in Figure 4.1:

- a short-range Bluetooth link from the patient to a mobile phone;
- a mobile cellular network (GPRS) to a base station and subsequently to an external Packet Data Network (PDN);
- a packet data network based on an Internet Protocol (IP), for example an Internet Service Provider (ISP), to a hospital server;
- a local area network to a clinician;
- a mobile cellular network (GPRS) from a hospital server to a clinician.

It is essential that the data transmitted to the hospital are archived and made available to clinicians as and when required, therefore the research includes the development of a hospital-based server. The incoming data from the patient includes the International Mobile Subscriber Identity (IMSI) number, which is unique to the Subscriber Identity Module (SIM) card present in the mobile phone. This number is used to identify the patient to the system and to permit the storing of additional records for that patient. Acknowledgment messages are then sent from the server to the mobile unit to indicate a successful telemedicine session. Once the clinician has been identified to the telemedicine database, a full list of telemedicine sessions recorded for a given patient can be displayed and individual records can be examined from the clinician’s own office using the internal LAN. Alternatively, the clinician who is away from the vicinity of the hospital may receive a patient’s data via the GPRS network, either through a mobile phone, a laptop or a personal digital assistant (PDA).
4.2 m-Health Processor Module

The m-Health processor module is based around an embedded processor running a proprietary operating system primarily designed to support the acquisition of data from biomedical sensors [4.1]. The module is responsible for sampling the medical signals from sensors on the patient and transforming the digitised medical data into a Bluetooth packet structure for a short-range transmission to a mobile phone.

The m-Health processor module comprises three sub-systems, as shown in Figure 4.2:

- a biomedical sensors unit;
- a microcontroller board;
- a Bluetooth communication module

![Figure 4.2 Schematic of m-Health processor module](image)

The processor module is shown in Figure 4.1 as part of the patient at remote locations section of the overall system. The biomedical signal inputs are fed into the m-Health processor module via the biomedical sensors unit. The biomedical signal inputs are ECG, body temperature, blood pressure, and oxygen saturation (SPO2). The biomedical sensors unit receives multi-channel inputs from an ECG signal conditioning module, a temperature sensor, a
blood pressure meter and a pulse oximetry finger probe. Connectors are used to provide a secure interface between the sensors and the sensor unit inputs. The input channel numbers and the corresponding types of medical signal assigned to the input channels are listed in Table 4.1.

<table>
<thead>
<tr>
<th>Channel Number</th>
<th>Type of Medical Signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channel 1</td>
<td>ECG</td>
</tr>
<tr>
<td>Channel 2</td>
<td>Body Temperature</td>
</tr>
<tr>
<td>Channel 3</td>
<td>Non-Invasive Blood Pressure</td>
</tr>
<tr>
<td>Channel 4</td>
<td>Oxygen Saturation</td>
</tr>
</tbody>
</table>

Table 4.1 Input channel numbers and the type of medical signals

The choice of sampling frequency for the multi-channel biomedical signals acquisition process is dominated by the minimum sampling frequency required for ECG signals. The selection of the appropriate sampling frequency for ECG signals depends on two main criteria: the ECG bandwidth and the desired signal processing task [4.2]. The ECG bandwidth for monitoring applications is in the range of 0.5 to 50Hz [4.3, 4.4] and is appropriate for the telemedicine sessions from a remote patient. The strength of an ECG signal is mainly concentrated between 0.25 and 35Hz, and in particular, most of the QRS complex is between 2 and 20Hz with a maximum at 12Hz [4.2, 4.5].

Sampling theory indicates that a sampling rate of twice the upper band limit (i.e., the Nyquist rate) is sufficient for digital signal reproduction and to minimise the aliasing – the creation of an inaccurate digital signal due to a low sampling rate [4.6]. Since the ECG bandwidth required for the mobile telemedicine application is in the range of 0.5 to 50Hz, an analogue-to-digital conversion sampling rate of 100 samples per second satisfies the minimum Nyquist rate requirement. In practice, however, a minimum sampling frequency for a fair representation of ECG signals is 128Hz and ambulatory ECG monitors also require a relatively low sampling frequency of 128Hz [4.5, 4.7]. Some commercial digital holter systems operate with an ECG sampling
frequency of 128Hz, for example, Marquette Holter System series 8500 and Medilog FD5 Digital Holter Recorder with 10 bits resolution [4.2, 4.7, 4.8].

To limit the amount of multi-channel data to be handled by the limited Random Access Memory (RAM) capacity on the microcontroller, an Initial choice of 150Hz sampling frequency is implemented. The use of 150Hz corresponds to the minimum ECG sampling frequency for commercial ambulatory and monitoring systems. With 10 bits accuracy of the ADC unit in the processor module, the ECG data are then wrapped into a predefined packet structure, as later shown in section 5.1.1. The packet is transmitted to the Bluetooth module via a Universal Asynchronous Receiver Transmitter (UART) port at 9.6 kbps.

4.2.1 Atmel Microcontroller Board

The use of an Atmel microcontroller development board is adopted for the research to allow rapid prototype development of the mobile telemedicine system. The microcontroller development board used is based on an 8-bit Atmel T89C51AC2 microcontroller. The board integrates all the circuitry required in order to implement a data acquisition unit, with the capability of mounting and interfacing with other Integrated Circuits (ICs), for example a Bluetooth module for short-range wireless connectivity. The board provides microcontroller supervisory circuitry such as a power supply connector and voltage regulator, I/O connector for access to microcontroller port pins, an RS232 communication interface, LEDs for status indicator, crystal clock and reset circuitry. The board uses an on-board 18.432MHz crystal clock.

The microcontroller board also supports Atmel's FLIP (Flexible In-System Programmer) that allows rapid set-up of the desktop environment for downloading the compiled program to the target microcontroller. The In-System Programming (ISP) capability allows successive downloads of the
program directly into the board. The ISP capability also removes the need of a traditional method of removing the microcontroller and places it in a programmer for downloading the compiled program.

The main feature of the T89C51AC2 microcontroller is the on-chip memory capacity that includes a 32 kB of Flash memory for user program code, together with a 1 kB of RAM for storing temporary variables. The RAM is the memory that is dynamically allocated during run-time, and there must be sufficient RAM to complete a task. A 2 kB EEPROM is also included for retaining preset values while the power is disconnected. The microcontroller also features a 2 kB on-chip Flash for the Atmel boot loader and for the Application-Programming-Interfaces (APIs) required for the ISP. The APIs allow programming the Flash memory of the microcontroller using a standard RS232 connection and the Atmel’s FLIP programming tool.

The microcontroller also provides a total of eight input channels of analogue to digital converter (ADC) with 10-bit accuracy and a UART communication interface. Initially, only four out of the eight analogue input channels are utilised for the telemedicine processor module. The output data from the microcontroller board is set to be in RS232 format and therefore is connected to the RS232 input interface on the Bluetooth communication module. In addition, the board is also capable of having direct digital input/output (I/O) connections should the features be required in a system. The eight input channels available in the microcontroller can be used for digital Input/output or as analogue inputs for the ADC. Since four out of the eight inputs are configured for the ADC analogue inputs, four more Inputs are available to be utilised as digital I/O connections. These connections can be configured by setting the parameters in the ADC Configuration Register during the initialisation process.
4.2.2 Bluetooth Communication Module

A Bluetooth communication module is integrated into the telemedicine processor module to provide a short-range wireless connectivity with the mobile phone. The module chosen for the research is the BlueWAVE RS232 PCB module from Wireless Futures UK Limited [4.10]. The module is designed to provide a standard Bluetooth interface conforming to the Bluetooth version 1.1 standard. The BlueWAVE Bluetooth module consists of a Bluetooth slave module that uses the serial port profile from the Bluetooth profiles specifications to provide a serial data connection with a remote Bluetooth device.

The BlueWAVE Bluetooth module is capable of establishing a serial data link with a remote device without a Bluetooth protocol stack running on the main microcontroller host. This is possible since the Bluetooth module provides a fully embedded Bluetooth protocol stack, ranging from the physical and lower layer of Radio, Baseband and Link Management Protocol up to the upper RFCOMM layer. The link establishment protocols require no interaction from the host system and the embedded Bluetooth stack reduces the processing load on the processor host.

The Bluetooth module is specified as power class 1 category with the highest output power of 100mW (20 dBm) and typical operating range of 100m. The connection between the Bluetooth module and the microcontroller board is based on the RS232 communication interface. The pin assignments of the Bluetooth module's connectors are shown in Table 4.2. Further details of the module are available from the data sheet [4.10].
Since the Bluetooth module is a slave module in a Piconet, the module is only capable of one active connection at a time. Therefore, once the telemedicine processor module is paired successfully with a mobile phone, other remote devices will not be able to discover the module. The mobile phone becomes the master device and controls the connection. The connection uses the Asynchronous Connection-Less (ACL) Bluetooth link, intended to support data applications. The telemedicine processor module remains undiscernable until the current active connection is terminated and subsequently unpaired by the master device. The module will then revert to discoverable mode, allowing other devices to see and connect to the telemedicine processor module. The master and slave concept for the Bluetooth link establishment is illustrated in Figure 4.3. The Bluetooth implementation within the telemedicine processor module is based on host-less implementation concept (as in Figure 3.10), as previously discussed in section 3.2.2.
4.2.3 Medical Sensing Nodes

Three-lead ECG monitoring was initially implemented with the telemedicine processor module. This is simple to apply and the gathering of useful ECG data is also very tolerant of misplaced leads. In the implementation of the mobile telemedicine system, keeping a simple approach to the data acquisition process is essential. The three-lead ECG is appropriate in a remote situation, considering that the telemedicine processor module will be mainly used by non-medically trained people.

The ECG amplifier and sensor module used with the telemedicine processor module is a commercially available Vernier EKG/ECG Sensor module [4.11]. The ECG module uses a three-lead system (with three electrodes), which is commonly used for single channel ECG sensors. The module provides the amplification and filtering necessary to reproduce the ECG signal. The module consists of an instrumentation amplifier, an isolation amplifier and a band-pass filter, which are standard components in an ECG amplifier.

The ECG sensor module is powered directly by the 5V rail of the microcontroller board. The module has a gain of 1000, which amplify a 1mV body potential to a 1V of sensor output. The range of frequencies applicable
to ECG signals is as previously discussed in section 3.2.1. The electrode patches used with the module is the silver-silver chloride (AgCl) electrode that approximates a non-polarisable electrode and has a very small offset potential [4.3]. Movements of these electrodes causes a much smaller baseline shift in the amplified ECG than that of a polarisable electrode such as nickel-plated brass electrode.

An example on how the electrodes are placed on the patient is illustrated in Figure 4.4. The example given is based on the electrodes positioned for the Lead I ECG signal. The first electrode is placed on the inside of the right elbow and is connected to the green (negative) clip. The second electrode is placed on the inside of the left elbow and is connected to the red (positive) clip. The third electrode is placed on the inside surface of the area behind the right ankle bone. It is connected to the 'reference' black clip to provide a reference point for the ECG baseline. In practice however, the third electrode is placed on the right wrist for flexibility of the arrangement and provides the same reference point for the ECG baseline.

![Figure 4.4 Electrodes positions for Lead I ECG reading on a patient](image)
The core body temperature is acquired by measuring the potential difference across a temperature sensitive device. The device used in measuring the temperature is an LM35 Precision centigrade temperature sensor [4.12]. The LM35 is a precision integrated circuit temperature sensor with an output voltage that is linearly proportional to the Celsius temperature. The temperature sensor provides a linear change in voltage output that depends on the ambient temperature. The output voltage is fed to the analogue input in the telemedicine processor module.

![Diagram of temperature sensor setup](image)

**Figure 4.5** Set up of a temperature sensor device

The set-up of the temperature sensor as shown in Figure 4.5 provides a temperature reading between 20 Celsius to 1500 Celsius. The core body temperature of interest is approximately in the region of 370 ± 30 Celsius.

One of the analogue inputs of the telemedicine processor module is set to read a non-invasive blood pressure (NIBP) during a patient monitoring session. The input channel is capable of reading the systolic pressure, which corresponds to the maximum pressure, and the diastolic pressure, which corresponds to the minimum pressure. During the development of the telemedicine processor module prototype, the NIBP reading is simulated by changing a potentiometer value set as the analogue input to indicate signal variation in blood pressure readings.
Oxygen saturation values obtained from pulse oximetry (SpO2) are one part of a complete assessment of the patient's oxygenation status. Pulse oximetry is a non-invasive measurement technique that enables measurement of blood oxygen saturation. The pulse oximetry finger probe implemented with the telemedicine processor module is the Viamed SPO2 Finger Probe [4.13]. The probe is placed at the end of patient's finger, as shown in Figure 4.6. The cable from the finger probe is connected to the analogue input of the telemedicine processor module and is assigned to channel 4 analogue input. Oxygen saturation is indicated as a percentage and normal values for a healthy individual are in the region of 97% to 99%.

![SPO2 finger probe connection](image)

Figure 4.6  SPO2 finger probe connection

### 4.2.4 Power Consumption and Physical Dimensions

The microcontroller board is equipped with a heavy duty power supply circuit, with a 5V voltage regulator designed for maintaining a 5V supply to the circuit under quite high load conditions. The DC input voltage to the board can be between 9 and 12 V, and capable of supplying at least 500mA of current. Output voltage of 5V is available at the connector to supply to an external hardware.
The average current drain of the microcontroller board in an active mode is about 8.6mA, and 6.8mA in an idle mode. In its power-down mode, the current drain is less than 160 μA. For future developments, the ZigBee standard will be considered to optimise the needs of remote monitoring using very low power consumption [4.14].

The DC input voltage to the Bluetooth Module is between 3.5V and 9V. The supply voltage on the Bluetooth Module is applied to pin number 1 and the ground voltage is applied to pin number 12, as specified in Table 4.2. The voltage supply must be able to supply a minimum of 100mA. The voltage supply is regulated on the Bluetooth Module to 3.3V. The Bluetooth module is powered by the microcontroller board via the 5V output voltage available at the connector.

The telemedicine processor module prototype has a choice of two power supply connections. The prototype has a standard plugpack connection input, which can be used to supply power to the telemedicine processor module directly from a wall socket using a universal regulator. The universal regulator used with the telemedicine processor module prototype is a Pro-Power IC Universal Regulator (Model PP500R), supplying 600mA of current at 9V supply voltage. The second supply connection option is from a 9V battery, which is utilised when the prototype is used in a portable and mobile environment, for example during a field trial around Leicestershire. The 9V alkaline battery has a rating of 550mAh and is connected to the polarised header supply connection in the prototype. The portable operation of the telemedicine processor module prototype with the 9V battery allows over an hour of continuous operation, i.e., equivalent to 6 to 7 of 10-minute telemedicine sessions.

The telemedicine processor module prototype has physical measurements of 155 (length) x 90 (width) x 55 (height) mm and weight about 500g. The prototype has analogue inputs on the front panel and power switch on the
side panel. The photograph of the prototype, together with the ECG sensor module is shown in Figure 4.7.

![m-Health processor module prototype](image)

Figure 4.7  m-Health processor module prototype [4.1]

### 4.3 Mobile Phone

The choice of a mobile phone model adopted for the research was determined by its functionality. The most important feature on the phone is the Bluetooth connectivity function and the software development capability. In addition, the choice of model is also based on the mobile phone specification that can utilise the GPRS multi-slot capability. The multi-slot feature helps to optimise the availability of radio resources in the GPRS network. Since a GPRS multi-slot class determines the amount of maximum up-link timeslots supported by the mobile phone, the choice was made with a phone that can transmit on more than one up-link timeslot. During the development stage, the choices of models that met the required functions were limited. At the time, a GPRS multi-slot Class 6 mobile phone offered two simultaneous up-link timeslots, which was the highest number available.
Besides the multi-slot capability, the specification for the mobile phone also provides the GPRS classes. These classes define the behaviour of mobile phones in terms of their operation in packet mode, circuit-switched mode or both, as specified by the European Telecommunications Standards Institute (ETSI) [4.15]. The majority of commercially available GPRS mobile phone models are of Class B, which support both circuit-switched and packet-switched services but the traffic exchange (either data or voice) must be only one at a time.

The Nokia 6600 is a Class B, GPRS multi-slot class 6 mobile phone, which was among the earliest models that are commercially available and supports software application development with Java technology on its Series 60 platform [4.16, 4.17].

4.3.1 Mobile Phone Specifications

Some of the main specifications that are directly related to the overall design of the mobile telemedicine system are presented here. The specifications ranging from the software development compatibility to the type of GPRS multi-slot classes that are supported.

The Series 60 platform provides the foundation for the mobile phone application development for the mobile telemedicine system. The Nokia 6600 mobile phone is running on the Symbian operating system that is responsible for normal operation and application management. The list of Java technology supported by the mobile phone determines the compatibility of the mobile phone application program, developed for patient communication gateway interface in the research. Further details on the Series 60 platform, Symbian operating system and Java technology are explained in the mobile phone application software in Chapter 5.
CHAPTER FOUR

The (3+1/2+2 slots) Class 6 GPRS Multi-slot specification indicates that the Nokia 6600 mobile phone has a maximum of four active time slots. Active slots mean the total number of time slots the mobile phone can operate simultaneously for both up-link and down-link transmissions. The mobile phone is capable of either three down-link and one up-link timeslots or two down-link and two up-link timeslots, depending on the network application running on the mobile phone. For example, if a user is performing a web browsing application on the mobile phone, then the mobile phone utilises the three down-link and one up-link timeslots configuration, since more data is downloaded into the mobile phone. If the mobile phone transmits data to a destination, as it is in the patient communication getaway in the mobile telemedicine system, then the two down-link and two up-link timeslots configuration is utilised.

Since Class B supports either data or voice operations at a given time, then if the mobile phone is transmitting medical data to a hospital server during a telemedicine session, the patient is not able to use the mobile phone for a voice call at the same time. If the patient does make a voice call during the medical data transmission, then the data operation is suspended and will be resumed after the voice call is terminated.

The limitation on GPRS mobile phone class and operation mode supported by mobile network operators determines the type of services and data operation GPRS users are able to experience.

4.3.2 GPRS Network and Mobile Terminal Configuration

The Vodafone GPRS network in the UK is provided by the Vodafone Group, as part of the Group's main products and services. The network coverage of the Vodafone GPRS network is currently available over most of the UK and to over 99% of the UK population. In 2002, Vodafone had 93 million mobile
subscribers and its operations had reached over 29 countries in every
continent in the world. This makes the Vodafone Group the largest
wireless/mobile phone operator in the world.

The decision in using the Vodafone GPRS network in the research was based
on a few factors from a practical point of view. The first factor is that the
Vodafone mobile network is one of the two choices for mobile network
contracts authorised by the university purchasing policy. The Vodafone
network was chosen since it offers a better package for the Nokia 6600
mobile phone than the Orange network at the time. The main factor is that
the Vodafone GPRS network provides a much more global roaming capability,
over 29 countries from different part of the world. In principle, the global
roaming capability allows the mobile telemedicine system developed in the
research to be deployed in different parts of the world without any
modification.

As for the pricing of GPRS data users, the cost is only based on the amount of
data transmitted. For example, the pricing structure is based on the amount
of data in Mbytes and the current price for the Vodafone GPRS network is
about £2.35 per Mbytes [4.20]. In the network architecture, the Serving GPRS
Support Node (SGSN) and the Gateway GPRS Support Node (GGSN) are
implemented together as a Combined GPRS Support Node (CGSN) [4.21] only
supports the CS1 and CS2 type of GPRS channel coding schemes.

The GPRS connection configurations required on the mobile phone are
specific to the Vodafone GPRS network settings. The main part is to configure
the proper Access Point Name (APN) for medical data transmission in the
mobile telemedicine system. The APN is a name of the point of entry to
external packet data networks from a GPRS network [4.22]. The significance
of the choice of APN for a mobile phone's application is that the service
requested by the mobile phone will be routed to the appropriate gateway for
the connection with the external network. For example, if a request for
general Internet access is mistakenly set up to use an APN that was specified by the network operator for Wireless Application Protocol (WAP) or Multimedia Messaging Service (MMS) services only, then there will be no routing of Internet traffic further than these gateways.

There are two APNs available from the Vodafone GPRS network, which are the Vodafone Internet GPRS and the "wap.vodafone.co.uk" access points. The Nokia 6600 mobile phone is configured to work with any of these two APNs. The Vodafone Internet GPRS APN is the point of entry of data packets from the GPRS network to the Internet as the external packet data network. The Vodafone MMS and the Vodafone Live services use the "wap.vodafone.co.uk" APN as a gateway. The Vodafone MMS connection is the configuration with a Vodafone MMS server for receiving and sending MMS messages. The Vodafone Live connection is used for browsing the WAP services from the Vodafone network.

The architecture of the mobile telemedicine system is designed to work with an Internet as the external packet data network. Therefore, the GPRS connection for the telemedicine sessions is configured to work with the Vodafone Internet GPRS access point. The access point routes all the data packets from the mobile phone to the destination server at the hospital during a telemedicine session. The mobile phone application program developed for the patient interface will request the patient to confirm the choice of Vodafone Internet GPRS access point before establishing a network connection with the hospital server. Further details on the mobile phone application program are described in section 6.4.

The next part of the mobile phone configuration is to determine the phone's identification and the user identity. The information obtained from the configuration is used to identify and verify a patient to the mobile telemedicine system, as part of the security features implemented in the system. The International Mobile Subscriber Identity (IMSI) number is a 15-
digit number that provides a unique way to identify a patient or a subscriber of a mobile phone connection. The IMSI number is unique to the Subscriber Identity Module (SIM) card present in the mobile phone. The structure of an IMSI number is given in Figure 4.8. The IMSI number is composed of three parts [4.22]:

1) Mobile Country Code (MCC) consisting of three digits. The MCC identifies uniquely the country of residence of the mobile subscriber;

2) Mobile Network Code (MNC) consisting of two or three digits. The MNC identifies the home mobile network of the mobile subscriber.

3) Mobile Subscriber Identification Number (MSIN) identifying the mobile subscriber within a mobile network.

![Figure 4.8 The structure of an IMSI number](image)

In addition, an International Mobile Equipment Identity (IMEI) number, which uniquely defines a particular mobile phone unit, is also available to provide a secondary identification of the patient. The combination of IMSI and IMEI numbers produces a secure and reliable patient identification feature in the mobile telemedicine system. These numbers also help to prevent and block any unauthorised attempts to access the system if the mobile phone is lost or stolen. All the identification information of the mobile phone is accessed by sending AT Commands to communicate with the mobile phone. The list of AT Commands available to access this information and how the AT Commands can be used are explained in Appendix E.
4.4 Summary

The chapter presents a new system of transmitting a patient's physiological signals directly to a hospital for monitoring applications, via a mobile phone. Conceptually, the packet transmission of four channels of biomedical data originates from a GPRS network and terminates at a hospital server in a packet data network. All the biomedical data are made available to a clinician, together with a patient's medical information, as and when required from a telemedicine database system. The system hardware design describes the hardware and modules implemented in the m-Health system. The design represents different parts of the overall system, from the biomedical data acquisition using the telemedicine processor module to the GPRS network transmission using a mobile phone. The embedded Bluetooth telemedicine processor module samples signals from up to four sets of sensors attached to the patient and transmits the digital data over a Bluetooth link to a mobile phone. The Nokia 6600 mobile phone used in the research is configured to work with the Vodafone GPRS network in the UK and controls the network connection with the hospital server. The data transmission includes the International Mobile Subscriber Identity number, which is unique to the Subscriber Identity Module card in the mobile phone and is part of the security features developed for the system.
4.5 References


[4.22] 3GPP, "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Numbering, addressing and identification," TS 23.003 version 6.6.0 Release 6, March 2005.

[4.23] 3GPP, "Digital cellular telecommunications system (Phase 2+); AT Command set for GSM Mobile Equipment (ME)," TS 07.07, March 2003.
5.1 Overview of m-Health System Software and Protocols

The integrated software implementation in the multi-channel m-Health system is summarised in Figure 5.1. The software functionality on the patient interface part consists of the m-Health processor module software and the patient user interface programs. The GPRS and IP protocols adopted for the system are based on the protocols implemented by mobile network operators. The server applications and the browser interface programs are part of the software functionality on the hospital and clinician interface. The software development in this chapter represents the development of the patient interface part of the system. The chapter focuses on the m-Health processor module and mobile phone software of the patient-user interface programs. The mobile phone software is later used, as described in Chapter Six, as a platform for the doctor browser interface programs. The GPRS and IP protocols implemented for the uplink transmission of the biomedical data to the hospital server are also discussed in detail.
Figure 5.1 Integrated software implementation of multi-channel GPRS-based m-Health system
5.2 m-Health Processor Module Software

The telemedicine processor module software development platform performs two main tasks; the development of program code for the microcontroller and the configuration of the Bluetooth module integrated in the telemedicine processor module. The first software development platform for the telemedicine processor module is set up based on the hardware and the type of microcontroller adopted for the system development. The development environment is chosen based on the requirement and the capability of the Atmel T89C51AC2 low power 8-bit microcontroller with 8051 architecture. The Atmel microcontroller is capable of In-System Programming (ISP) that allows code modification over the total lifetime of the system [5.1]. This capability is utilised with the FLIP (Flexible In-system Programmer) utility for microcontroller programming supported by Atmel.

The telemedicine processor module software was designed using Keil 8051 Microcontroller Development Tools [5.2]. The tool provides a development environment for designing, simulating and debugging the program code for the microcontroller. The development environment supports either 8051 assembler or C programming to program the microcontroller. The main advantage of the chosen development environment is its capability to simulate assembler or C code of the 8051 architecture to test and verify the program code without the need of running the actual program on the hardware. The FLIP software is later used to download the final program code into the Flash memory of the microcontroller.

The second development platform is set up to configure the Bluetooth module. The platform allows the use of AT commands to set the parameters of the Bluetooth module. The configuration is done by invoking the module into Command mode and uses the set of AT commands. A terminal application, in this case a Microsoft HyperTerminal, is configured to
communicate with the Bluetooth module via the serial interface for the module's configuration process. The module enters the Command mode after receiving three consecutive ASCII '+' characters over its RS232 interface.

5.2.1 Main Software Program

The main program for the telemedicine processor module is implemented in low-level assembly language. The program is responsible for the whole operation of the module, including the transmission of medical data via the Bluetooth link. The program executes the same routine continuously until the system is reset, at which time the program will reinitialise. This adds an additional safeguard should the system crash due to an unexpected hardware fault, an unexpected internal hardware exception, etc. The initialisations involve setting up the clock register, the timer and interrupt modes as well as the baud rate for the internal UART connection.

The telemedicine processor module performs the following routines sequentially on each of the four enabled channels on the microcontroller. The codes first identify the channel number and perform the analogue to digital conversion (ADC) on the channel. The digitised data are then wrapped into a packet structure as shown in Figure 5.2.

![Figure 5.2 Packet structure of m-Health processor module](image)

The packet header indicates the start of a new packet of data and the channel number of the analogue input. The trailer signifies the end of a packet and the whole packet structure is in hexadecimal format. The packet is then sent to the Bluetooth module via the UART port at 9600-baud rate. The
Bluetooth module handles the complete transmission protocols of the Bluetooth packets. The Bluetooth module runs an embedded Bluetooth protocol stack and reduces the processing burden on the microcontroller host. The codes then repeat the same procedures for the next analogue input channel. The codes produce a sync character at the end of every procedure for channel number four. The data output format of the telemedicine processor module is shown in Figure 5.3. The character 'T' indicates the trailer of the packet from each channel. The flowchart of the main software program is shown in Figure 5.4.

![Figure 5.3 Digitised data output format of m-Health processor module](image)

![Figure 5.4 Flowchart of the main software program](image)
5.2.2 Bluetooth Module Configuration Program

The Bluetooth module is capable of performing different functions, depending on the application requirements. The baud rate of the module for the serial interface connection can be configured between 1200 to 230,400 baud. The module has a sleep mode if the system is in an idle state and there is also a function to change the password of the module.

The configuration program of the Bluetooth module is an interface on the HyperTerminal application on a standard desktop environment. The terminal application is configured to communicate with the module via a serial interface. The list of all the compatible configuration commands for the module is available in the module’s datasheet [5.3]. The sequence of procedures in configuring the Bluetooth module is given below:

1. Connect the Bluetooth module to a serial interface available on a personal computer.
2. Open a HyperTerminal application and set configuration to 115200 bps, no parity, 8 data bits and 1 stop bit.
3. Power up the module to a 5V supply.
4. Type ‘+++’ then enter to access the Command mode.
5. Type ‘AT+BWB=3’ then enter to set baud rate to 9600.
6. Type ‘AT+BWE’ then enter to exit the Command mode.

In Command mode, no data can be received from a remote Bluetooth device. Data received from the local RS232 interface is used for the configuration process. The Bluetooth module has a green LED indicator to indicate its current state. The indicator is very useful during the configuration of the module and during the development of the telemedicine processor module. The status of the indicator provides an insight to the current operation of the telemedicine processor module. The LED will flash once on power up and the description of the subsequent LED status is described in Table 5.1.
5.2.3 Bluetooth Link Establishment Protocols

Once the Bluetooth module has been successfully integrated into the telemedicine processor module, standard link establishment protocols are required in order to connect to the telemedicine processor module. In general, a remote device needs to pair with the telemedicine processor module for every connection. This is one of the security features implemented to ensure authorised use of the module. On power up, the telemedicine processor module is in Data mode and in Data mode, all data are transmitted from the module to a remote Bluetooth unit and vice versa. The data exchange takes place during the pairing up process. The telemedicine processor module is a slave unit in a Bluetooth piconet.

The module is set to a specific password and the pairing remote device will have to request for the password. When paired successfully, the remote master device will be able to see and connect to the telemedicine processor module. The connection is a virtual serial port connection to the module and data can then be transmitted in both directions. If the connection is dropped by the master unit then the telemedicine processor module will revert back to discoverable mode, allowing any other device to see and request a connection.

<table>
<thead>
<tr>
<th>LED Indication</th>
<th>Mode</th>
<th>Bluetooth Connection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single flash</td>
<td>Data</td>
<td>None</td>
</tr>
<tr>
<td>Slow Flash (1Hz)</td>
<td>Command</td>
<td>None</td>
</tr>
<tr>
<td>Quick Flash (2Hz)</td>
<td>Command</td>
<td>Active</td>
</tr>
<tr>
<td>LED Off Steady</td>
<td>Data</td>
<td>None</td>
</tr>
<tr>
<td>LED On Steady</td>
<td>Data</td>
<td>Active</td>
</tr>
</tbody>
</table>

Table 5.1 Description of the LED status in the Bluetooth module
5.3 GPRS and IP Protocols

A GPRS network allows users to send and receive data in an end-to-end packet transfer mode based on standardised network protocols supported by the GPRS bearer services. GPRS supports applications and interworking with external packet data networks based on Internet Protocol (IP) and X.25 network protocols [5.4, 5.5]. Currently IP-based protocols and services are widely used in the Internet, and are continuously being developed for providing better performance in wireless environment. One of the well-known services is the Internet's World Wide Web (WWW) application.

IP is used as the network layer protocol for the GPRS backbone, for example to connect to Serving GPRS Support Node (SGSN) and Gateway GPRS Support Node (GGSN). A new tunnelling protocol is defined in the backbone network, known as GPRS Tunnelling Protocol (GTP) [5.6]. Tunnelling is a two-way point-to-point process of transferring encapsulated data units within and between the mobile networks from the point of encapsulation to the point of decapsulation. The new protocol is built on top of an IP network to handle mobile terminal mobility, and to support registration and authentication procedures.

Each registered GPRS user who wants to exchange data packets with the IP network gets an IP address. The address resolution between IP address and GSM address is performed by the GGSN, using the appropriate Packet Data Protocol (PDP) context, a concept that will be explained in the next section. The GPRS address can be either dynamic (e.g. the user's IP address is allocated from a pool of unused IP addresses every time the subscriber activates the access to an IP network) or static (e.g. a certain IP address is permanently allocated to a particular subscriber) [5.5]. GGSN is the first node that processes IP packets from a mobile terminal to an external Packet Data Network (PDN), for example an Internet host [5.7]. It means that GGSN is the first hub for every IP packet regardless of packet destinations.
In most commercial GPRS implementations, a mobile terminal is dynamically assigned a private IP address [5.8]. Therefore a Network Address Translation (NAT) server is required to translate the IP addresses of the packets delivered between the private IP address domain of the GPRS network and the public IP address domain of the external packet data network.

The private IP address means that the IP address are reserved for use on private networks, and should never appear in the public Internet. The Internet Assigned Numbers Authority (IANA) has reserved three blocks of IP address space for private networks, as shown in Figure 5.5 [5.9].

![Private IP address blocks](image)

The use of a private IP address is introduced to allow private local networks in different organisations to use the same IP addresses, as long as these self-contained clouds of IP address users stay within their own network and are not connected to the public network. These private IP addresses can be translated to public IP addresses by the NAT. Details on specifications of a private IP address are available from the technical and organizational notes of the Internet Engineering Task Force (IETF), known as Requests for Comments (RFC) document series [5.10, 5.11].

In contrast, the public IP address means that the IP addresses are visible to the public and allow other users to know about and access the computer, for
example a Web server. The specific security and authentication process implemented in a server determines the authorised use of the connection.

5.3.1 GPRS Data Flow

Figure 5.6 summaries the packet delivery flows in the m-Health system. The system starts with Bluetooth packets from the m-Health processor module which are then transformed into the GPRS IP packets by the mobile phone. The GPRS IP packets are later translated into a standard IP packet for TCP/IP transmission in a public network by the NAT. The application server then performs the necessary protocols to recover the original medical data sent by the patient.

The section now focus on how the digitised medical data received by the mobile phone via its Bluetooth link is encapsulated into the required segments and frames based on GPRS radio access protocols. The overview of the data flow process and the relevant transformations at each protocol layer is illustrated in Figure 5.7. The multiplexing of the multi-channel medical data is performed by the processor module. The multiplexed data received by the mobile phone are now the user payload at the application layer. The first stage of the transformations is the encapsulation of the multi-channel medical data from the application layer into a TCP segment format for the transport layer. The TCP segment includes information on the destination port address of the application server, which is set to arbitrary port number 2206. The next part is the formation of an IP packet for the network layer that includes, among other things, the destination IP address of the application server, which is 158.125.51.22.

The IP packet is then forwarded to the Subnetwork Dependent Convergence Protocol (SNDCP) layer where the IP packet is formatted for packet transmission over the GPRS network [5.12, 5.13]. The SNDCP layer is
Figure 5.6 Packet delivery flows in m-Health system
Figure 5.7  GPRS data flow process and relevant transformations at each protocol layer
responsible for segmentation of network protocol data into the Logical Link Control (LLC) protocol data and handles data transfer between SGSN and mobile station. At the SNDCP layer, digitised medical data that have been encapsulated into the IP packet are then transformed into SNDCP segments. The next part is the formation of LLC frames at the LLC layer. If the size of the IP packet does not exceed the maximum LLC frame size of 1520 bytes, each IP packet is mapped onto a single LLC frame. The process of adding several layers of protocol to the payload of user medical data down to the LLC frame adds protocol overheads to the user data. The size of each overhead added to the user data is shown in Table 5.2 [5.14, 5.15]. The size of the TCP and IP header are assumed to be 24 bytes and 20 bytes respectively, as implemented by the Vodafone GPRS network used in the research. The SNDCP layer adds a 4-byte header to each IP packet. The LLC frame adds a 7-byte overhead of frame header and frame check sequence.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>24</td>
</tr>
<tr>
<td>IP</td>
<td>20</td>
</tr>
<tr>
<td>SNDCP</td>
<td>4</td>
</tr>
<tr>
<td>LLC</td>
<td>7</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>55</strong></td>
</tr>
</tbody>
</table>

Table 5.2 Size of overhead added at each protocol layer (in bytes)

Each LLC frame is further segmented into Radio Link Control (RLC)/Medium Access Control (MAC) blocks at the RLC/MAC layer. Each block is transmitted in four bursts of consecutive TDMA frames at the physical layer. The bursts are then received by a Base Station System (BSS), which later performs the necessary protocols implementation to transfer data packets to the final destination address/terminal. The details of the protocols implementation from the BSS to the final terminal are beyond the scope of the thesis and are available in various documents [5.4, 5.5, 5.13, 5.14].

86
The format of a GPRS RLC/MAC block for data transfer is shown in Figure 5.8 [5.16]. The RLC/MAC block consists of a MAC header and an RLC data block. The RLC data block consists of an RLC header, an RLC data unit and spare bits. The size of each block depends on the GPRS channel coding schemes (CS1, CS2, CS3, or CS4) for the transmission [5.15-5.17], as shown in Table 5.3. The RLC/MAC blocks are then encoded and arranged into GPRS time slots for transmission over the radio interface.

<table>
<thead>
<tr>
<th>Coding Scheme</th>
<th>RLC/MAC Block</th>
<th>RLC/MAC Overhead</th>
<th>RLC Data Block Size</th>
<th>RLC Data Unit</th>
<th>Spare Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS1</td>
<td>181</td>
<td>21</td>
<td>176</td>
<td>160</td>
<td>0</td>
</tr>
<tr>
<td>CS2</td>
<td>268</td>
<td>28</td>
<td>263</td>
<td>240</td>
<td>7</td>
</tr>
<tr>
<td>CS3</td>
<td>312</td>
<td>24</td>
<td>307</td>
<td>288</td>
<td>3</td>
</tr>
<tr>
<td>CS4</td>
<td>428</td>
<td>28</td>
<td>423</td>
<td>400</td>
<td>7</td>
</tr>
</tbody>
</table>

Table 5.3  Size of GPRS RLC/MAC block for data transfer (in bits)

The minimal number of RLC/MAC blocks required for the multi-channel medical data transmission can be determined based on the information from Table 5.2 and Table 5.3. The application program on the mobile phone is set to transmit 1036 bytes of medical data per IP packet, which is equivalent to 28 set of samples (37 bytes per sample) of multiplexed data from four analogue input channels.
This payload size is chosen based on the average user data size for GPRS transmission, to minimise the overheads from the intermediate radio access protocols. The choice of IP packet size is a balance between the overheads added by the protocols for each IP packet and the limited buffer size of the mobile phone processor. If the size of the IP packet is set to be too large, then the limited buffer on the mobile phone may not be able to accommodate the payload size of the IP packet, i.e. the mobile phone processor may not be able to handle a large amount of data before it can transmit and clear the buffer. If the size of the IP packet is too small, then more overheads are added for the total number of IP packets required for transmission.

The size of the IP packets also has a dominant effect on the performance of the GPRS network transmission. The GPRS throughput is directly related to the size of IP packets set for the transmission. A small IP packet, in the region of 100 bytes or less, has a substantially lower throughput compared with larger IP packet size [5.18].

<table>
<thead>
<tr>
<th>Coding Scheme</th>
<th>CS1</th>
<th>CS2</th>
<th>CS3</th>
<th>CS4</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Data (bytes)</td>
<td>1036</td>
<td>1036</td>
<td>1036</td>
<td>1036</td>
</tr>
<tr>
<td>Total Data (bytes)</td>
<td>1091</td>
<td>1091</td>
<td>1091</td>
<td>1091</td>
</tr>
<tr>
<td>RLC Data Unit (bytes)</td>
<td>20</td>
<td>30</td>
<td>36</td>
<td>50</td>
</tr>
<tr>
<td>RLC/MAC Blocks</td>
<td>55</td>
<td>37</td>
<td>31</td>
<td>22</td>
</tr>
</tbody>
</table>

Table 5.4 Minimal number of RLC/MAC blocks required for 1036 bytes of medical data per IP packet

Table 5.4 shows the minimal number of RLC/MAC blocks required for the 1036 bytes of medical data per IP packet. The minimal number of radio blocks needed is calculated from Equation 5.1, rounded up to the nearest whole block. The amount of overheads for each RLC/MAC block depends on the GPRS channel coding schemes.
The Vodafone GPRS network, and most handsets, only supports CS1 and CS2 coding schemes. The high incidence of errors when CS3 and CS4 coding schemes are used prohibits their use under most circumstances [5.15]. The percentage of the protocol overheads with respect to the user medical data for the mobile telemedicine system can be estimated from Equation 5.2 and Equation 5.3.

\[
\text{Number of RLC Blocks} = \frac{\text{Total User Data (bytes) + Protocol Overheads (bytes)}}{\text{RLC Data Unit (bytes)}}
\]

Equation 5.1 Number of RLC Blocks

\[
\text{Total Transmission Overheads} = \text{Protocol Overheads (Bits)} + \left( \text{Number of RLC Blocks} \times \left\lceil \frac{\text{RLC/MAC Overhead (Bits)}}{\text{Bits}} \right\rceil \right)
\]

Equation 5.2 Total Transmission Overheads

\[
\text{Overhead Percentage} = \frac{\text{Total Transmission Overheads (Bits)}}{\text{Total User Data (Bits)}} \times 100\%
\]

Equation 5.3 Overhead Percentage

The example calculations are done based on CS1 and CS2 coding schemes since these are adopted by the Vodafone GPRS network. Both coding schemes yield a protocol overheads percentage of less than 20% for every IP packet sent in the mobile telemedicine system. Example overhead percentage calculation for the CS1 coding scheme:
CHAPTER FIVE

Total Transmission Overheads ( Bits ) \[= (55 \times 8) + (55 \times 21)\]
\[= 1595\]

Overhead Percentage \[= \frac{1595}{(1036 \times 8)} \times 100\%\]
\[= 19.24\%\]

Example overhead percentage calculation for the CS2 coding scheme:

Total Transmission Overheads (Bits) \[= (55 \times 8) + (37 \times 28)\]
\[= 1476\]

Overhead Percentage \[= \frac{1476}{(1036 \times 8)} \times 100\%\]
\[= 17.80\%\]

5.3.2 PDP Context Activation

GPRS connectivity is provided by a 'context', which is specified by the Packet Data Protocol (PDP). PDP is a network protocol, which is used by external Packet Data Network (PDN) to communicate with GPRS networks. The Internet is a PDN with IP as its PDP type and is the first type of PDN to be supported by GPRS. Table 5.5 lists the PDP types that are supported by a GPRS network [5.15].
A GPRS attach procedure is required before a mobile terminal can receive any GPRS services. During a GPRS attach procedure, a user is registered with an SGSN of the GPRS network by checking whether the user is authorised, copying the user profile from the Home Location Register (HLR) to the SGSN, and assigning a Packet Temporary Mobile Subscriber Identity (P-TMSI) to the user [5.4]. This attach procedure can be performed immediately after the mobile terminal has been switched on or at a later time as the user decides to use the GPRS services.

The simplified GPRS attach procedure is described as follows [5.8]:

**Step A:** The Mobile Station (MS) initiates the attach procedure by sending the International Mobile Subscriber Identity (IMSI) together with other parameters to the SGSN. The IMSI uniquely identifies the MS, which is the only key to search the MS record in the HLR.

**Step B:** The SGSN accesses the HLR using the IMSI to obtain the subscriber data of the MS (including the PDP contexts of all subscribed services). Then it informs the MS that the attach procedure is successful.

The implementation of the GPRS attach procedure in the mobile telemedicine system is done by configuring the mobile phone to stay registered with the SGSN, then initiating the attachment procedure immediately after the mobile phone is switched on. The configuration steps vary according to mobile phone models but the important part is to set the GPRS connection settings on the

<table>
<thead>
<tr>
<th>PDP Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>X25</td>
<td>ITU-T/CCITT X.25 Layer 3</td>
</tr>
<tr>
<td>OSPIH</td>
<td>Internet Hosted Octet Stream Protocol</td>
</tr>
<tr>
<td>PPP</td>
<td>Point to Point Protocol</td>
</tr>
</tbody>
</table>

Table 5.5 List of GPRS PDP types
mobile phone to use the GPRS network whenever the GPRS network is available. The availability of the GPRS network is normally indicated on the service level indicator on a mobile phone display once the settings have been activated.

For example, the sequence of configuration steps for Nokia 6600 mobile phone is as given below:

1. Select the Main Menu option on the mobile phone
2. Open the Tools folder on the Main Menu
3. Open the Settings option on the Tools menu
4. Open the Connection option on the Settings menu
5. Open the GPRS option on the Connection menu
6. Select “When available” option of the GPRS Connection on the GPRS menu
7. Exit to return to normal display

A GPRS Detach procedure performs the opposite of the attachment procedure and informs the network that a user does not want to access the SGSN-based services any longer. The detach procedure also allows the network to inform a mobile terminal that it no longer has access to the SGSN-based services.

When a mobile terminal needs to exchange data packets with external PDN after a successful GPRS attach, it initiates a procedure called PDP Context Activation. The procedure specifies the application-layer PDP and routing information for the communication session [5.19]. The mobile terminal will then be allocated with resources from the radio access network and GGSN allocates a dynamic PDP address for the mobile terminal. For example, if the mobile terminal needs to use IP-based services, the GGSN allocates a dynamic private IP address for the mobile. A successful PDP Context Activation means a packet-switched data connection between the mobile terminal and the GGSN has been established and data traffic is being delivered. In the case of an uplink transmission, as is the case with
transmission from a patient to the server at a hospital in the mobile telemedicine system, the data traffic flow is from radio access network to the SGSN, which then tunnels the data to a GGSN, and the GGSN routes the data to the correct external packet data network based on destination IP address of the hospital server.

The PDP Context Activation procedure activated by the Mobile Station (MS) is summarised as follows for IP-based services [5.4, 5.8]:

**Step 1)** The MS sends the *Activate PDP Context Request* message to the SGSN. The SGSN already has the PDP context information of the MS during the attach procedure.

**Step 2)** The SGSN sends the *Create PDP Context Request* message to the GGSN. The message contains the necessary information to create the PDP context of the MS in the GGSN. The activation also creates a logical link between the SGSN and the GGSN for this PDP context. The GGSN assigns an IP address to the MS.

**Step 3)** The GGSN replies to the SGSN with the *Create PDP Context Response* message. This message provides the GGSN address and the allocated IP address. The SGSN sends the IP address to the MS through the *Activate PDP Context Accept* message.

The state of the GPRS attachment and the PDP Context Activation can be verified by communicating with the mobile phone via AT Commands. A terminal application, for example a HyperTerminal program, is used to set up the serial interface with the mobile phone. Since the Nokia 6600 mobile phone used in the research has Bluetooth connectivity, a Bluetooth link has been set up to communicate with the mobile phone. The AT Commands are sent to the mobile phone to verify its current GPRS status.
The PDP Context Activation procedures are transparent to the user. The procedures are done by the mobile phone as soon as the patient grants the mobile telemedicine system the permission to use the GPRS network to transmit the medical data on the mobile phone program. Further details on the mobile phone program will be explained in the following section 5.3.2.

5.4 Mobile Phone Application Software

The development of the mobile phone application program for the mobile telemedicine system involves various software platforms. The development platform consists of specifications from a Java platform to the Symbian operating system for the mobile phone. This section of the thesis highlights and explains various software requirements and standards used in producing the mobile phone application program for the mobile telemedicine system.

The most important foundation for the development is the background on Java 2 Platform by Sun Microsystems. The development of the mobile phone application program is based on standards and architecture defined by the Java 2 Platform, supported by the compatibilities with the mobile phone’s operating system. The Sun Java platform architecture consists of three editions, the Java 2 Platform Standard Edition (J2SE), the Java 2 Platform Enterprise Edition (J2EE) and the Java 2 Platform Micro Edition (J2ME) [5.21-5.25]. The J2SE platform is mainly targeted for standard desktop and workstation applications with a complete Java functionality and components. These applications include web-based and standalone desktop applications. The J2EE platform on the other hand is targeted for the server side of a network and contains all the functionality for extensive server systems operations. The J2ME platform is the one adopted for the mobile telemedicine system, which targets development for small and memory constrained portable devices, such as PDAs and mobile phones.
The J2ME platform provides a common base for application developments on small portable devices with limited memory and processing power. Therefore, the J2ME only supports limited Java functionality meant for simple and lightweight application in terms of processing power. The J2ME architecture consists of two main building blocks: configuration and profile [5.26]. The overview of J2ME building blocks is illustrated in Figure 5.9.

The J2ME configuration defines a minimum platform required for a group of portable devices and comprises a Java virtual machine, Java language features and minimum Java-class libraries. Currently, there are two J2ME configurations that have been defined, which are Connected Limited Device Configuration (CLDC) and Connected Device Configuration (CDC) [5.27, 5.28]. The CLDC defines a standard for small and resource constrained devices such as mobile phones. The CDC on the other hand defines a platform for devices that have larger memory and processing power than CLDC devices, such as PDA. Both configurations consist of Java language, Java virtual machine and Java libraries but the main different is the amount of Java functionality and classes supported by each configuration.

![Figure 5.9 The overview of J2ME building blocks](image)
The J2ME profile extends a configuration and addresses the needs of a specific device group. Consumer devices for example, have different functionalities and requirements. Therefore, the J2ME platform provides device specific profiles such as a profile for a mobile phone and a profile for a PDA. The CLDC has only one profile that has been defined so far, known as Mobile Information Device Profile (MIDP) [5.29]. The MIDP specification defines the architecture and API needed for mobile information devices, which share the common features such as limited memory, small screen size and limited network bandwidth. The CDC also has only one profile that has been defined at this stage, known as Personal profile.

Descriptions and specifications for Java platform is given by standard known as Java Specification Requests (JSRs). JSRs are the actual descriptions of proposed and final specifications for all the three different editions of Java platforms. There are about 70 JSRs that are currently available for the J2ME platform [5.30] and only the related JSRs for the mobile phone application development will be briefly discussed. The review and approval process of JSRs are continuous and at any one time there are numerous JSRs moving through the process. A JSR is referred to with a number and the summary of the specification. For example, JSR 30 is known as J2ME Connected, Limited Device Configuration and defines a standard configuration of the J2ME platform for small, resource-limited, connected devices. Some of the related JSRs are summarised in Table 5.6 [5.26, 5.27, 5.29, 5.31-5.34].

The MIDP provides the essential information and standards for software program development for mobile phones. MIDP defines a programming API for mobile phone applications on MIDP compliant mobile phone models. A MIDlet is an application development based on the MIDP profile that runs on a compatible mobile phone.
<table>
<thead>
<tr>
<th>Reference</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JSR 30</td>
<td>J2ME Connected, Limited Device Configuration</td>
<td>Define a standard platform configuration of the J2ME for small, resource-limited, connected devices</td>
</tr>
<tr>
<td>JSR 37</td>
<td>Mobile Information Device Profile for the J2ME Platform</td>
<td>Define a profile that will extend and enhance the JSR 30, enabling application development for mobile information appliances and voice communication devices</td>
</tr>
<tr>
<td>JSR 68</td>
<td>J2ME Platform Specification</td>
<td>Define the next major revision of the J2ME</td>
</tr>
<tr>
<td>JSR 82</td>
<td>Java APIs for Bluetooth</td>
<td>Standardizes a set of Java APIs to allow Java-enabled devices to integrate into a Bluetooth environment.</td>
</tr>
<tr>
<td>JSR 118</td>
<td>Mobile Information Device Profile 2.0</td>
<td>Define a profile that will extend and enhance the JSR 37</td>
</tr>
<tr>
<td>JSR 139</td>
<td>Connected Limited Device Configuration 1.1</td>
<td>Define a revised version of the CLDC 1.0</td>
</tr>
</tbody>
</table>

Table 5.6  JSRs related to the m-Health system

Some models of data-enabled mobile phones running operating systems. Two of the most dominant mobile phone operating systems are Symbian OS by Symbian and Windows Mobile 2003 for smart phones by Microsoft [5.35, 5.36]. On top of the operating system, there is a development platform that is widely available on mobile phones, known as the Series 60 platform. Series 60 is a platform for smart phones built on the Symbian operating system [5.37]. The Series 60 platform offers common user interface components, development tools, a suite of ready-made applications and support for large colour screen for new applications development. The Series 60 platform architecture is made of three functionality layers, namely Applications, User Interface and Enablers layers, as shown in Figure 5.10 [5.37].
The Enablers layer is the layer responsible for application execution and supports either Symbian C++ or Java virtual machine environment. The Java application environment in the Series 60 platform is utilised for the research that supports, among other things, the MIDP 2.0 profile (JSR 118), the CLDC (JSR 30) and the Bluetooth API (JSR 82) for J2ME applications. The Nokia 6600 mobile phone model used for the research is based on the Series 60 platform, as given by the specifications listed in Chapter 4, section 4.3.1. Some other Series 60 mobile phone models are Nokia 6630, Panasonic X700, Siemens SX1. An up to date list of Series 60 mobile phones is available from Series 60 platform official website [5.38].
5.4.1 J2ME Development Process

The next step in mobile phone application development is to set up a working development environment to produce J2ME MIDlets for the mobile phone used in the research. An Eclipse development platform offers integrated development environments that can be used to create applications for web sites, embedded Java programs and C++ programs [5.40]. The Eclipse platform is therefore used to produce a Java-based, J2ME application for the mobile phone program in the m-Health system. The uniqueness of building J2ME MIDlets is that the source code is compiled on the desktop development environment but the MIDlet itself runs on a mobile phone. The application management software in a mobile phone controls the installation and execution of the MIDlet. The MIDlet is deployed in a package, known as a MIDlet suite, which contains the Java Archive (JAR) file and the Application Descriptor file. The MIDlet suite is transferred to the mobile phone via a serial cable, an infra-red or a Bluetooth link. The details of the development process in producing the J2ME MIDlets for the m-Health system are provided in Appendix F.

5.4.2 Telemedicine MIDlet Suite

The J2ME MIDlet suite deployed for the mobile telemedicine system is called “Telemedicine MIDlet” suite that represents the patient interface program on a mobile phone for the GPRS network. The flow and the functions performed by the patient interface mobile phone program will be explained in the next chapter. This section explains some of the detail of the Java source code for the J2ME Telemedicine MIDlet suite which includes the MIDlet life cycle, the Bluetooth link connection with the telemedicine processor module, the socket connection establishment with the hospital server and the subsequent IP packet formation of the medical data for the GPRS network transmission.
In general, a MIDlet goes through a few states depending on the instructions called by the application manager on a device, as shown in Figure 5.11 [5.26]. When a MIDlet is about to run, the MIDlet’s constructor is prepared and invoke the MIDlet into a \textit{Paused} state. The MIDlet then enters an \textit{Active} state when the application manager calls the \textit{startApp()} function and executes the instructed operations. The execution of the MIDlet may be suspended by the application manager by calling the \textit{pauseApp()} function and returns the MIDlet back in the \textit{Paused} state. The MIDlet execution can be terminated at any time by the application manager with the \textit{destroyApp()} function.

The Bluetooth link connection establishment with the telemedicine processor module utilises the Serial Port Profile defined in Bluetooth system profiles. The J2ME optional package defined by the JSR 82, the Java APIs for Bluetooth, provides a common API for Bluetooth application development in mobile phones. The Java APIs in the mobile phone perform the initialisation of the Bluetooth stack, the discovery of devices or services that are in proximity, the connection operations (open, close, wait or initiate connections) and the input/output operations.
Figure 5.12 illustrates the relationship between the Java APIs for Bluetooth and the J2ME platform for Series 60 mobile phones, i.e. with MIDP 2.0 and CLDC 1.0 stack. The upper layers are the MIDP application (MIDlet) for the mobile telemedicine system on the top, followed by the MIDP 2.0 profile stack, the CLDC 1.0 configuration and the Java APIs JSR 82 optional package. The two lower layers are the Bluetooth stack and the hardware with the operating system, which in the Nokia 6600 model, is the Symbian OS version 7.0 operating system. The Bluetooth part of the MIDlet is responsible for discovering a remote device and services available from the remote device. The MIDlet retrieves the name of the telemedicine processor module and authenticates the module based on protocols highlighted in section 5.1.3.

The socket connection establishment with the hospital server allows the patient to request for connection via the GPRS network. The socket connection code specifies the IP address of the hospital server together with the assigned port number for the mobile telemedicine applications. An example of socket connection code is given by:
When the Telemedicine MIDlet attempts to establish a network connection, the MIDlet asks the patient for permission to use the GPRS network and asks the patient to specify the access point name for the GPRS network gateway. The access point name is used by the mobile phone to perform the GPRS PDP Context Activation procedures, as highlighted in section 5.2.2.

The Telemedicine MIDlet also handles the packet payload formation of the medical data for the GPRS IP packet transmission. The received data from the Bluetooth serial link is placed in a temporary transmit buffer until the buffer size reaches a pre-set limit of the IP packet size. The Telemedicine MIDlet then places the payload onto the socket connection output stream for the IP packet formation. The packet payload format of the medical data is the same as the multiplexed output format of the telemedicine processor module, as shown in Figure 5.3. The mobile phone then encapsulates the medical data to the appropriate GPRS and IP protocols, as explained in section 5.2.1.

5.4.3 Mobile Phone Limitation and Known Issues

The development of the mobile phone application for the m-Health system based on J2ME specifications allows the application to be compatible with various mobile phone models. The use of a multi-platform development environment for the J2ME MIDlets makes it possible to deploy the Telemedicine MIDlet Suite to other mobile phones with different CLDC and MIDP versions. In additions, the use of the Series 60 platform mobile phone demonstrates the compatibility of the application with world's leading mobile phone manufacturers such as LG Electronics, Nokia, Panasonic, Siemens and Samsung, who together sell millions of Series 60 platform devices [5.37].
Deployment of the Telemedicine MIDlet Suite into mobile phones which are not in the Series 60 platform is also possible, provided the mobile phone supports the crucial Java API (JSR 82) functionality in order to receive data via the Bluetooth link from the telemedicine processor module. A mobile phone with Bluetooth connectivity and which supports J2ME functionality is still not capable of receiving the medical data using the Telemedicine MIDlet Suite, if the JSR 82 functionality is not supported by the mobile phone. This means the mobile phone is able to connect via the Bluetooth link but is not able to pass the data received to the rest of the MIDlet application because the J2ME platform and the Bluetooth connectivity cannot communicate with each other. Some examples of incompatible mobile phones are Siemens SL65 and most of Sony Ericsson models except the P900/P910 models, which support the JSR 82 API [5.45, 5.46].

The Nokia 6600 mobile phone adopted for the research is among the first commercial mobile phone models implemented with the Series 60 platform and was first released in 2003. Due to being first implemented with the MIDP 2.0 specifications, the mobile software is prone to unforeseen issues and incompatibility within the platform itself. This is well noted by the manufacturer in documentation released to present ongoing known issues related to the MIDP 2.0 implementation. Developers and researchers who are working on the Series 60 software gradually discover the known issues in the first software version. One of the critical issues that specifically affect the implementation of the mobile phone application is the use of Bluetooth Serial Port in MIDlets [5.47]. One of the functions in the Bluetooth Serial Port profile is reported as potentially corrupting the execution of the MIDlet. The “reading incoming data stream” function does not work properly and causes the Midlet to receive end-of-file characters and corrupts the subsequent incoming stream to the mobile phone. This issue explains the instability of the patient’s mobile phone application program that may cause the incoming data to be unrecognisable and sometimes cause the program to crash.
5.5 Summary

The software development in this chapter focuses on the development of the m-Health processor module and the J2ME MIDlet suite application for the mobile phone interface. The main software program for the m-Health processor module is developed based on low-level assembly language. The Bluetooth module integrated in the implementation has an embedded Bluetooth stack, thus reducing the processing load on the processor host.

The GPRS and IP protocols adopted in the system implementation are based on the standard specifications operated by the Vodafone GPRS network in the UK. The network protocols overhead added to the user medical data during the GPRS network transmission has been estimated to be less than 20% of the entire data traffic.

The J2ME development platform which has been successfully set up for building the J2ME MIDlet suite application offers a common base for more application developments on small portable devices with limited memory and processing power.

The compatibility issue of the J2ME MIDlet suite application for the mobile phone has been considered during development stage and the adoption of Series 60 platform applications allows the J2ME Telemedicine MIDlet suite to be compatible with Series 60 platform devices. The current known issues discovered in the development of Bluetooth-enabled MIDlets in Series 60 platform devices are crucial in devising future research on the m-Health system.
5.6 References


[5.4] 3GPP, "General Packet Radio Service (GPRS); Service Description; Stage 2 (Release 6)," 3GPP TS 23.060, December 2003.

[5.5] ETSI, "Digital cellular telecommunications system (Phase 2+); General Packet Radio Service (GPRS); Service description; Stage 1 (Release 1998)," GSM 02.60 version 7.5.0, July 2000.


[5.13] 3GPP, "Digital cellular telecommunications system (Phase 2+); Mobile Station (MS) - Serving GPRS Support Node (SGSN); Subnetwork Dependent Convergence Protocol (SNDCP)," 3GPP TS 44.065, September 2004.


[5.16] 3GPP, "Digital cellular telecommunications system (Phase 2+); General Packet Radio Service (GPRS); Mobile Station (MS) - Base Station System (BSS) interface; Radio Link Control/ Medium Access Control (RLC/MAC) protocol," 3GPP TS 44.060, February 2005.


[5.20] 3GPP, "Digital cellular telecommunications system (Phase 2+); AT Command set for GSM Mobile Equipment (ME)," TS 07.07, March 2003.


6.1 Overview of a Client-Server Model

A client-server interaction is a fundamental concept in peer-to-peer networking systems and forms a basis for most computer communication. A server is a program running on a remote machine that provides services to clients and waits for incoming requests from clients before responding accordingly. A client is a program running on a local machine requesting service from a server; it is started by a user and terminates upon completion of the service. In the context of the m-Health system, a patient from a given location becomes a client when accessing the telemedicine server at a hospital to start a session.

Client-server application programs rely on a Transmission Control Protocol/Internetworking Protocol (TCP/IP) suite to provide basic transport services. The TCP/IP protocol consists of five layers, namely physical, data link, network, transport, and application. The application layer can be compared with the combination of session, presentation, and application layers of an OSI model [6.1]. At the transport layer, TCP/IP defines two
protocols: TCP and User Datagram Protocol (UDP). The TCP transport protocol provides a more reliable end-to-end packet delivery than UDP. The TCP is based on connection-oriented interactions, which means that a connection must be established between both ends of a transmission before either side starts to transmit any data. With the view of its level of reliability supported for its underlying system, TCP is adopted for the transport protocol in the telemedicine client-server applications.

The client-server model of the m-Health system is shown in Figure 6.1. The patient unit is the client side connecting to the hospital server via the GPRS network. The hospital server consists of a Telemedicine Application Server, which controls the telemedicine connections, and a MySQL Telemedicine Database Server which store the medical data processed by the application server. The application server is also a client to the database server at the same time. The transmitted medical data are available to clinicians as and when required by accessing the database server as a client. These may be achieved by connecting from the hospital LAN or from the GPRS network while on the move.
Figure 6.1 Client-server model of the m-Health system
6.1.1 Development Environment

The client-server application programs for the m-Health system have been developed using a Microsoft Visual Basic 6.0 environment. This is capable of producing client-server programs by using a control property known as Windows Sockets (Winsock), which is part of the Visual Basic toolbox called ActiveX Controls. Winsock defines a network programming interface for Microsoft Windows and allows client-server applications to communicate using both TCP and UDP transport protocols.

Besides the various graphic tools available in Visual Basic 6.0 for producing graphical user-interface programs, this environment is also capable of providing a serial communication control to allow the client program to read input data from the telemedicine processor module via a Bluetooth link. Therefore, the Winsock and communication control properties are utilised in producing client-server application programs for the m-Health systems. A client program is used by a patient to read medical data from the hardware and transmit the data to a server program at the hospital side. A browser program is also produced for clinicians to view the data from a telemedicine database as and when required. These application programs, which allow interaction with a database, are possible with the database technology supported by Visual Basic, known as ActiveX Data Object (ADO). The ADO database technology allows SQL query to be embedded into the Visual Basic program and to access a database via an Open Database Connectivity (ODBC) database Application Programming Interface (API) [6.2]. In view of its versatility in the application development, Visual Basic was chosen as the main platform in developing application programs for the m-Health system.

Most of client-server application programs are commonly developed on a web-based platform. A web browser (for example Internet Explorer), is used to view an immense volume of pages of information available on the web. A
web server stores much of the information in documents using a HyperText Markup Language (HTML) which is displayed by the web browser as a web page. Additional modules and approaches are required in order to integrate database applications into the web-based environment. Some of the current approaches in integrating databases in web-based applications are [6.2, 6.3]:

- Use of scripting languages such as VBScript, JavaScript, Jscript, Perl (Practical Extraction and Report Language) and PHP (PHP: Hypertext Preprocessor)
- Common Gateway Interface (CGI)
- Web server extensions such as Netscape API and server APIs
- Java platform for web environment (Servlets, JavaServer Pages)
- Microsoft’s web solution platform such as Active Server Pages (ASP) and .NET

Details on each of the approaches are beyond the scope of this thesis and are available in [6.2, 6.4]. The important information to highlight is the additional modules and components needed to implement a web-based database application as compared to application programs developed for the m-Health system.

The m-Health application programs based on Visual Basic environment have the ability to interface to a database without the need for a web server. The graphical user interface is produced to simplify and to improve database access. The CGI approach for database integration in web-based application requires the web server to convert data from or to an HTML document for every request from client or response given by the database server [6.2, 6.5]. This process of executing CGI scripts is time consuming as they are loaded and unloaded each time. In addition, the database server has to perform the same log-on and log-out procedure for every query with the CGI approach, even for subsequent queries submitted by the same user.
6.2 MySQL Telemedicine Database Server

A database is a single, large repository of data that can be simultaneously accessed by many users. It is defined as a shared collection and description of logically related data designed to meet the information needs of an organization [3.39]. A Database Management System (DBMS) is a software system that allows interaction between the users' application programs and the database. The DBMS enables users to define, create, maintain, and control access to the database. Users are not capable of accessing data files directly and have to communicate with the DBMS. A query language is used to produce statements that are processed by the DBMS to access the underlying data files. The most common query language for relational DBMS is the Structured Query Language (SQL) which consists of statements to insert, update, delete, query and protect data [3.40].

A relational database model is a dominant data processing model in use today. In the relational model, all data is logically structured within relations (tables) [3.39]. The model is based on the mathematical concept of a relation, which is represented as a table. Some terminology used for relational database model is listed in Table 6.1.

<table>
<thead>
<tr>
<th>Terminology</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relation</td>
<td>A table with columns and rows</td>
</tr>
<tr>
<td>Attribute</td>
<td>A column name of a relation</td>
</tr>
<tr>
<td>Domain</td>
<td>Set of allowable values for one or more attributes</td>
</tr>
<tr>
<td>Tuple</td>
<td>A row of a relation</td>
</tr>
</tbody>
</table>

Table 6.1 Terminology of relational database model

The terminology of relational model may also be referred to using alternative terms. The relation may be referred to as a table, the attribute may be referred to as a field and the tuple may be referred to as a record. The relational model and its alternative terminology are used in creating a
database for the m-Health system. The database is mainly used to store patient information and multi-channel medical data transmitted during a telemedicine session. Details on the telemedicine database implemented for the system are presented in section 6.2.

MySQL is one of the DBMS [3.41]. MySQL is capable of controlling several databases at a time and decides how the data are stored, sorted and retrieved. MySQL is also part of a client-server system and provides a multi-threaded, multi-user, and robust database server. The MySQL database server is an open source database and is available for free for non-commercial use. A suitable client program allows remote access to a database controlled by the MySQL server and therefore is adopted for the creation of the telemedicine database server.

A MySQL relational database was set up to store data related to the m-Health system. MySQL is capable of controlling several databases at a time and is part of a client-server system. MySQL also provides a multi-threaded, multi-user, robust database server. The MySQL database can be accessed from various application platforms via the standard ODBC driver [6.6]. The driver is also known as MySQL Connector/ODBC (MyODBC), which allows connection to the MySQL database server using the ODBC database API on all Microsoft Windows and most UNIX platforms. A user interface program is designed to perform data access through the ODBC. Figure 6.2 summarises the database connectivity concept. The user interface programs access the telemedicine database via the ODBC and perform the relevant SQL commands to upload and to retrieve data from the database.
The MySQL database server consists of two main databases, namely mysql database and telemedicine database. The mysql database is the default database that holds the user account and administrative information of the database server, such as the usernames, passwords, host location and user privileges. The administrative privilege is for the system administrator, which has a full access to the MySQL database. User privileges vary according to the level of authority or level of usage authorised by the system administrator. For example, a clinician may be granted with privileges to add, update, delete and retrieve data into the telemedicine database. A patient may only be granted with a limited access to retrieve data from a table directly related to the patient’s user information. The telemedicine database stores patient information as well as various data generated during telemedicine sessions.

6.2.1 Telemedicine Database Structure

The telemedicine database was created to support the m-Health system by providing storage and archive capacity to allow clinicians to view the transmitted data as and when required. The traditional approach to relational
database design is based on the logical organization of the data into a number of tables [6.7]. The main process in a database design is to determine the tables and fields entries. The process determines which data must be stored, the structure of the data, and defines a primary key for each table [6.8]. The telemedicine database for the m-Health system consists of three tables, namely Patient Information table, Sessions table and Signal Sets table. The Patient Information table provides patient information such as a unique patient identification number (patient ID), name, address, age and gender. The Sessions table provides the information on the telemedicine session created by a patient and consists of session number, patient ID, time and date when a session is created. The Signal Sets table stores the medical data transmitted during the telemedicine session and consists of session number, patient ID, and data from four different channels. Table 6.2 to Table 6.4 list all the tables in the telemedicine database, together with the corresponding fields for each table. Figure 6.3 shows the relationships among tables in the telemedicine database.

Figure 6.3  Relationships between the telemedicine database tables
### Table 6.2 Description of the Patient Information table

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Patient ID</td>
<td>Varchar</td>
<td>A unique patient identification number assigned by the healthcare provider</td>
</tr>
<tr>
<td>Name</td>
<td>Varchar</td>
<td>A full name of the patient</td>
</tr>
<tr>
<td>Address</td>
<td>Varchar</td>
<td>A full corresponding address of the patient</td>
</tr>
<tr>
<td>Age</td>
<td>Int</td>
<td>Age of the patient</td>
</tr>
<tr>
<td>Gender</td>
<td>Char</td>
<td>Gender of the patient: M (male) or F (female)</td>
</tr>
</tbody>
</table>

### Table 6.3 Description of the Sessions table

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session No</td>
<td>Int</td>
<td>A unique telemedicine session number assigned by the server for a given patient</td>
</tr>
<tr>
<td>Patient ID</td>
<td>Varchar</td>
<td>A unique patient identification number assigned by the healthcare provider</td>
</tr>
<tr>
<td>Time</td>
<td>Text</td>
<td>Time recorded when a telemedicine session is completed (Hours:Minutes:Seconds format in 24 hours clock display)</td>
</tr>
<tr>
<td>Date</td>
<td>Text</td>
<td>Date on which the telemedicine session is completed (Day:Month:Year format)</td>
</tr>
</tbody>
</table>

### Table 6.4 Description of the Signal Sets table

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session No</td>
<td>Int</td>
<td>A unique telemedicine session number assigned by the server for a given patient</td>
</tr>
<tr>
<td>Patient ID</td>
<td>Varchar</td>
<td>A unique patient identification number assigned by the healthcare provider</td>
</tr>
<tr>
<td>Channel 1</td>
<td>Longtext</td>
<td>The digitised medical data from channel one of the telemedicine processor module (stored in Hex format e.g.&quot;03FF&quot;)</td>
</tr>
<tr>
<td>Channel 2</td>
<td>Longtext</td>
<td>The digitised medical data from channel two of the telemedicine processor module (stored in Hex format e.g.&quot;03FF&quot;)</td>
</tr>
<tr>
<td>Channel 3</td>
<td>Longtext</td>
<td>The digitised medical data from channel three of the telemedicine processor module (stored in Hex format e.g.&quot;03FF&quot;)</td>
</tr>
<tr>
<td>Channel 4</td>
<td>Longtext</td>
<td>The digitised medical data from channel four of the telemedicine processor module (stored in Hex format e.g.&quot;03FF&quot;)</td>
</tr>
</tbody>
</table>
A primary key in a database is a data element that uniquely identifies a row of data elements within a table. The patient ID is a primary key in the Patient Information table, which means that there can be no duplication of patient ID and all patients have their own unique patient IDs. A combination of session number and patient ID is the primary key for both Sessions table and Signal Sets table. There can be many sessions for the same patient ID in the Sessions table but there is no duplication of session number for the same patient ID in the Sessions table. The exact conditions also apply to the Signal Sets table to avoid duplicate or redundancy of data stored in the database.

A foreign key is a primary key of another table used to draw a relationship between two tables. The primary key in the Patient Information table is the patient ID and each row in the Sessions table contains the patient ID of the patient who created the session. Therefore the patient ID in the Sessions table is a foreign key. That is, the patient ID in the Sessions table is the primary key of the Patient Information table.

The Signal Sets table is created to store the multi-channel data from the telemedicine session instead of using the Sessions table for storing the multi-channel data. Although both tables have the same session number and patient ID fields, a separate table for the multi-channel data is essential in order to minimise the fields and entries in the Sessions table, since it is retrieved extensively in the Telemedicine Application Server as well as the Doctor Browser application program for the m-Health system. Additional fields can be created in the Patient Information table should more patient details information are required in the database.
6.3 Telemedicine Application Server

In general, the main function of the Telemedicine Application Server is to handle the telemedicine session in m-Health system. The server controls the client connection request from a patient user interface program and responds accordingly to accept the request and establish a secure point-to-point link with the patient. The Telemedicine Application Server is also responsible for communication with the MySQL Telemedicine Database Server and performs the appropriate query process required during a telemedicine session. These include storing information such as the time and date of a session besides creating an entry in the database to store the multi-channel medical data transmitted to the hospital. The Application Server also provides a graphical user-interface for healthcare providers to monitor and to administer the m-Health system. Some of the administrative functions include adding a new patient to the system, updating or deleting existing patient information and monitoring the number of sessions performed by a patient.

As described in the previous development environment section, the Telemedicine Application Server is developed based on Visual Basic programming with the ADO database technology, which allows interaction with the MySQL Telemedicine Database via an ODBC database API. The client-server connection is implemented with the Winsock network programming interface to allow communication using the TCP transport protocol. Figure 6.4 summarises the architecture of the Telemedicine Application Server.
6.3.1 Server Protocols

The protocols implemented in the Telemedicine Application Server can be categorised into three distinct routines, which are the packet extraction, telemedicine session creation and data updating process into the telemedicine database. The server first requests the user login and password besides the host name for the MySQL Telemedicine Database Server. The packet extraction process begins when the server decodes the first incoming packet after a connection has been established with the client-side. The first packet from the patient program will include the patient ID and an IMSI number, which is used to identify the patient to the rest of the system. The next part of the packet extraction process is to decode the incoming multiplexed multi-channel medical data and place the data from each channel to the appropriate buffer. The data store in each buffer is later used in the data update process. The last part of the packet extraction is to check for the end
of transmission characters when the patient ends a telemedicine session and prepares the server to listen for the next connection request. The protocols reset the incoming data buffer and terminates the existing connection, then displays a message on the user interface that the server is ready for the next telemedicine session. Figure 6.5 summarises the packet extraction process in a flowchart.

![Flowchart of packet extraction process]

The telemedicine session creation routine is activated when the server has identified the patient ID from the packet extraction of the first incoming packet. The routine uses the patient ID to count the number of existing sessions for the patient and create the next session number in the Session table in the database. The current time and date of the session is also
updated in the table. Then the session creation routine uses the session number and patient ID to create the corresponding entry in the Signal Sets table to store the transmitted medical data. The flowchart of the telemedicine session creation routine is shown in Figure 6.6.

![Flowchart of Telemedicine Session Creation Routine](image)

**Figure 6.6** Telemedicine session creation routine

The data updating process is performed when the amount of data in each buffer from the packet extraction stage reaches a preset limit. The data from each channel is updated into the data fields in the Signal Sets table. The current time when the updating process is done is again updated in the Session table to reflect the latest time recorded for the session. The process then clears or resets each buffer to prepare the server for the next set of incoming data. The data updating process is repeated every time the amount
of data in each buffer reaches a preset limit and the latest set of data is appended to the preceding set of data stored in the Signal Sets table. If the end of transmission characters are received during the packet extraction and the amount of data in each buffer is below the preset limit, then the data updating process is also performed to store the current data in each buffer. The data updating process is summarised in a flowchart given by Figure 6.7.

![Data updating process flowchart](image)

Figure 6.7 Data updating process

### 6.3.2 Server Graphical User Interface

The Telemedicine Application Server provides four different interfaces that correspond to four main functions performed by the server, namely the login procedure, active connection control, session list and patient registration. The first interface is the login procedure when a user (e.g. an authorised medical
staff member) starts to run the server and prompt for login name, password, host name for the telemedicine database (in this case the Telemedicine Centre) and the preset port number on the host. The user then proceeds with the log-in option and the next active connection control interface will appear. Figure 6.8 shows the login procedure interface of the Telemedicine Application Server.

Figure 6.8 Telemedicine Application Server Login interface

The active connection control interface consists of administrative controls, data processing display, connection settings and controls as well as transmission information. The active connection control interface for the Telemedicine Application Server is shown in Figure 6.9.

The administrative controls provide options for the user to monitor the number of sessions performed by a patient in the session list interface and to select the patient registration interface. The two options in administrative controls display the session list and patient registration interface respectively. The data processing display shows the size of each buffer from the packet extraction process and the medical data from each channel. The data is displayed in four bytes of hexadecimal format. The display also indicates
when medical data has been successfully updated into the telemedicine database.

![Active connection control interface](image)

**Figure 6.9  Active connection control interface**

The connection settings and controls show the active connection parameters such as the server state, local port number, remote port number, request identification from the client, and options to either reset or shutdown the server. The transmission information displays the variation of incoming data size to the server (in number of bytes) in a graphical format. The total amount of multi-channel medical data received by the server for a particular telemedicine session is displayed in kilobytes (KB). A real-time plot of an average transmission throughput is also included in the transmission information section. In addition, the incoming multi-channel medical data are also displayed on the Telemedicine Application Server, together with its corresponding analogue signal displays. The displays on the active connection control interface are created for the development and demonstration of the
m-Health system. The displays provide useful information and details during the development process of the prototype version; these are normally not essential for an actual or a commercial version.

The session list interface provides a list of patient information and requests for the patient ID to view. A list of telemedicine sessions performed by the patient with the requested patient ID then appears on the interface. The view allows the medical staff to monitor the overall progress of telemedicine sessions for the hospital. There is also an option on the interface to allow the user to return to the active connection control interface. The session list interface is shown in Figure 6.10.

![Session list interface](image)

Figure 6.10 Session list interface

The patient registration interface displays a form for the medical staff to register a new patient to the m-Health system. After a patient has been registered to the system, a current list of patients appears on the interface and the medical staff can then verify that the new patient has been successfully added to the system. The staff can also either update or delete
existing patient information when required. In addition, the interface provides an option for the user to return to the active connection control display. The display of patient registration interface is shown in Figure 6.11.

![Welcome to Mobile Telemedicine Application Server](image)

**Figure 6.11 Patient registration interface**

Both the session list and the patient registration interface require the Telemedicine Application Server to interact and to perform queries to the telemedicine database via SQL to retrieve the data from Session and Patient Information tables respectively.

### 6.4 Patient User Interface Program

The patient user interface program is for the patient to perform the data acquisition process from the telemedicine processor module. In principle, the patient program will read the multi-channel medical data from the processor module via the Bluetooth link and transmit the data to the Telemedicine Application Server via either a fixed-IP network (a LAN, Dial-up internet or
Broadband internet from a service provider) or the GPRS network. The patient program also transmits the patient ID assigned by the hospital upon a successful network connection. This is to allow the Telemedicine Application Server at the hospital to initialise the necessary process to store the data from the telemedicine session into the database. At the beginning of the procedure, the patient would first be required to establish a point-to-point Bluetooth link with the telemedicine processor module, as explained in section 5.1.2. The patient would then need to set up a wide area network connection either via the fixed-IP network or the GPRS network.

6.4.1 Patient Interface on a laptop for fixed-IP or GPRS network

Before running the patient user interface program on a laptop, the patient needs to perform the two initial set-up procedures, as mentioned earlier, the Bluetooth and the wide area connection. The wide area connection set-up for the fixed-IP network follows the same set-up procedure given by an internet service provider for normal internet usage. In the case of a LAN, the connection is normally active when the user logs into the network. The wide area connection with the GPRS network can be achieved by either using a mobile phone or a GPRS data card as a modem. The use of a mobile phone as a modem would require a link between the laptop and the mobile phone, either via a cable, an infrared or a Bluetooth link, as illustrated by Figure 6.12.

![Figure 6.12 Use of a mobile phone as a modem](image)
The patient user interface program provides the graphical interfaces of the necessary controls and displays in order to complete a connection to the m-Health system. The interface is made up of different sections that perform various functions of the program, as shown in Figure 6.13.

The data settings section is where the patient id is entered and the duration of telemedicine session is preset. The connection control section consists of the connection to the Telemedicine Application Server and the Bluetooth connection to the telemedicine processor module.
The patient is asked to enter a password in order to establish a connection with the server. The patient program will be acknowledged once the server has granted the connection request. An option appears on the interface waiting for the patient to start the telemedicine session. The data display section is where the decoded data from the hardware is displayed in a graphical format as well as in a numerical format. Each channel is displayed on a separate window as shown in Figure 6.13.

The received data from the telemedicine processor module are first demultiplexed and placed into temporary buffers for the transmission process. The demultiplexed data for each channel are displayed on the interface program in real-time. The first packet for transmission consists of the patient ID and IMSI number information. The subsequent packets are formed by multiplexing the channel sequentially, as explained in the payload format in chapter five section 5.2.1. The payload format is in Figure 5.3, but without the trailer and the sync character. The data flow process in the patient interface program is summarised in Figure 6.14.

At the end of the telemedicine session, the patient program sends end-of-transmission characters to indicate the last packet. Once the server receives and recognises the end-of-transmission characters, a message is sent back to the client program to acknowledge that all the data transmitted have been received and the server closes the point-to-point connection with the patient.
Figure 6.14 Data flow process in the patient interface program
6.4.2 Patient Interface on a mobile phone for GPRS network

In principle, the application program in the mobile phone reads the multi-channel medical data from the telemedicine processor module via its Bluetooth link. The mobile phone then transmits the data to the Telemedicine Application Server at the hospital via the GPRS network. This set-up offers greater flexibility and ease of use compared with the patient program running on a laptop. However, due to the limitation of processor and hardware capability with a much smaller screen size, the mobile phone patient program has less graphical user interface compared with the patient program on a laptop.

The two initial set-up procedure required on a laptop interface, the Bluetooth and the wide area connection, are performed once the patient starts to run the mobile phone patient program rather than before hand. The program starts with establishing a Bluetooth link with the telemedicine processor module. The next part is to utilise the data connection on the mobile phone by specifying an access point that will be used by the GPRS network to complete the connection. The mobile phone patient program is set to communicate with the Telemedicine Application Server at the hospital by setting the IP address of the server as the destination IP address. Once the client-server point-to-point connection is established, the mobile phone then transmits the medical data to the hospital. The multi-channel data from the telemedicine processor module are also displayed on the screen while being transmitted to the hospital. The software details of the mobile phone patient program, for example destination IP address, socket connection and Bluetooth connection settings, were explained in section 5.3.1 and in Appendix F. The user interfaces on the mobile phone patient program are shown in Figures 6.15 (a) and 6.15 (b). The overall procedures for the mobile phone patient program are summarised in Figure 6.16.
Figure 6.15 (a) Mobile phone patient program- Main menu [6.9]

Figure 6.15 (b) Mobile phone patient program- Test mode [6.9]
Figure 6.16 Overall procedures for mobile phone patient program
The payload format of the mobile phone patient program IP packets differs slightly from the payload format of the patient user interface program on a laptop. The first packet for the network transmission still consists of the patient ID and IMSI number information but the subsequent packets are based on the packet format transmitted by the telemedicine processor module, as explained in section 5.1.1. The subsequent packet payload format for the mobile phone patient program is shown in Figure 6.17.

![Figure 6.17 Packet payload format for mobile phone patient program](image)

The mobile phone patient program are set to transmit the actual data format received via the Bluetooth link due to the small processor and memory capacities on the mobile phone. Only a small buffer size is available to handle the multi-channel data before GPRS network transmission. Therefore, the demultiplexing process of the multi-channel medical data from the telemedicine processor module is done on the server side for the mobile phone patient program. When a patient chooses a stop-session option on the user interface, the patient program sends end-of-transmission characters to indicate the last packet for the session. Once the server has received and recognised the end-of-transmission characters, a message is sent back to the mobile phone patient program to indicate that all the data transmitted has been received and then the server closes the point-to-point connection with the mobile phone.
6.5 Doctor Browser Interface

The doctor browser interface program is to allow clinicians to view and to monitor data from telemedicine sessions as and when required. The browser program allows the clinicians to retrieve patient information, session information and the actual medical data from the telemedicine database. Clinicians may also be given administrative privileges to add, delete or update some information on the telemedicine database.

The browser program can be used with a laptop or portables devices such as a mobile phone and a PDA equipped with network connections. The clinicians may use the browser program from either a fixed-IP network (a LAN, Dial-up internet or Broadband Internet from a service provider) or the GPRS network while on the move. The system architecture of the doctor browser interface program is shown in Figure 6.18. The browser program developed for the m-Health system has two user-interfaces, an interface for a program running on a laptop and an interface for a program running on a mobile phone.

6.5.1 Doctor Browser on a Laptop

The browser program on a laptop allows the clinician to direct access to the MySQL telemedicine database. The ADO database technology is again adopted to allow an SQL query to be embedded into the doctor browser interface program and communicate directly with the telemedicine database via the ODBC database API. These are the same concepts used in implementing the Telemedicine Application Server for the m-Health system.

The main functions in the browser program are the login, the patient selection and the data display process. The user will be first asked to login to
Figure 6.20  System architecture of the doctor browser interface program
the MySQL telemedicine database server by providing a username and password and once authorised by the database server, the patient list from the Patient Information table appears on the interface. The user then specifies a patient ID and the list of telemedicine sessions performed by the patient appears with session details such as session date and time. The user continues to choose a specific session by specifying the session number given in the session list. Once a session number has been specified, a data display screen appears with related information of the patient, telemedicine sessions and the display of medical data. The user then has the option either to view further sessions for the same patient or to enter a new patient ID to view medical data from a different patient. The interface on the browser program at different stages are shown in Figure 6.19 (a,b,c). The flowchart of the doctor browser program on a laptop is given by Figure 6.20.

![Figure 6.19 (a) Doctor browser program Login interface](image-url)
Figure 6.19 (b) Patient list and sessions list interface

Figure 6.19 (c) Doctor browser program data display interface
6.5.2 Doctor Browser on a Mobile Phone

The main advantage of this browser program is to allow the use of a portable device and the GPRS network to access information from the m-Health system while on the move. The browser program described here is specifically for mobile phone usage, although the same concepts and the design platform can be extended to produce a browser program for PDA applications.

In principle, the function of the mobile phone browser program is the same as the browser program on a laptop, which is to retrieve and to display information from the telemedicine database. The main difference between the
two browser programs is the system architecture and the implementation of the SQL queries. The functions supported by the J2ME platform are limited to applications on small portable devices such as mobile phones. The CLDC configuration for mobile phone applications does not support the use of Java Database Connectivity (JDBC) API for communication with a database [6.10]. Therefore, the platform is not capable of embedding an SQL query to communicate with the telemedicine database directly, as it is possible with the browser program on a laptop. One solution is to have a server to act as an intermediate platform to process and to perform the SQL queries to the telemedicine database. The intermediate server connects to the database, retrieves the records, and then passes back to the mobile phone application. The Telemedicine Application Server for the m-Health system is therefore utilised as the intermediate server platform to process and to perform the SQL queries for the mobile phone browser program. The use of the Telemedicine Application Server as the intermediate is transparent to the server user interface. There is no extra user control required on the server and it only manages and controls connection requests from the mobile phone transparently. A certain section of the server implementation handles SQL related requests from the mobile phone and interacts with the telemedicine database. J2ME configurations for applications on the PDA, however, do support the JDBC API for accessing and processing data stored in the database and are part of the JDBC Optional Package for the CDC/Foundation Profile API (JSR 169) [6.11]. Therefore, a PDA application is capable of performing SQL queries to the telemedicine database directly and runs the same operations as in the browser program on a laptop.

The mobile phone browser program starts by establishing a client-server point-to-point connection with the Telemedicine Application Server via its socket connection property. The user is identified by the username and password. The program then sends a request for accessing the telemedicine database using specific protocols and instructions recognised by the server. The instructions from the mobile phone depend on the SQL query required to
retrieve patient and telemedicine session information. Once the server has received the instructions, it processes and decodes the instructions and performs the corresponding SQL query to the telemedicine database. The data and information retrieved from the database are set to a compatible format and protocol and are transmitted back to the mobile phone. The browser program then processes and displays the retrieved data on the limited screen in text format.

There is another possible implementation for the mobile phone browser program using a web-based http connection property of a mobile phone. A web server is required to handle the http request from the mobile phone browser and perform the SQL queries with the telemedicine database. An example of a web server is the Apache HTTP Server which is an open-source web server that can be used with MySQL PHP API to access the telemedicine database [6.6]. PHP is a server-side, HTML-embedded scripting language that may be used to create dynamic web pages and can be compiled as a module for use with the Apache HTTP Server. The PHP codes in the web server perform the SQL queries based on the http request sent by the mobile phone. The retrieved data from the telemedicine database are downloaded to the mobile phone via the http connection and displays on the web browser running on the mobile phone.

The web-based architecture is also compatible with a Java-based server. It is basically the same concept as the PHP-based implementation, but using a different server platform. A system developer with Java language expertise may prefer the Java-based implementation, using a Java server to store a Java servlet required for the program. One of the popular Java servers is the Apache Tomcat Server that supports the Java servlet and Java Server Pages (JSP) implementation. The servlet is the collection of Java codes that understand the request from the mobile phone and performs the SQL queries with the telemedicine database. Instead of using the MySQL PHP API, the java code requires a Java Database Connectivity (JDBC) API to interact with
the telemedicine database [6.10, 6.12]. The servlet then outputs the retrieved data from the database into an html format on a web browser and the same display will appear on the web browser supported by the mobile phone.

Since the mobile phone browser program implemented using the point-to-point connection with the Telemedicine Application Server has limited text only display, the browser program is extended to a Java-based architecture as the intermediary server [6.9]. The queries are submitted to the telemedicine database via the Java-based intermediary server using a http request. These queries are sent using specific protocols and instructions recognised by the intermediary server, which are then translated into the appropriate SQL queries. The response is used to stream the session data back to the mobile phone browser via the Servlets http response. The possible architectures for the mobile phone browser program are summarised in Figure 6.21.
The mobile phone browser program developed in the research has demonstrated a working system for a J2ME-based database access application. The development platform and the system architecture that has been implemented opens up to further expansion on the mobile phone applications. The user interface on the browser program can certainly be enhanced with the availability of higher resolution mobile phones.

6.6 Security Issues

Security is an issue of concern in m-Health applications. The security features that have been implemented with the system are based on specifications produced by the security requirement standard. A comparison of security and encryption for wireless technologies is shown in Table 6.5. The security and encryption features help to protect both the users and the network operator against possible undesirable intrusion of third parties. The security measures are also implemented to insure maximum protection of the individuals' confidentiality. The Bluetooth and the GPRS security and encryption features are implemented in the m-Health system, together with additional security protocols to ensure a secure session for the m-Health applications.

<table>
<thead>
<tr>
<th>Technology</th>
<th>Security / Encryption</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bluetooth</td>
<td>128-bit authentication key and 8-128 bit encryption key [3.42]</td>
</tr>
<tr>
<td>WLAN</td>
<td>Wired Equivalent Privacy (WEP) protocol with RC4 Encryption Algorithm [3.43]</td>
</tr>
<tr>
<td>GPRS</td>
<td>Three-tier security with A3 algorithm for user authentication, A8 Ciphering Key Generating Algorithm and A5 Ciphering Algorithm for Data Encryption [3.44]</td>
</tr>
<tr>
<td>3G</td>
<td>f8 UMTS Confidentiality Algorithm and f9 UMTS Integrity Algorithm [3.45]</td>
</tr>
</tbody>
</table>

Table 6.5 Security and Encryption of Wireless Technologies
The Bluetooth link between the telemedicine processor module and the patient's getaway device (a mobile phone or a laptop) implements the 128-bit authentication key and 8-128 bit encryption key security [6.13]. The getaway devices would have to enter a password in order to establish a secure link with the telemedicine processor module. The patient interface on the laptop also requires a password in order to send a connection request to the Telemedicine Application Server. In addition, the first packet transmitted to the Telemedicine Application Server includes the patient ID and the IMSI number, which will be used by the server to verify the identity of the patient with the system.

The data transmission between the patient and the hospital are protected by the security protocols implemented in the GPRS network. The three-tier security, with the A3 algorithm for user authentication, A8 ciphering key generating algorithm and A5 ciphering algorithm for data encryption security features in the GPRS network, ensures a secure session for the m-Health applications [6.14].

The security on both the Telemedicine Application Server and the MySQL Telemedicine Database Server is implemented based on the MySQL server security model. In general, MySQL implements security based on Access Control Lists for all connections, queries, and other operations that users may attempt to perform [6.2, 6.6]. The users are administered based on an access privilege system, which ensures that all users may perform only the operations allowed to them. The privilege system authenticates a user connecting from a given host, and associates that user with privileges granted by the system administrator. In practice, the users are involved with a two-stage server access control. The first stage is when a user is checked for authorisation to connect to the database. In the second stage, the user is checked based on each request made to see whether the user has the right set of privileges to perform the queries. The security protocol on the Telemedicine Application Server is controlled by the access privilege system in
the MySQL Telemedicine Database Server. The first procedure to run the Telemedicine Application Server is for the user to be granted an access by the MySQL server before proceeding to the server applications.

Both the mobile phone and the laptop browser programs are developed using the MySQL server security model. All the security procedures on the doctor browser programs are similar to the procedures implemented on the Telemedicine Application Server. The only difference is the set of privileges granted to the user, which varies according to their roles and level of authority.

The security protocols discussed here are essentially the first level of security features to ensure a secure connection and to protect medical confidentiality. In discussing security issues, it is acknowledged that the main emphasis is on protecting the entire server host network (not just the MySQL server) against all types of possible attacks: eavesdropping, altering, playback, and denial of service. A further security implementation is the encryption and subsequent decryption of the medical data to increase data protection.
6.7 Summary

The m-Health system is integrated with client-server application programs to allow the monitoring and the management of medical data. The Telemedicine Application Server is responsible for handling the telemedicine session and controlling the client connection request from a patient. The medical data transmitted during a telemedicine session are stored in a database, together with patient information and telemedicine session details. The patient user interface programs are developed for the client side to receive digitised medical data from the telemedicine processor module and send the data to the hospital server via the GPRS network. The patient programs are available for both a laptop and mobile phone environment. The doctor browser programs allow the doctors to view and to monitor the medical data as well as the patient details from the database. The browser program on a laptop offers much more controls and options to the clinicians than the limited browsing capability for the browser program on portable devices. The browser program can be used with a fixed-network in the hospital or with a mobile network while on the move. A comparison of security and encryption for wireless technologies highlights the security features available. The security features implemented in the m-Health system help to protect the clinical data integrity and the medical confidentiality of the patient.
6.8 References


CHAPTER SEVEN:

PERFORMANCE ANALYSIS AND TECHNICAL RESULTS

7.1 Overview of m-Health System Performance

The choice of utilising the commercially available GPRS network for the m-Health system was based on several factors. The main factor was the network coverage offered by the GPRS network, which is currently available over most of the UK. The second factor was the availability of GPRS mobile phone models (with large memory and sophisticated displays). There are many commercially available models that support GPRS, only with different multi-slot classes. Although high-bandwidth Third Generation (3G) cellular networks are readily available, the availability of the 3G networks and 3G mobile phones is still limited at this stage.

However, the m-Health system is inter-operable with 3G networks. The processor module is capable of handling more analogue and digital data by modifying the main software program. The communication gateway can be simply upgraded to a 3G mobile phone that supports J2ME MIDlets. The client-server platform, i.e. the application server, the browser interface, and
the telemedicine database, are all based on IP protocols. These features allow the system to be explored beyond 3G technologies, in the future all IP-based 4G networks with user access to any system at anytime and anywhere [7.1].

The performance of the mobile telemedicine system is evaluated based on the theoretical analysis of GPRS throughput and delay. The performance focuses on the uplink channel of the GPRS network and how the GPRS access network affects the behaviour of mobile telemedicine applications. The measurements obtained during mobile tests are compared with estimated theoretical results. The mobile tests contribute to the system evaluation considered from both user and theoretical perspective. The mobile tests also represent how the system can be used in telemedicine sessions and include possible variations in mobile environments that reflect the patient mobility situations. The mobile tests were performed in an indoor location, at a local town centre, along country roads and along motorways. The mobile tests used digitised ECG records from the MIT-BIH Arrhythmia database as well as real medical signals from a test patient [7.2]. The mobile tests were done with the patient interface on a laptop and the patient interface on a mobile phone.

### 7.2 Performance Analysis over the GPRS network

With the introduction of packet data traffic in the GPRS network, the scheme used in managing resource allocations for the GPRS radio access is done dynamically according to the "capacity on demand" principles [7.3]. Allocation of physical channels for the GPRS radio access is flexible and radio resources are taken from the common pool of available physical channels in a radio cell. The GPRS network does not require permanently allocated packet data traffic channels and the resource allocation is based rather on the needs for actual packet transfers. The flexibility of channel allocations in the GPRS service allows one to eight channels to be allocated to a user or one channel can be shared by several users [7.4]. In addition, the channel allocations are also
based on demand to ensure flexible adaptation to different traffic conditions [7.5]. However, if the network receives a service request of higher priority, these data channels can also be de-allocated if required. Due to technical limits of the GPRS network and randomness of radio resources availability, current research continues to improve the uplink radio resource utilisation and to optimise the dynamic resource allocation procedure [7.6].

7.2.1 Throughput Analysis

Several studies have investigated the performance of the GPRS network based on a network simulation model, an analytical model and a practical evaluation approach [7.5, 7.8-7.11]. The results and findings from these studies are used as benchmarks in comparing the performance of the mobile telemedicine system. The discussion on the system performance of the GPRS network in this research focuses mainly on the data throughput, error control techniques and packet delays.

The first parameter for the system performance analysis is the data throughput of the GPRS network. This section provides an overview on the theoretical part the GPRS data rate and its throughput performance.

The choice of coding scheme in GPRS data transmission reflects its throughput performance. The CS1 coding scheme gives a data rate of 9.05 kbps and the CS4 coding scheme gives the highest data rate of 21.4 kbps, as listed in Table 3.9. The choice of coding scheme also has a stronger influence on throughput performance for long data packets than for short data packets [7.12]. For example, the CS1 coding scheme is sufficient for the transmission of data packets of length 100 bytes under all channel conditions. In contrast, it is critical to select a coding scheme best suited for a certain channel condition for the optimum transmission of data packets of length 1600 bytes.
In network implementation, link adaptation is widely used to maximise throughput performance under varying channel conditions. The network dynamically switches among coding schemes to achieve the highest throughput [7.13]. The operation is based on the knowledge of which coding scheme produces the best performance under any condition.

With the limitation of only CS1 and CS2 coding schemes being implemented by most of the GPRS network operators, it is expected to observe much lower uplink throughputs than the theoretical values for the mobile telemedicine system during the field tests. With the variation in data packet sizes, mobile channel conditions and slot contentions from multiple users, significant variations in the GPRS throughput performance are predicted.

7.2.2 Packet Loss and Packet Delays

In general, there is always a possibility of data being corrupted during the transmission from a source to its destination. Reliable communication systems must therefore have a mechanism for detecting and correcting such errors. Therefore, the next parameters for the system performance analysis are the packet loss and packet delays in the GPRS data transmissions. Packet loss and packet delays are largely influenced by the erroneous nature of the wireless channel (due to distance losses, shadowing, and multi-path fading). Further discussion on the erroneous nature leads to the theoretical overview on error control techniques and packet delays in the GPRS network.

Forward Error Correction (FEC) and Automatic Repeat Request (ARQ) are two error control techniques implemented in the GPRS network [7.10]. In FEC, redundant bits are added to information bits to detect and correct any channel-induced errors. In ARQ technique however, error control is achieved not through error correction, but through the retransmission of erroneous data packets. While FEC is applied at the physical layer, ARQ can be applied
at different layers of the protocol stack. Unsuccessful error correction by the FEC at the physical layer can later be handled by the ARQs at higher layers. For example, the ARQs are implemented at the radio link-control (RLC) and logical link-control (LLC) layers in the GPRS protocol stack, in addition to employing FEC at the physical layer.

The channel coding schemes implemented at the physical layer provides protection to the transmitted data packets against transmission errors via the FEC [7.4, 7.15, 7.16]. The CS1 to CS3 coding schemes use convolutional codes and block-check sequences of varying strengths, so as to produce different rates. The CS1 to CS3 coding schemes are based on a 1/2 rate convolutional code, which is punctured to obtain approximate rates 1/2, 2/3, and 3/4, respectively. On the other hand, CS4 only provides error detection functionality.

Since the coding scheme adopted in the GPRS transmission varies according to radio channel conditions, the amount of radio blocks required in the data transmission also depends strongly on the radio channel conditions [7.8]. The Carrier-to-Interference ratio (C/I) represents a channel condition and is indicated by the Bit Error Rate (BER). If the C/I ratio indicates a radio channel with a high BER then the RLC protocol has to use strong coding redundancy in order to assure good radio block error resilience and requires more radio blocks for the transmission. On the other hand, if the C/I ratio represents a good radio channel condition then the coding redundancy can be smaller and fewer radio blocks are required to transport the same IP packet.

The coding scheme adopted for a particular GPRS data transmission, based on the C/I ratio, provides the channel protection at the physical layer. In addition, unsuccessful error correction at the physical layer is later handled by the ARQs at a higher layer. One important aspect of ARQ is the influence of ARQ retransmission time on the delay of radio blocks [7.8]. The retransmission time occurs when the receiver sends back to the transmitter a
report about corrupted radio blocks in the transmission so that the transmitter
can retransmit these corrupted radio blocks.

Due to the ARQ implementation at the RLC layer, packets are rarely lost even
when link outages occurred, just grossly delayed in the GPRS network. The
combination of FEC and the ARQ techniques help to improve the GPRS radio
link in terms of packet error and packet loss. However, packet loss does occur
over GPRS links in both the downlink and uplink directions, but the incidence
is relatively rare and hard to quantify.

Theoretically, the GPRS network allows different Quality of Service (QoS)
profiles in terms of service priority, reliability, delay and throughput [7.16].
The reliability guarantees certain maximum values for the probability of loss,
duplication, mis-sequencing, and corruption (an undetected error) of packets.
There are three reliability classes available, as shown in Table 7.1.

<table>
<thead>
<tr>
<th>Class</th>
<th>Probability for</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lost Packet</td>
</tr>
<tr>
<td>1</td>
<td>$10^{-9}$</td>
</tr>
<tr>
<td>2</td>
<td>$10^{-4}$</td>
</tr>
<tr>
<td>3</td>
<td>$10^{-2}$</td>
</tr>
</tbody>
</table>

Table 7.1  GPRS reliability classes

The delay parameters define maximum values for the mean delay. A delay is
defined as the end-to-end transfer time between two communicating mobile
stations or between a mobile station and the interface to an external packet
data network. This includes all delays within the GPRS network, e.g., the
delay for request and assignment of radio resources and the transit delay in
the GPRS backbone network. The delays classes defined in the GPRS network
are shown in Table 7.2. In practice however, mobile network operators support the best-effort service class.

<table>
<thead>
<tr>
<th>Class</th>
<th>128 Byte Packet</th>
<th>1024 Byte Packet</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean Delay</td>
<td>95% Delay</td>
</tr>
<tr>
<td>1</td>
<td>&lt;0.5s</td>
<td>&lt;1.5s</td>
</tr>
<tr>
<td>2</td>
<td>&lt;5s</td>
<td>&lt;25s</td>
</tr>
<tr>
<td>3</td>
<td>&lt;50s</td>
<td>&lt;250s</td>
</tr>
<tr>
<td>4</td>
<td>Best Effort</td>
<td>Best Effort</td>
</tr>
</tbody>
</table>

Table 7.2 GPRS delays classes

At the application layer, it is difficult for a user application to distinguish between a packet that appears to be lost due to large delays and a truly lost packet. The packet delays in GPRS data transmission are reflected by the time taken to complete a transmission. The essential question is how long is it worth waiting for one packet. In the mobile telemedicine application, the essential part is to receive securely all the transmitted medical data. With the implementation of the telemedicine database to store the transmitted data at a predetermined interval, the waiting time for data packets in the system is not as critical as in real-time applications.

The performance aspects of GPRS network also involves the unavoidable delays that network users do experience. Common sources of delays in the GPRS network are as follow [7.19, 7.20]:

- Processing and queuing delays in each protocol stack
- Buffering delays caused by the data collection for packet formation
- Substantial delays while negotiating for access to radio resources
- Delays caused by the retransmission of lost radio blocks and packets
- Delays due to mobility management
The sources of delays in the GPRS network are mainly on the radio interface part. Delays inside the operators' Base Station Subsystem onwards and in the GPRS backbone network are small relative to the delay of the radio interface. As predicted, the delay in the radio interface is directly related to the size of data packet being sent [7.20].

The user mobility pattern in the GPRS network also contributes to the possible delays. The case where a user is transmitting and receiving data packets while on the move may cause small or large delays due to handover process from one serving base station to another. Large delays are expected during a new logical channel establishment when a user moves to a cell being served by another Serving GPRS Support Node (SGSN). These signalling delays are predicted to be at least several seconds [7.20].

The theoretical overview on possible sources of transmission delays in the GPRS network is used as a guideline in predicting or explaining the outcome of the mobile field tests. The variation in the test parameters is performed to investigate the effects of varying the sources of delays to the results of the measured parameters.

### 7.3 Mobile Field Tests

The prototype of the mobile telemedicine system was tested with the Vodafone GPRS network. The coding schemes adopted for the transmissions were therefore assumed to be the CS1 and CS2 schemes implemented by the Vodafone network. The mobile tests utilised the up-link transmissions of TCP/IP over the GPRS network. The medical data were transmitted from remote locations using the patient interface on a laptop and the patient interface on a mobile phone to the Telemedicine Application Server in the lab. On both patient interface programs, the medical data were encapsulated into IP packets, which were then transformed into GPRS access protocols format
for the transmission to the external packet data network at the server end.

The mobile tests were performed to observe any variation in the performance of the system based on a few parameters. The mobile environment, the total data, the multi-slot classes and the time of day parameters were varied and the effects on the system performance were monitored. The performance of the system was measured based on the total time taken to complete a transmission, referred to as completion time, the average throughput of a GPRS transmission, and verification that the total data received matched the total data transmitted.

### 7.3.1 Test Parameters

The mobile tests were performed using clinical ECG records from the MIT-BIH arrhythmia database and real multi-channel medical data from a test patient. The clinical ECG records were the MIT-BIH record 103, belonged to a male patient. The ECG lead configurations used for record 103 were a modified limb lead II (MLII), obtained by placing the electrodes on the chest and a modified lead V2, also with the electrodes placed on the chest.

The real multi-channel medical data from a test patient were obtained from the four analogue inputs of the telemedicine processor module. The ECG signal was obtained by placing the electrodes on the test patient with the Lead I ECG configuration, as shown in Figure 4.8 in Chapter 4. The other three input signals were from a temperature sensor, a potentiometer that simulates blood pressure readings and a test signal from an unconnected analogue input connector.

The performance of the system was investigated in four different mobile environments that reflect possible patient mobility pattern in the real world. The mobile tests were performed in an indoor location (within the department), at a local town centre, along country roads and along high-
speed motorways, in and around Loughborough area. These variations in mobile channel environment represented differences in path delays, mobile speed and noise conditions of the GPRS network.

The variation in the total data reflects the duration of the recorded medical signals being transmitted. These data were transmitted from the patient interface program on the laptop and data packets were captured by the Telemedicine Application Server in the lab. The test procedures involved are further explained in section 7.3.3. The recorded ECG signals of lead MLII and V2 from the MIT-BIH database were compiled in a data format recognised by the system with different data sizes, where one half of the total recording duration were signals from each lead configuration. For example, the 84 Kbytes data size reflects 60 seconds of recorded ECG signals with 30 seconds of ECG signals from the MLII lead and 30 seconds of ECG signals from the V2 lead. The data size variations used in the mobile tests are listed in Table 7.3. On the other hand, the data size variations for the real data input from the test patient was set to short, medium and long duration of telemedicine sessions. The short duration was around 20 to 50 Kbytes of data, the medium duration was around 90 to 110 Kbytes of data and the long duration was about 200 to 220 Kbytes of data.

<table>
<thead>
<tr>
<th>Data Size (Kbytes)</th>
<th>ECG Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>10 Seconds</td>
</tr>
<tr>
<td>84</td>
<td>60 Seconds</td>
</tr>
<tr>
<td>168</td>
<td>2 Minutes</td>
</tr>
<tr>
<td>337</td>
<td>4 Minutes</td>
</tr>
<tr>
<td>506</td>
<td>6 Minutes</td>
</tr>
</tbody>
</table>

Table 7.3 Data size variations used in the mobile tests

The variation in terms of the GPRS multi-slot classes was achieved by using different mobile phone models during the mobile tests. Three models were tested and were expected to produce some variation in the measured values
of completion time and average throughput. The models and their corresponding multi-slot capabilities used in the mobile tests are listed in Table 7.4.

<table>
<thead>
<tr>
<th>Mobile Phone Model</th>
<th>Mobile Class</th>
<th>Multi-slot Capability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sony Ericsson T610</td>
<td>Class 8</td>
<td>4+1</td>
</tr>
<tr>
<td>Nokia 6600</td>
<td>Class 6</td>
<td>3+2</td>
</tr>
<tr>
<td>Nokia 6630</td>
<td>Class 10</td>
<td>4+2</td>
</tr>
</tbody>
</table>

Table 7.4  Mobile phone models used in the mobile tests

Since the GPRS radio resources are allocated based on the user requests and the resources may be decreased due to slots contention from multiple users, the variation in terms of the time of day was also taken into consideration. The availability of GPRS radio resources during peak and off-peak hours may have some effects on the average throughputs and the completion time. Mobile tests were performed between 12pm and 4pm for the peak hour’s scenario. The off-peak hours tests were performed at around 9pm and during a weekend.

7.3.2  Mobile Tests Procedure

In general, test parameters were varied one at a time. For example, the data size was varied to indicate different monitoring durations while maintaining the other existing test parameters. The completion time and average throughput were then measured, and any packet lost was verified. The measured values were stored in the telemedicine database for further analysis. Acknowledgement messages were sent by the server to the remote patient upon receiving a complete data set.

The variation in data size was performed in all four different mobile environments and with all three GPRS multi-slot classes. Both the MIT-BIH
clinical ECG records and the real multi-channel medical data from a test patient were transmitted from different mobile environments, except for the motorway environment where only the MIT-BIH clinical ECG records were used. The time of day variation was only performed in an indoor mobile environment with different total data sizes and GPRS multi-slot classes. The list of parameters variation in each mobile environment are summarised in Table 7.5.

<table>
<thead>
<tr>
<th>Mobile Environment</th>
<th>Parameters Variation</th>
<th>Medical data types</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indoor</td>
<td>Data Size</td>
<td>MIT-BIH ECG Records</td>
</tr>
<tr>
<td></td>
<td>Multi-slot Classes</td>
<td>Real Multi-channel Data</td>
</tr>
<tr>
<td></td>
<td>Time of Day</td>
<td></td>
</tr>
<tr>
<td>Town/Urban</td>
<td>Data Size</td>
<td>MIT-BIH ECG Records</td>
</tr>
<tr>
<td></td>
<td>Multi-slot Classes</td>
<td>Real Multi-channel Data</td>
</tr>
<tr>
<td>Countryside</td>
<td>Data Size</td>
<td>MIT-BIH ECG Records</td>
</tr>
<tr>
<td></td>
<td>Multi-slot Classes</td>
<td>Real Multi-channel Data</td>
</tr>
<tr>
<td>Motorways</td>
<td>Data Size</td>
<td>MIT-BIH ECG Records</td>
</tr>
<tr>
<td></td>
<td>Two Multi-slot Classes</td>
<td></td>
</tr>
</tbody>
</table>

Table 7.5 List of parameters variation in each mobile environment

The average throughput of the transmission was also displayed by the Telemedicine Application Server in real-time. The incoming data arrivals into the server were averaged over fixed 5-second intervals that enabled variations in average throughput to be observed. Since the calculation for the average throughput was made based on the incoming user data (the payload of IP packets), the protocol overheads involved in the GPRS network and packet data network transmission were not taken into account. Therefore, the average throughput measured by the server estimates the effective data rate, which is lower than the total data rate for the GPRS network.
In addition, the incoming multi-channel medical data from a test patient were also displayed on the Telemedicine Application Server, together with its corresponding analogue signal displays.

7.3.3 Mobile Tests Set-up

The first set-up for the mobile tests was the set-up used for transmitting ECG records from the MIT-BIH database using a patient interface program on a laptop. The laptop was first configured to access the GPRS network via the mobile phone as a modem, as illustrated in Figure 7.1. The patient interface program then sends a connection request to gain access into the Telemedicine Application Server in the lab. The IP address of the server was 158.125.51.22 and the application server was running on port 2206. The patient interface program on the laptop used for the mobile tests is shown in Figure 7.2. Once the point-to-point connection with the server is established, the server then sent a notification to the patient interface to inform a successful connection request. The patient interface then starts the data transmission and was set to transmit the digitised ECG data stored in the laptop.

The start time of the transmission was recorded from the first packet being transmitted and the end time was recorded when all the data have been received by the server. The patient interface then received an acknowledgement message from the server upon successful transmission of all the ECG data. These procedures were repeated for sending different data sizes representing different monitoring durations. For the variation in different mobile environments, the set-up was tested at different locations with different mobile speeds. For the variation in the multi-slot classes, different mobile phones were used as a modem in the set-up.
Figure 7.1  Set-up for transmitting ECG records from the MIT-BIH database using a patient interface program on a laptop.

Figure 7.2  Patient interface program on the laptop for the MIT-BIH database.
The second set-up for the mobile tests was for transmitting real multi-channel medical data from a test patient using a mobile phone, as shown in Figure 7.3. Four analogue inputs of the telemedicine processor module were connected and the digitised data from the processor module were then transmitted to the mobile phone. The patient interface program on the mobile phone handled the Bluetooth link establishment with the processor module and the GPRS data transmission to the Telemedicine Application Server. The patient interface program on the mobile phone for the mobile tests is shown in Figure 7.4. The completion time and the total data transmitted were recorded in the database when the test patient stopped the test session using the 'stop session' option on the mobile phone interface. The start time was recorded when the first packet from the test patient was received by the server. The end time was recorded when the end of transmission characters were received by the server as soon as the test patient presses the stop option on the mobile phone interface. The implementation of the second set-up during the field test is shown in Figure 7.5.

![Figure 7.3 Set-up for transmitting real multi-channel medical data from a test patient using a mobile phone interface](image)
The third set-up for the mobile tests remained for transmitting real multi-channel medical data from a test patient, but required a laptop as an intermediate platform to gain access to the GPRS network. This set-up was only used for the indoor test environment due to its complexity, as depicted in Figure 7.6. A patient interface on the laptop read the digitised data from the processor module via its Bluetooth link and transmitted the data to the
Telemedicine Application Server via the GPRS network. The GPRS connection to the Telemedicine Application Server was again configured using the mobile phone as a modem. This set-up also demonstrated how the mobile telemedicine system could be used with other wireless networks, such as a Wireless Local Area Network (WLAN) and a Metropolitan Area Network (MAN) (e.g. IEEE 802.16, Local Multi-point Distribution Service (LMDS)) and with fixed-IP networks (e.g. ISDN, Asymmetric Digital Subscriber Line (ADSL), Asynchronous Transfer Mode (ATM), etc.). The start and the end time for the third set-up were configured similar to the second test set-up.

![Diagram of set-up](image)

**Figure 7.6** Set-up for transmitting real multi-channel medical data with a laptop as an intermediate platform

The test set-ups for the different mobile environments were prepared in a car and were evaluated while driving in a car at the town centre, along country roads and along motorways, as shown in Figure 7.7.

![Image of car](image)

**Figure 7.7** Test set-up was prepared and evaluated while driving in a car
A field test table was created in the database to store the total data transmitted, the date, the start time and the end time of the transmissions. The completion time of a transmission was the time duration between the start time and the end time recorded. It is acknowledged that the method used did not give a precise measurement of the transmission time of the data packets. However, the same method was consistently used throughout the mobile tests to best estimate the transmission time. The estimated throughput for the transmission was produced from dividing the total data transmitted over the completion time recorded.

7.3.4 Indoor

All three test set-ups and the corresponding test parameters variations were successfully evaluated in the indoor mobile environment. The tests were performed in a static position in the lab and while walking along a corridor in the building.

7.3.5 Town/Urban

The test set-up for MIT-BIH ECG records and the test set-up for the actual data from a test patient using a mobile phone were evaluated in the town environment. Both set-ups were used while driving in the Loughborough town centre with slow moving traffics and multi-path effects from surrounding buildings and multi-storey car parks. The mobile speed was on average between 10 to 20 mph. The mobile test set-up for the actual data from a test patient was performed in rainy weather conditions.
7.3.6 Countryside

The performance of the mobile telemedicine system was also evaluated along the country roads around the Loughborough area. The same test set-ups for the town environment were used while driving along the country roads. The mobile tests were performed along Breakback Road, along Dean's Lane and near the top of Beacon Hill. These test locations are shown in Appendix G. The surroundings of the test locations were farmland, woods and close to local walking paths. The mobile speeds were varied from stationary to walking pace and to the cruising speed of about 20 mph. The system was also tested at a speed of up to 60 mph at the Snell's Nook Lane towards the Nanpantan Road.

7.3.7 Motorways

The mobile tests in the motorways environment were performed on the M1 motorway (between Junction 23 and Junction 22), along the A50 dual carriageway towards Markfield and along the A50 towards Coalville. The mobile speed variations were about 70 mph on the M1, up to 80 mph on the A50 towards Markfield and about 55 mph and down to a halt on the A50 towards Coalville. The mobile tests along the motorways were performed during a clear and sunny weather condition. The only test set-up used for the motorways environment was with the MIT-BIH ECG records on the laptop interface.

7.4 Mobile Tests Result

The mobile tests were performed with the aim of demonstrating and investigating the performance of the mobile telemedicine system under different mobile environments. The results were used in evaluating the
effectiveness of using the mobile network for the up-link transmission of multi-channel biomedical signals with the actual patient mobility scenario.

The three different test set-ups were successfully evaluated under the indoor, town/urban, countryside and motorways environments. Although different test set-ups were used in different mobile environments, all the three test set-ups were successfully tested in the indoor environment. The system performance results based on the variations in the test parameters are presented according to the test set-up used and the medical data transmitted.

7.4.1 ECG Records from the MIT-BIH Arrhythmia Database

The results presented in this section were from the set-up used for transmitting ECG records from the MIT-BIH database using a patient interface program on a laptop. The ECG data were from the lead MLII and V2 configurations. The tests were performed in four different mobile environments and various data sizes. In terms of the time of day, the transmissions of the ECG records were performed during peak hours, between 12 to 5pm on weekdays. The graphical display of the lead MLII ECG signals received and stored in the telemedicine database is shown in Figure 7.8. The ECG signals are displayed by the doctor browser interface.
Figure 7.9 illustrates the comparison of the total completion time for various data sizes under the respective mobile environments. The mobile tests were performed using a GPRS Class 8 (4+1 slots) Sony Ericsson T610 mobile phone with one up-link transmission slot. As expected, the completion time increased with the increment in the data size and complies with the theoretical fact that delays in the radio interface are directly related to the size of data packet being sent. Among the four mobile environments, the indoor environment has the fastest completion time for the transmission of all the data sizes.

The expected pattern of increasing completion time with the increase in data size is consistent in all the tests for both mobile classes. However, the range of estimated throughput values for both tests are much higher than the theoretical GPRS throughput values (even under ideal conditions). For example, the maximum possible throughput for the Class 8 mobile phone with one up-link timeslots under CS2 coding scheme is 13.4 kbps. Since the throughputs for the tests are estimated from the completion time of each
transmission, a suitable explanation of the behaviour is that the completion time is much faster than the minimum completion time possible. For example, the transmission of 337 Kbytes of data would require at least 206 seconds of completion time with the maximum possible throughput of 13.4 kbps. From the results, the completion time for the transmission of 337 Kbytes of data under the indoor mobile environment with the Class 8 mobile phone is 73 seconds, which is one third the minimum completion time possible. Theoretically, this is only possible with the implementation of a data compression mechanism during the transmission.

After a thorough investigation, it turns out that the software used for accessing the GPRS network on the laptop interface was defaulted to a compression mode for GPRS data transmission. Since it was commercial software provided by the network operator, the end users had no information on the exact compression ratio implemented. In order to estimate the compression ratio, the tests for the indoor mobile environment for both mobile classes were repeated with the compression mode turned off.

The results for the indoor mobile environment with the compression mode turned off showed an almost identical pattern to the results when the compression mode was activated. The only difference is that the transmissions with the compression mode turned off required a much longer completion time than for the previous tests. Therefore, the results for other mobile environments were expected to produce the same general pattern as for the previous field tests, only on a different scale. Under this assumption, the results for the other mobile environments with the compression mode turned off are estimated, based on the ratio calculated from the indoor mobile environment. Although it is acknowledged that the exact completion time for the transmissions with other mobile environments may not be directly related to the ratio from the indoor tests, the estimation is made to illustrate the possible results and behaviour of the system under different mobile environments.
From the graph, there is no significant difference on the completion time among the four mobile environments for data transmissions of less than 100 Kbytes. The transmissions of data sizes larger than 100 Kbytes performed under the countryside mobile environment are far more distinct, requiring a longer completion time compared to the other mobile environments. The plots with longer completion times in Figure 7.9 illustrate the total completion time for various data sizes for the transmissions when the compression mode is turned off, using the GPRS Class 8 mobile phone. For comparison, the completion time recorded for the 337 Kbytes of data for the indoor environment is now 315 seconds. With the maximum possible throughput of 13.4 kbps (assuming CS2 coding scheme), the minimum completion time possible for the transmission is 206 seconds. The results are now within the theoretical values.

Figure 7.10 illustrates the estimated throughputs of the mobile tests under the four different mobile environments with respect to the different data sizes. As predicted, the indoor environment allowed much higher throughputs than other mobile environments. On average, the throughputs are in the
region of around 15 to 40 kbps with the compression mode. On average, the throughputs for all the mobile environments with no compression are now in the region of 4 to 9 kbps. The throughputs are within the maximum theoretical values of 13.4 kbps under the CS2 coding scheme for one up-link timeslot capability.

![Figure 7.10](image)

Figure 7.10 Throughputs for various data sizes under different mobile environments using a GPRS Class 8 mobile phone

Figure 7.11 illustrates the comparison of the total completion time for various data sizes under the same test set-up but using a GPRS Class 6 (3+2 slots) Nokia 6600 mobile phone with two up-link transmission slots. The indoor environment still produces the fastest completion time for all the transmissions. The interesting point to note is that the completion time for all four mobile environments are quite close together for transmission of up to around 200 Kbytes before starting to spread out for larger data sizes. The total completion time for the transmissions of various data sizes when the compression mode is turned off are significantly longer. The completion time recorded for the 337 Kbytes of data under the indoor environment is now 188 seconds and complies with the minimum completion time possible of 103 seconds for two up-link timeslots with the CS2 coding scheme.
Figure 7.11  Completion time for various data sizes under different mobile environments using a GPRS Class 6 mobile phone

Figure 7.12 illustrates the estimated throughputs for the Class 6 mobile phone. Transmissions with compression mode yield throughputs on average in the region of 30 to 40 kbps for the country and motorway environments. The throughputs for the indoor and town environments are quite high with the compression mode, which in this case is in the region of 45 to 65 kbps. The throughputs for the country and motorway environments are on average in the region of 7 to 10 kbps when the compression mode is turned off. On the other hand, the average throughputs for the indoor and town environments are now in the region of 10 to 15 kbps. However, a high throughput of about 24 kbps was recorded for the indoor environment and was quite close to the maximum possible throughput of 26.8 kbps under the CS2 coding scheme.
Figure 7.12 Throughputs for various data sizes under different mobile environments using a GPRS Class 6 mobile phone

The results match the expectation of the performance difference when two different GPRS mobile classes were used. The completion time for the Class 6 mobile phone is approximately half to that of the Class 8 and the average throughput for the Class 6 is almost double that of the Class 8 mobile phone.

7.4.2 Actual Multi-channel Data via a Mobile Phone Interface

The second set-up for the mobile tests was for transmitting real multi-channel medical data from a test patient using a mobile phone. The tests were performed using a Nokia 6630 Class 10 (4+2) mobile phone. The indoor tests were performed in the lab and along the corridor in the department. The first part of the outdoor tests were around the town centre location, from stationary to cruising speeds moving towards a country road. The last of the outdoor tests were done on a country road moving towards the university. Most of the outdoor mobile tests were performed under rainy weather conditions.
The variation in total data transmitted from the test patient is set to represent a short, medium and long duration telemedicine sessions. The short duration is around 20 to 50 Kbytes of data, the medium duration is around 90 to 110 Kbytes and the long duration is about 200 to 220 Kbytes. The choice of total data transmitted is made on the basis that these would be reasonable data lengths for monitoring sessions. The actual total data are clearly variable, depending on the specific medical requirements. The display on the mobile phone interface while transmitting the medical signals is shown in Figure 7.13.

![Mobile phone interface display](image)

**Figure 7.13** The mobile phone interface display while transmitting medical signals during the mobile tests

Figure 7.14 illustrates the completion time for the transmissions of multi-channel medical data for both indoor and outdoor mobile test environments. The indoor completion time is slightly faster than the outdoor time for most of the transmissions. The completion time for the indoor transmissions increases proportionally with the increase in total data size.
Figure 7.14  Completion time for multi-channel medical data with both indoor and outdoor mobile tests environments

The estimated throughputs for the mobile phone interface transmissions are illustrated in Figure 7.15. The throughputs for the indoor tests are quite constant at around 5.5 to 6.5 kbps. On the other hand, the throughputs for the outdoor tests are generally lower than indoor and were only around 4 to 6 kbps. These throughputs observed for the transmission of real multi-channel medical data from the test patient for the indoor and the outdoor environments were much lower than the possible up-link throughputs for the Class 10 mobile phone. The Class 10 mobile phone is capable of up to 26.8 kbps for the up-link transmission.

Figure 7.15  Throughputs for multi-channel medical data with both indoor and outdoor mobile tests environments

176
The significantly low throughput observed in the tests is very likely due to the packet size transmitted by the mobile phone. Since the GPRS throughput is directly related to the size of IP packets set for a transmission, a small IP packet has a substantially lower throughput compared with larger IP packet size. Based on the visual observation on the incoming IP packet sizes on the Telemedicine Application Server during the indoor tests, the average incoming payload size for each IP packet is between 200 to 300 bytes. This shows that the mobile phone is transmitting a small data per IP packet.

The size of IP packets transmitted by a mobile phone is controlled by its hardware configuration. The J2ME platform on the phone is responsible for handling incoming data from the Bluetooth link and preparing the data for GPRS transmission via the socket connection. The MIDP profile in the J2ME platform includes networking support and defines the socket stream connection interface. The sending buffer size option informs the low level networking code about the intended usage patterns that the application will use with the socket connection [7.23, 7.24]. The option is also a hint to the J2ME platform of the sizes intended for the underlying network I/O buffers. The sending buffer size option can be preset but the actual upload size is controlled by the phone and the actual transmissions are based on the size chosen by the phone. The phone's system may adjust the buffer sizes to account for better throughput available based on the data from the current network information.

7.4.3 Actual Multi-channel Data via a Laptop Interface

The set-up for this section is for transmitting multi-channel medical data from a test patient using a laptop as an intermediate platform to gain access to the GPRS network. The set-up was only used in the indoor test environment using the Class 8 Sony Ericsson T610 and the Class 6 Nokia 6600 mobile phones as modems. The transmissions of the multi-channel medical using the set-up
were performed during off-peak hours, i.e. at about 9pm in the evening and on a weekend. The variation in total data transmitted from the test patient is the same as the variation with mobile phone interface sessions. As before, the short duration is around 20 to 50 Kbytes of data, the medium duration is around 90 to 110 Kbytes and the long duration is about 200 to 220 Kbytes. The display of the patient interface program on the laptop while transmitting medical signals for the indoor test environment is presented in Figure 7.16.

Figure 7.16 Display of the patient interface program while transmitting medical signals for the indoor environment
CHAPTER SEVEN

Figure 7.17 illustrates the completion time for the transmission of multi-channel medical data with the laptop as the intermediate interface. As expected, the completion time with the two up-link timeslots, Class 6 mobile phone, is much faster than the single up-link timeslot, Class 8 mobile phone. Although the completion time is faster for all data sizes, the difference is more distinct for data sizes that are larger than 50 Kbytes.

Since the tests were performed during off-peak hours, the previous results from the indoor tests during peak hours with the MIT-BIH ECG records were used as comparisons. The transmissions of up to 337 Kbytes of the ECG records were compared with transmissions of multi-channel data for Class 6 and Class 8 mobile phones. All the tests indicate a steady proportional increase in completion time as the data sizes increase. The main observation from these tests is that the completion time for the transmissions during off-peak hours is generally faster than the completion time during peak hours.

Figure 7.17  Completion time for the mobile tests using the laptop interface with the time of day variation

Figure 7.18 illustrates the estimated throughputs for the indoor tests with the laptop interface. The throughputs for the off-peak transmissions are quite consistent in the region of 16 to 19 kbps with the Class 6 mobile phone, whereas for the Class 8 mobile phone yields throughputs in the region of 8 to
12 kbps. Both mobile classes had throughputs within the maximum possible rates of 26.8 kbps and 13.4 kbps for Class 6 and Class 8, respectively.

As comparisons, the estimated throughputs for the peak hour's transmissions are on average between 13 to 15 kbps for the Class 6 and between 7 to 9 kbps for the Class 8 mobile phone. The throughputs using the Class 8 for the off-peak and peak hours tests are quite close for the data sizes of up to 110 Kbytes. Overall, the throughputs for the off-peak transmissions are more consistent and slightly higher than the throughputs for the peak hour's transmissions.

![Figure 7.18](image)

**Figure 7.18** Throughputs for the mobile tests using the laptop interface with the time of day variation

In addition to the completion time and throughput estimation, the mobile tests with the laptop interface also allows the verification on the total data received by the Telemedicine Application Server to match the total data transmitted by the remote patient for each data size. These observations help to highlight that in the GPRS data transmissions, packets are rarely lost even when link outages occurred, but just grossly delayed. During the mobile tests, all the packets for the medical data transmissions were received by the server. Only the completion time may be affected due to retransmissions of any lost packets before reaching the final destination IP address. Any packet
loss or packer errors may have been rectified by the FEC and ARQ error control techniques over the radio interface and may have significantly increased the completion time. However, the incidence of packet loss does occur over GPRS links in both the downlink and uplink directions, but the mobile tests showed that packet loss is relatively rare or does not occur at all.

7.5 Summary

The mobile tests performed have shown successful transmissions of multi-channel medical data via the GPRS network. The field tests in four different mobile environments reflect a variety of possible patient mobility scenarios. As expected, the performance in the indoor mobile environment had faster completion times and higher throughputs than other mobile environments. The mobile test interfaces and procedures were designed to best estimate the completion times and data throughputs of the GPRS network. The consistency in the measurement techniques throughout the field tests is essential in producing results that reflect the general pattern in the mobile telemedicine system performance. The mobile tests also show consistency with the expected theoretical results in the behaviour, throughputs and data quality of the GPRS network. However, it is acknowledged that more precise and accurate measurements on the GPRS network are possible with the use of commercial and portable air interface test tool solutions for verification, maintenance, and troubleshooting of mobile networks. These tools are normally used by the operation staffs of mobile network operators for networks maintenance and optimisation. For example, the TEMS™ Pocket GSM and the TEMS™ Investigation 6.0 test tools from Ericsson are capable of giving details and precise GPRS traffic performance measurements such as the data throughput for up-link and down-link, the percentage of the data blocks re-sent/erroneously decoded, GPRS status details and other performance parameters [7.25, 7.26].
With the randomness of radio resources availability in the GPRS network, the results obtained in these field tests may vary slightly if all the tests were to be repeated. Due to the flexible allocation of GPRS physical channels, based on the common pool of available resources in a radio cell at any particular time, there is some variation in throughputs of the mobile telemedicine system. The results may also be influenced by non-ideal radio conditions such as traffic congestion, frequency interference, poor radio coverage and weather conditions. The highest average throughput observed in the mobile tests was only between 16 to 20 kbps under the indoor environment, using the two up-link timeslots mobile class, although the transmission with a small data size (about 15 Kbytes) did give a throughput of around 24 kbps. With the limitation of only CS1 and CS2 coding schemes implemented by most GPRS network operators, the uplink throughputs of the mobile telemedicine system can only reach the maximum theoretical data rate of 26.8 kbps.

Although the data throughputs and the transmission delays vary according to the mobile environment of the remote patient, the mobile telemedicine system has significantly demonstrated the real-time possibility with the display of incoming signals at the Telemedicine Application Server. As the IP packet network only offers a 'Best-Effort' service and does not retain the timing information, integration with a Real-Time Transport Protocol may improve the real-time capability, as used in video communications over GPRS [7.27].

With the successful up-link transmission of multi-channel physiological signals via the GPRS network, the mobile telemedicine system has demonstrated the practicality of integrating healthcare delivery with wide area connectivity using a mobile phone. The system also reflects the role of mobile technologies in medical connectivity developments for monitoring and on-the-move m-Health applications [7.1].
7.6 References


[7.3] 3GPP, "Digital cellular telecommunications system (Phase 2+); General Packet Radio Service (GPRS); Overall description of the GPRS radio Interface; Stage 2," 3GPP TS 43.064 version 6.6.0 Release 6, April 2005.


CHAPTER EIGHT: CONCLUSIONS AND FUTURE RESEARCH

8.1 Conclusions

The aim of this research has been to design and implement an integrated m-Health system with an interface processor for the transmission of multi-channel biomedical signals over a Bluetooth link to mobile cellular networks. The design allows uplink transmission of physiological signals from a patient directly to a hospital using a mobile phone on a nationally deployed commercial GPRS network. This is achieved by developing a m-Health processor module that samples, processes and transmits up to four of the most common parameters used in medical monitoring. The patient’s flexibility of use for the processor module is increased by having a Bluetooth interface with a mobile phone. From the foregoing work, combined together with the results from the practical evaluation, the following conclusions may be drawn:

1. A review of the early concept of bio-telemetry leads to the emergence of applications in m-Health. The categorisation of existing m-Health systems is based on the technologies used, namely satellite links, 'short-range'
networks and links, and mobile cellular networks. The bulky and expensive satellite equipment and the limited wireless connectivity of 'short-range' networks create a need for integrating wireless medical sensors with wide area communication gateways. As greater patient mobility gradually becomes a trend in remote monitoring, the integration of medical sensors with global connectivity seems to be the next step in providing telemedicine services. A prototype of a m-Health processor module with Bluetooth interface was therefore produced for transmitting a patient's multi-channel biomedical signals directly to a hospital server via a mobile phone. The processor prototype samples signals from up to four sets of sensors attached to the patient and transmits the digital data over a Bluetooth link to the mobile phone.

2. The wide area connectivity of the system is designed to utilise the up-link transmissions with TCP/IP protocols over the GPRS network. The IP and GPRS protocols adopted in the system implementation are based on the standard specifications operated by the Vodafone GPRS network in the UK. The digitised medical signals are encapsulated into IP packets by the patient interface programs, which are then transformed into GPRS access protocols format by the mobile phone for the transmission to the external packet data network at the hospital server end.

3. The patient user interface program is for the patient to perform the data acquisition process from the processor module. There are two patient user interface programs designed for the system. One is a user interface on a laptop for connection with a fixed-IP network, or with the GPRS network using a GPRS data card or a mobile phone as a modem. The other is a user interface on a mobile phone, which handles the Bluetooth link establishment with the processor module and the GPRS data transmission to the hospital server.

4. The patient user interface program on the mobile phone is based on J2ME MIDlet suite application. The mobile phone compatibility issue of the J2ME MIDlet suite application has been considered during the development stage and therefore the Series 60 platform has been adopted. The
platform allows the J2ME Telemedicine MIDlet suite to be compatible with other Series 60 platform portable devices. The J2ME development platform that has been successfully set up for building the Telemedicine MIDlet suite offers a common base for more application developments on small portable devices.

5. The m-Health system is integrated with client-server application programs to allow the monitoring and management of medical data. The Telemedicine Application Server is responsible for handling the telemedicine session and controlling the client connection request from a remote patient. All the medical data transmitted during a telemedicine session is stored in a database together with the patient information and telemedicine session details. The security features implemented in the m-Health system are essentially on the first security level to ensure a secure connection and to protect data integrity and medical confidentiality of the patient.

6. Other than having the potential of providing greater access to quality medical care irrespective of geographical mobility for the patient, the design of the system also offers flexibility to the clinician. Since the consequences on working conditions for clinicians and healthcare experts for adopting m-Health systems are yet to be assessed, it is essential to minimise the effects of introducing new technology to existing healthcare institutions. These considerations led to the development of doctor browser programs as part of the m-Health system. The browser programs allow clinicians to view and monitor data from telemedicine sessions along with the patient details as and when required, by accessing the database server. The browser programs also offer additional flexibility for the clinicians, since they can be used with either a fixed-network in the hospital or with a mobile network while on the move.

7. The practical implementation of the research helped to investigate the performance of the system as part of future m-Health applications. The prototype system allowed real-world mobile tests to be carried out successfully in demonstrating and evaluating the concept as well as
assessing the effectiveness of using mobile networks for the transmission of multi-channel biomedical signals. The medical data were transmitted from remote locations using the patient interface on a laptop and the patient interface on a mobile phone to the Telemedicine Application Server in the lab. The mobile tests were performed using clinical ECG records from the MIT-BIH arrhythmia database and real multi-channel medical data from a test patient. The successful up-link transmissions via the GPRS network in various patient mobility scenarios provide valuable insights into real user experience with m-Health systems.

The end product of the research highlights the role of mobile technologies in monitoring and on-the-move m-Health applications. The design and implementation in the research contributes to the evaluation of fundamental concepts that could lead to the emergence of future wearable, intelligent and personalised healthcare systems. The practical evaluation of the system attempts to promote a wider user acceptance of mobile telemedicine applications that undoubtedly will be a means of improving the quality of life of patients, particularly for those undergoing post-operative care and living in remote areas, away from a hospital.

Alternatively, the system is technically capable of sampling any type of signal and transmitting it from a remote location, via the cellular network to a server at a main location. The received signals will then be stored for further analysis and expert opinion. These capabilities allow the system to be utilised to its potential in other areas of research or monitoring applications.


8.2 Recommendations

The research work opens up new ideas and the discovery of new problems and limitations. As the research had focused on the hardware realisation and successful implementation of a working overall system, some features and technologies were only implemented on a necessary basis. Various new discoveries during the course of the research work have inspired some interesting future research topics and opportunities.

8.2.1 System Hardware

The design and implementation of the processor module was based on a micro-controller development board. The choice was made in order to rapidly produce a working prototype for proof-of-concept and real world field trial. The design of the processor module can be further improved by designing a dedicated hardware module, possibly using a System-on-Chip (SoC) solution or an Application-Specific Integrated Circuit (ASIC) technology. With dedicated hardware integrated with a Bluetooth capability for short-range wireless connection, the physical size of the processor module could be reduced to a size of a credit card or even down to the size of one’s thumb.

Since the primary use of the processor module would be for remote and portable monitoring, having low power consumption is a necessary criterion. With the introduction of a ZigBee standard for multi-month to multi-year battery life solution, further improvement on the processor module design should utilise the ZigBee standard and optimise the portability feature of the m-Health system.

The design of the patient interface program on the mobile phone was strictly limited to the development platform supported by the mobile phone. The introduction of the J2ME platform on portable and handheld devices is rather
new and can be considered as at an early stage. The research had discovered some limitations on the mobile phone’s software and some reliability issues with the phone’s operating system. These are acknowledged by some mobile phone manufacturers who continuously release a “known issues” document related to their product and software. Since many applications developments with mobile phones focus on entertainment and basic personal management applications, there is limited development on real-time data handling with a communication interface. A mobile phone may have a GPRS function, a Bluetooth link capability and a large graphical display, but a dedicated program is required to utilise and to integrate all these functions, hence it performs a specific process.

One of the issues or limitations discovered was during the deployment of the J2ME MIDlet suite. Some of the user interface options in the MIDlet suite worked well on the Wireless Toolkit emulator but did not work or function properly when the MIDlet suite was installed on the Nokia 6600 mobile phone. Surprisingly, the entire user interface options worked as if it were on the emulator when the same MIDlet suite was installed on a Nokia 6630 mobile phone, a model with the next version of the same mobile phone operating system. Since the deployment of a patient interface program on the mobile phone mainly relies on the development platform of its model, a close collaboration with the mobile phone’s software developer would be advantageous. For example, a collaboration with the Nokia Series 60 developer would allow researchers to have more information on the software limitation or perhaps the sharp way to optimise the existing mobile software development platform.
8.2.2 System Software and Protocols

The research carried out has concentrated on producing a working system that allows the transmission of multi-channel medical data via the GPRS network. Achieving this is a step towards producing a more versatile system that optimises the cellular network behaviour based on the results from the real-world mobile tests.

Integration of data compression and signal processing algorithms, such as wavelet techniques, can help to optimise the performance of the prototype system under different mobile environments. The compression effects on the system performance were accidentally demonstrated during the mobile tests with the commercial GPRS software on the laptop interface. The compression mode allowed the data transmission to be completed in a much shorter time.

Further development on data encryption and subsequent decryption of the medical signals increases the data integrity and the medical confidentiality of the patient. Additional data encryption allows the system to be more secure and robust against any possible security attacks such as eavesdropping, altering, playback, and denial of service.

In discussing network security issues, the main emphasis is on protecting the entire server host network at the receiving end. An enhanced security mechanism would provide a more sophisticated and advanced network protection against unauthorised access and attackers threats. The Telemedicine Application Server and the telemedicine database could be implemented with a more robust and layered protection mechanism of security protocol. This includes the integration of a firewall and a secure shell (ssh) program that provides authentication and secure communications over unsecured channels. A server system that could provide an extendable platform for the server administrators to implement their own additional authentication mechanisms also increases the security level of the system. In
addition, a secure connection interface could also increase the security protection, such as the secure socket stream connection in the J2ME platform. Producing an optimum security implementation with m-Health applications could alone be an interesting research topic.

8.3 Future Research

The concepts and prototype presented in this thesis have potential uses in other areas of research. The ideas and proposal presented here are aimed to highlight such potential.

One direct application is to exploit the system in veterinary medicine. The transmissions of multi-channel biomedical signals from a patient could be directly be applied to biomedical signals from animals for transmissions to veterinary clinics. The system could also be used as mobile monitoring applications in local farmlands.

The concepts, the development platform and the prototype system have also been exploited in proposing a research project to the Engineering and Physical Sciences Research Council (EPSRC) for designing an underwater acoustic monitoring system. The proposed system can be deployed off-shore to transmit the echo-location and vocalisation signals produced by cetaceans (porpoises, dolphins and whales) and other marine sources to a shore-based server via a mobile phone. The underwater acoustic system is also equipped with a Global Positioning System (GPS) unit in tracking these animals. It was also reported in the BBC News in May 2005 that a mobile phone SIM card has been explored to keep track of Dusky dolphins off the coast of Cape Town. The proposed system provides a means of monitoring bioacoustic signals and noise in particular locations, with the bonus that the data can be made available to anyone, anywhere in the world, via cellular networks and the Internet.
The prototype system from the research could also be extended for environmental monitoring on land by adopting a mobile phone to transmit data representing wind speed, air temperature, humidity, and so on, from particular locations.

One useful feature that could be integrated with the prototype is a radio frequency (RF) receiver. This will allow the system to transmit signals from RF based sensors via the cellular network. A potential research collaboration has been initiated with the project on an ingestible capsule with a miniaturised multi-sensor micro-system. The prototype system can then process and further transmit the physiological parameters of the gastro-intestinal tract from the ingestible capsule via the mobile phone.

~ The End ~
PUBLICATIONS


APPENDICES

APPENDIX A - ECG display from the MIT-BIH Arrhythmia Database
APPENDIX B - Software and Development Tools
APPENDIX C - System Hardware Designs
APPENDIX D - Mobile Phone Specifications
APPENDIX E - GPRS AT-Commands
APPENDIX F - J2ME Development Process
APPENDIX G - Countryside Test Locations
APPENDIX A

ECG DISPLAY FROM THE MIT-BIH ARRHYTHMIA DATABASE
APPENDIX B

SOFTWARE AND DEVELOPMENT TOOLS
<table>
<thead>
<tr>
<th>Tool</th>
<th>Version</th>
<th>Website</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATMEL FLexible In-system Programmer (FLIP)</td>
<td>2.4.4</td>
<td><a href="http://www.atmel.com/dyn/products/tools_card.asp">http://www.atmel.com/dyn/products/tools_card.asp</a></td>
</tr>
<tr>
<td>Apache Tomcat server</td>
<td>5.5.7</td>
<td><a href="http://jakarta.apache.org/tomcat/">http://jakarta.apache.org/tomcat/</a></td>
</tr>
<tr>
<td>Eclipse IDE</td>
<td>3.0</td>
<td><a href="http://www.eclipse.org">http://www.eclipse.org</a></td>
</tr>
<tr>
<td>EclipseME Plugin</td>
<td>0.6.1</td>
<td><a href="http://eclipseme.org">http://eclipseme.org</a></td>
</tr>
<tr>
<td>Java 2 Runtime Environment</td>
<td>1.4.2</td>
<td><a href="http://java.sun.com/j2se/desktopjava/jre/index.jsp">http://java.sun.com/j2se/desktopjava/jre/index.jsp</a></td>
</tr>
<tr>
<td>Keil µVision IDE</td>
<td>2.36</td>
<td><a href="http://www.keil.com/uvision2/">http://www.keil.com/uvision2/</a></td>
</tr>
<tr>
<td>Microsoft Visual Basic</td>
<td>6.0</td>
<td><a href="http://msdn.microsoft.com/vbasic/">http://msdn.microsoft.com/vbasic/</a></td>
</tr>
<tr>
<td>MIT-BIH WView</td>
<td>1.03</td>
<td>MIT-BIH Arrhythmia Database CD-ROM <a href="http://ecg.mit.edu/">http://ecg.mit.edu/</a></td>
</tr>
<tr>
<td>MySQL Database Server</td>
<td>4.1</td>
<td><a href="http://www.mysql.com">http://www.mysql.com</a></td>
</tr>
<tr>
<td>MySQL Java Connector</td>
<td>3.0.10</td>
<td><a href="http://www.mysql.com/products/connector/j/">http://www.mysql.com/products/connector/j/</a></td>
</tr>
<tr>
<td>MySQL ODBC Connector</td>
<td>3.51</td>
<td><a href="http://www.mysql.com/products/connector/odbc">http://www.mysql.com/products/connector/odbc</a></td>
</tr>
<tr>
<td>Nokia Connectivity Framework</td>
<td>1.1</td>
<td><a href="http://www.forum.nokia.com">http://www.forum.nokia.com</a></td>
</tr>
<tr>
<td>Nokia Developer Suite</td>
<td>2.2</td>
<td><a href="http://www.forum.nokia.com">http://www.forum.nokia.com</a></td>
</tr>
<tr>
<td>Nokia Mobile Internet Toolkit</td>
<td>4.1</td>
<td><a href="http://www.forum.nokia.com">http://www.forum.nokia.com</a></td>
</tr>
<tr>
<td>Nokia PC Suite</td>
<td>6.41.6</td>
<td><a href="http://www.nokia.co.uk">http://www.nokia.co.uk</a></td>
</tr>
<tr>
<td>Nokia Series 60 J2ME SDK</td>
<td>0.3.1</td>
<td><a href="http://www.forum.nokia.com">http://www.forum.nokia.com</a></td>
</tr>
<tr>
<td>Sun Microsystems Wireless Toolkit</td>
<td>2.0</td>
<td><a href="http://java.sun.com/j2me/">http://java.sun.com/j2me/</a></td>
</tr>
<tr>
<td>Sun Microsystems Wireless Toolkit</td>
<td>2.2</td>
<td><a href="http://java.sun.com/j2me/">http://java.sun.com/j2me/</a></td>
</tr>
<tr>
<td>Vodafone Connect Me Internet Access</td>
<td>1.4</td>
<td><a href="http://www.vodafone.co.uk">http://www.vodafone.co.uk</a></td>
</tr>
</tbody>
</table>
APPENDIX C

SYSTEM HARDWARE DESIGNS
System Hardware Module

Overall board layout of m-Health processor module

Block diagram of Atmel T89C51AC2 microcontroller

Bluetooth communication module PCB
APPENDIX D

MOBILE PHONE SPECIFICATIONS
Nokia 6600 Mobile Phone Specifications

Software Platform:
Series 60 Developer Platform 2.0
Symbian OS 7.0s Operating System

Java Technology:
CLDC 1.0
MIDP 2.0
Nokia User Interface API
Wireless Messaging API (JSR-120)
Mobile Media API (JSR-135)
Bluetooth API (JSR-82)

Screen Display:
Colour Depth: 65536 Colours (16 bit)
Resolution: 176 x 208

Hardware:
ARM9 CPU, 104 MHz
Class B GPRS Mobile Phone
GPRS Multi-slot Class 6 (3+1/2+2 slots)

Local Connectivity:
Infrared
Bluetooth

Physical Measurements:
Dimensions: 109 x 58 x 24 mm
Weight: 125g
APPENDIX E

GPRS AT-COMMANDS
GPRS AT Commands

<table>
<thead>
<tr>
<th>AT Commands</th>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+CGDCONT</td>
<td>Define PDP context</td>
</tr>
<tr>
<td>AT+CGQREQ</td>
<td>Specify a Quality of Service Profile</td>
</tr>
<tr>
<td>AT+CGATT</td>
<td>GPRS attach or detach</td>
</tr>
<tr>
<td>AT+CGATT</td>
<td>PDP context activate or deactivate</td>
</tr>
<tr>
<td>AT+CGDATA</td>
<td>Enter data sate</td>
</tr>
<tr>
<td>AT+CGCLASS</td>
<td>GPRS mobile station class</td>
</tr>
<tr>
<td>AT+CGREG</td>
<td>GPRS network registration status</td>
</tr>
</tbody>
</table>

Some of GPRS AT Commands and its functions

```
+CGATT: 1
OK
```

AT Commands for the GPRS Attachment and PDP Context Activation status

The command AT+CGATT? returns the current GPRS attachment status. Response "+CGATT: 1" means the mobile phone is attached to the network whereas "+CGATT: 0" means it is detached. The AT+CGACT? command returns the current activation state for the PDP context of the mobile phone. The PDP context status response is in the format "+CGACT: <cid>, <state>" where:
**APPENDIX E**

**<cid>:** PDP Context Identifier which specifies a particular PDP context definition

**<state>:** Indicates the state of PDP context activation with 0 = deactivated and 1 = activated

Response "+CGACT: 1,0" means the mobile phone is supporting a PDP context with PDP Context Identifier = 1 and it is currently deactivated.

---

**Mobile Phone Identification Information**

The AT Commands are used with an interface on a HyperTerminal application on a standard desktop environment. The terminal application is configured to communicate with the Nokia 6600 mobile phone via a Bluetooth link.

<table>
<thead>
<tr>
<th>AT Commands</th>
<th>Command Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT + CGMI</td>
<td>To identify the manufacturer of the mobile phone</td>
</tr>
<tr>
<td>AT + CGMM</td>
<td>To identify the specific model of the mobile phone</td>
</tr>
<tr>
<td>AT + CGSN</td>
<td>To identify the individual product serial number identification of the mobile phone, typically the IMEI number</td>
</tr>
<tr>
<td>AT + CINI</td>
<td>To identify the IMSI number based on the SIM card used with the mobile phone</td>
</tr>
</tbody>
</table>

AT Commands to access identification information of the mobile phone
The identification information of the Nokia 6600 mobile phone, together with the IMSI number of the mobile subscriber account used in the research, are shown on the AT Commands interface. The IMEI number of the Nokia 6600 mobile phone is “353383007620191”. The IMSI number for the Vodafone GPRS network used in the research is “234159023009111” where:

Mobile Country Code for the UK = 234
Mobile Network Code for the Vodafone network = 15
Mobile Subscriber Identification Number of the research = 9023009111
APPENDIX F

J2ME DEVELOPMENT PROCESS
J2ME Development Process

There are four major tools or software components needed in establishing a development platform for J2ME applications. The four main tools and software components are a Java 2 RunTime Environment (J2RE) component, an Eclipse Software Development Kit (SDK), a J2ME Wireless Toolkit, and an EclipseME plugin for the Eclipse SDK. All these tools and software components are available as open source from the Sun Microsystems Developer Network website and the Eclipse Foundation website.

The J2RE is a component that provides the Java virtual machine and all the minimum classes needed to run Java 2 programs. The J2RE is a component required in order to run and execute a Java 2 application on a computer. The Eclipse SDK is a development kit that provides a universal platform for application development with multi-language capability. The J2ME Wireless toolkit is a toolbox that creates mobile applications based on CLDC and MIDP specifications. The toolbox provides emulation environments, performance optimisation and tuning features for rapid mobile applications development. The toolkit is integrated with the Eclipse SDK to create MIDlets for J2ME applications. The final main component is the EclipseME plug-in that helps to integrate the wireless toolkit to the Eclipse development environment and performs all the necessary configurations required in Eclipse platform in order to develop J2ME MIDlets.

The first step in setting up the development environment is to install the J2RE. After that the Eclipse SDK is installed with Eclipse SDK version 3.0.1. The next step is to install the J2ME wireless toolkit using the Eclipse platform, which automatically integrates the wireless toolkit features into the Eclipse platform. The last part is to install the EclipseME plug-in via the Eclipse platform, so that the configurations for developing J2ME MIDlets on the platform are performed.
The flow of setting up the J2ME development environment

The flow of building the J2ME MIDlet application

The Java source code for the program is written with a standard text editor and saved as a source file with .java extension in the development environment. The source code is then compiled and pre-verified. The pre-
verify step of the Java class files is specified by CLDC. The integrated emulator on the Eclipse platform is responsible for providing standard mobile phone operations on the desktop development environment. The emulator on the wireless toolkit allows the platform to run the pre-verified MIDlet as emulated J2ME MIDlet and the emulator appears as a generic mobile phone interface. The emulator runs the MIDlet application as it operates on a mobile phone.

![View of wireless toolkit emulator](image)

If the MIDlet runs as expected, the next stage is to package the MIDlet into a Java Archive (JAR) in order to deploy the application into the actual mobile phone. An additional file is also created, known as an Application Descriptor file with .jad extension that describes the contents of MIDlet JAR. Since the
MIDlet is used with the Series 60 platform mobile phone model, the Application Descriptor file shows the CLDC 1.0 configuration and MIDP 2.0 profile that were configured to work with Series 60 Platform. An example of an Application Descriptor file contents is given below:

MIDlet-Version: 1.0.0
MIDlet-Vendor: Midlet Suite Vendor
MIDlet-Jar-URL: TelemedicineMIDlet.jar
MicroEdition-Configuration: CLDC-1.0
MicroEdition-Profile: MIDP-2.0
MIDlet-1: TelemedicineMIDlet,
telemedicineomidlet.TelemedicineMIDlet
MIDlet-Jar-Size: 3557
MIDlet-Name: TelemedicineMIDlet Midlet Suite

The application management software in a mobile phone controls the installation and execution of the MIDlet. The MIDlet is deployed in a package, known as a MIDlet suite, which contains the JAR file and the Application Descriptor file. The final stage is to transfer the MIDlet suite to the mobile phone via a serial cable, an infra red or a Bluetooth link. The MIDlet is installed in the mobile phone by opening the MIDlet JAR and follow the steps guided by the application manager on the mobile phone.

The next figure is a snapshot of a completed MIDlet suite on the Eclipse platform. The package explorer window on the left hand side shows the source code, the J2ME wireless toolkit platform and the MIDlet suite package deployed which includes the .jar and .jad files. The Application Descriptor Editor window shows the MIDlet suite properties that make up the Application Descriptor file. The detail explanation on how to explore the Eclipse platform is available in the Eclipse Java Development user guide.
Snapshot of a completed MIDlet suite on the Eclipse platform
APPENDIX G

COUNTRYSIDE TEST LOCATIONS
**Countryside Test Locations**

Mobile field test locations for the countryside mobile environment

The route for the mobile tests is highlighted with a blue line. The location of the Telemedicine Application Server in the department is indicated by the blue circle with a 'Lab' label. The surroundings of the test locations were farmland, woods and close to local walking paths.