



Technological University Dublin  
**ARROW@TU Dublin**

---

Masters

Engineering

---

2008-01-01

## A Mobile ECG Monitoring System with Context Collection

Jin Peng Li

*Technological University Dublin*

Follow this and additional works at: <https://arrow.tudublin.ie/engmas>



Part of the [Biomedical Engineering and Bioengineering Commons](#), and the [Electrical and Electronics Commons](#)

---

### Recommended Citation

Li, J. (2008). *A mobile ECG monitoring system with context collection*. Masters dissertation. Technological University Dublin. doi:10.21427/D7391M

This Theses, Masters is brought to you for free and open access by the Engineering at ARROW@TU Dublin. It has been accepted for inclusion in Masters by an authorized administrator of ARROW@TU Dublin. For more information, please contact [yvonne.desmond@tudublin.ie](mailto:yvonne.desmond@tudublin.ie), [arrow.admin@tudublin.ie](mailto:arrow.admin@tudublin.ie), [brian.widdis@tudublin.ie](mailto:brian.widdis@tudublin.ie).



This work is licensed under a [Creative Commons Attribution-Noncommercial-Share Alike 3.0 License](#)





# **A Mobile ECG Monitoring System with Context Collection**

**Jin Peng Li**

School of Electrical Engineering System  
Dublin Institute of Technology

A thesis presented to Dublin Institute of Technology  
Faculty of Engineering  
For the degree of  
Master of Philosophy

2008

Supervisors:

Mr. Damon Berry  
Dr. Richard Hayes  
Ms. Gaye Stephens

## ABSTRACT

An objective of a health process is one where patients can stay healthy with the support of expert medical advice when they need it, at any location and any time. An associated aim would be the development of a system which places increased emphasis on preventative measures as a first point of contact with the patient.

This research is a step along the road towards this type of preventative healthcare for cardiac patients. It seeks to develop a smart mobile ECG monitoring system that requests and records context information about what is happening around the subject when an arrhythmia event occurs. Context information about the subject's activities of daily living will, it is hoped, provide an enriched data set for clinicians and so improve clinical decision making. As a first step towards a mobile cardiac wellness guideline system, the focus of this work is to develop a system that can receive bio-signals wirelessly, analyzing and storing the bio-signal in a handheld device and can collect context information when there are significant changes in bio-signs. For this purpose the author will use a low cost development environment to program a state of the art wireless prototype on a handheld computer that detects and responds to changes in the heart rate as calculated from the interval between successive heart beats.

Although the general approach taken in this work could be applied to a wide range of bio-signals, the research will focus on ECG signals. The pieces of the system are,

- A wireless receiver, data collection and storage module.
- An efficient real-time ECG beat detection algorithm.
- A rule-based (Event-Condition-Action) interactive system.
- A simple user interface, which can request additional information from the user.

A selection of real-time ECG detection algorithms have been investigated and one algorithm was implemented in MATLAB [110] and then in Java [142] for this project. In order to collect ECG signals (and in principle any signals), the generalised data collection architecture has also been developed utilizing Java [142] and Bluetooth [5] technology. This architecture uses an implementation of the abstract factory pattern [91] to ensure that the communication channel can be changed conveniently. Another core part of this project is a “wellness” guideline based on Event-Condition-Action (E-C-A) [68] production rule approach that originated in active databases. The work also focuses on design of a guideline based expert system which an E-C-A based implementation will be fully event driven using the Java programming language.

Based on the author's experience and the literature review, some important issues in mobile healthcare along with the corresponding reasons, consequences and possible solutions will be presented.

## **DECLARATION**

I certify that this thesis which I now submit for examination for the award of Master of Philosophy in Electrical / Electronic Engineering, is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

This thesis was prepared according to the regulations for postgraduate study by research of the Dublin Institute of Technology and has not been submitted in whole or in part for an award in any other Institute or University.

The work reported on in this thesis conforms to the principles and requirements of the Institute's guidelines for ethics in research.

The Institute has permission to keep, to lend or to copy this thesis in whole or in part, on condition that any such use of the material of the thesis be duly acknowledged.

Signature: \_\_\_\_\_

Date: \_\_\_\_\_

# TABLE OF CONTENTS

<b>CHAPTER 1 INTRODUCTION.....</b>	<b>1</b>
1.1 CORE CONTEXT INFORMATION .....	1
1.2 HOLTER MONITORING – AMBULATORY ECG MONITORING .....	4
1.2.1 <i>Current state of Holter monitoring technology.....</i>	<i>5</i>
1.2.2 <i>The approach taken within this work – Local analysis with context awareness .....</i>	<i>6</i>
1.3 PERSONAL AREA NETWORK .....	6
1.4 OBJECTIVES AND RESEARCH QUESTIONS .....	9
1.5 OVERVIEW OF THE REMAINING CHAPTERS.....	10
<b>CHAPTER 2 MEASURING CARDIAC SIGNALS AND OTHER TOPICS .....</b>	<b>12</b>
2.1 THE ELECTRICAL ACTIVITY IN THE HEART AND THE ELECTROCARDIOGRAM.....	12
2.1.1 <i>The human heart .....</i>	<i>12</i>
2.1.2 <i>The conduction system of the heart.....</i>	<i>14</i>
2.1.3 <i>Electrocardiogram - ECG.....</i>	<i>16</i>
2.1.4 <i>QRS complex.....</i>	<i>17</i>
2.2 CURRENT HOLTER MONITORING SYSTEMS.....	18
2.2.1 <i>Type one – Dumb Holter .....</i>	<i>19</i>
2.2.2 <i>Type two – Cardiac Event Monitor - Recording with remote classification.....</i>	<i>20</i>
2.2.3 <i>Type three - Recording plus local ECG classification but no ADL context awareness .....</i>	<i>21</i>
2.2.4 <i>Type four ADL context awareness through accelerometers .....</i>	<i>21</i>
2.3 BLUETOOTH.....	23

2.4 EVENT – CONDITION – ACTION (E-C-A) .....	25
2.5 KVM.....	26
2.6 JAVA ARCHIVE FILE (JAR) .....	26
2.7 FINAL REMARKS.....	27
<b>CHAPTER 3 METHODOLOGY AND OVERVIEW OF PROPOSED SYSTEM.....</b>	<b>28</b>
3.1 INTRODUCTION TO THIS CHAPTER.....	28
3.1.1 System constraints that are imposed by the signals .....	28
3.1.2 System constraints relating to the wireless transmission .....	29
3.1.3 Platform constraints.....	29
3.2 METHODOLOGIES USED IN THIS PROJECT .....	29
3.2.1 Open Distributed Processing and its use in this dissertation.....	29
3.2.2 Unified Modeling Language .....	31
3.2.3 Throwaway Prototyping – System development process .....	32
3.4 ENTERPRISE VIEWPOINT OF THE COMPONENTS OF THE SYSTEM .....	32
3.4.1 Data transportation module.....	35
3.4.2 ECG beat detection module .....	37
3.4.3 E-C-A Event processing module for Context information collection .....	38
3.4.4 Bluetooth enabled Electrocardiograph Simulator for a PC.....	39
3.5 HOW THE SYSTEM WORKS – AN ENTERPRISE VIEW .....	40
3.5.1 From transmitted ECG sample to sample for beat detection .....	41
3.5.2 From sample in algorithm to beat event .....	43
3.5.3 From beat event to action .....	43

3.7 TECHNOLOGY VIEWPOINT – TECHNOLOGY CHOICES .....	44
3.7.1 Choice of Development and Testing Platform.....	45
3.7.2 Choice of wireless communication medium between electrocardiograph and PDA.....	49
3.7.3 Other choices affecting wireless sample transmission .....	50
3.7.4 Choice of beat detection algorithm.....	51
3.7.5 Java event model for implementation of E-C-A .....	52
<b>CHAPTER 4 ANALYSIS .....</b>	<b>53</b>
4.1 A SCENARIO OF USE FOR BODY AREA NETWORK FOR INTELLIGENT AMBULATORY ECG RECORDING .....	53
4.2 SYSTEM DESIGN ENVIRONMENT .....	55
4.3 SYSTEM DESIGN PROTOCOL – SAECA .....	56
4.4 SYSTEM DESIGN – DATA TRANSPORTATION .....	58
4.4.1 Transportation structure .....	58
4.4.2 Dealing with ECG data.....	60
4.4.2.1 Data resolution .....	60
4.4.2.2 Data transfer speed.....	61
4.4.2.3 Data accuracy .....	62
4.4.3 ECG data transportation process .....	63
4.4.3.1 Phase one – sending .....	64
4.4.3.2 Phase two – receiving.....	64
<b>CHAPTER 5 ALGORITHM .....</b>	<b>66</b>

5.1 INTRODUCTION .....	66
5.2 ZERO – CROSSING ALGORITHM INTRODUCTION.....	68
5.3 ALGORITHM ILLUSTRATION.....	70
5.4 FLOWCHART OF THE ALGORITHM FOR JAVA IMPLEMENTATION .....	81
<b>CHAPTER 6 DATA TRANSPORTATION AND CONTEXT COLLECTION.....</b>	<b>83</b>
6.1 DATA TRANSPORTATION .....	83
6.1.1 Simulator - sending .....	83
6.1.2 PDA - receiving.....	86
6.2 CONTEXT REVIEWED .....	89
6.3 CONTEXT COLLECTION – GENERAL STRUCTURE .....	89
6.4 CONTEXT INFORMATION COLLECTION .....	91
6.4.1 Event process .....	91
6.4.2 Condition evaluation.....	94
6.4.3 User interface.....	95
<b>CHAPTER 7 SYSTEM TESTING.....</b>	<b>99</b>
7.1 SYSTEM SPECIFICATION .....	99
7.2 A BRIEF EXPLANATION OF UNIT TESTING .....	100
7.3 UNIT TESTING OF DATA TRANSPORTATION MODULE.....	100
7.3.1 Introduction.....	100
7.3.2 Unit testing detail on data transportation module .....	101
7.4 UNIT TESTING OF BEAT DETECTION ALGORITHM.....	105



7.4.1 Comparison between Java and MATLAB .....	106
7.4.2 Unit testing of beat detection algorithm in PDA.....	110
7.5 UNIT TESTING OF EVENT BASED CONTEXT COLLECTION .....	112
7.5.1 Outfit of the interface window.....	113
7.5.2 Memory usage of the interface window.....	114
7.6 INTEGRATION TESTING.....	116
<b>CHAPTER 8 RESEARCH EVALUATION .....</b>	<b>120</b>
8.1 PDA PERFORMANCE .....	120
8.2 ECG DATA RECORDING.....	122
8.3 ECG ALGORITHM AND CONTEXT COLLECTION .....	123
8.4 MINIMUM HARDWARE SPECIFICATIONS .....	125
8.5 CONCLUSION.....	125
8.6 FUTURE WORK.....	126
8.6.1 DATA COMMUNICATION ENHANCEMENTS .....	127
8.6.2 ALGORITHM.....	128
8.6.3 CONTEXT COLLECTION (E-C-A) .....	128
<b>REFERENCE .....</b>	<b>130</b>
<b>GLOSSARY .....</b>	<b>147</b>
<b>APPENDIX A DETECTION ALGORITHMS.....</b>	<b>154</b>
A.1 ALFONSO FILTER BANKS.....	154
A.2 DIFFERENTIATION TECHNIQUE .....	161

A.3 PAN AND TOMPKINS .....	166
<b>APPENDIX B JAVA CODING ALGORITHM.....</b>	<b>178</b>
<b>APPENDIX C TECHNICAL DESCRIPTION OF THE HARDWARE AND SOFTWARE</b>	
<b>PLATFORM HP IPAQ 5550 AND CREME KVM.....</b>	<b>183</b>
C.1 THE IPAQ 5550 PERSONAL DIGITAL ASSISTANT .....	183
C.2 CREME KVM.....	185
<b>APPENDIX D JAVA API.....</b>	<b>188</b>
<b>APPENDIX E MIND MAP OF CONCEPTS .....</b>	<b>193</b>
<b>APPENDIX F OPEN DISTRIBUTED PROCESSING (ODP) .....</b>	<b>196</b>
<b>APPENDIX G JSR 82 .....</b>	<b>197</b>
<b>APPENDIX H TESTING ENVIROMENT .....</b>	<b>200</b>
<b>APPENDIX I UML DIAGRAMS.....</b>	<b>202</b>
I.1 CLASS DIAGRAM .....	202
I.2 SEQUENCE DIAGRAM .....	203
<b>APPENDIX J BLUETOOTH ISSUSE IN HP IPAQ .....</b>	<b>205</b>
<b>APPENDIX K FINITE IMPULSE RESPONSE FILTER .....</b>	<b>207</b>

# TABLE OF FIGURES

FIGURE 1: OVERVIEW OF THE SYSTEM .....	9
FIGURE 2: BLOOD FLOW THROUGH THE HEART .....	13
FIGURE 3: THE CONDUCTION SYSTEM OF THE HEART .....	14
FIGURE 4: A COMPLETE CYCLE OF A HEART BEAT IN ECG FORM WITH INDICATIONS OF THE ACCOMPANYING ELECTRICAL CHANGES .....	16
FIGURE 5: DIFFERENT POINTS FOR ECG DETECTION .....	17
FIGURE 6: QRS COMPLEX, A PORTION OF ECG .....	18
FIGURE 7: THE PROPOSED SYSTEM IN ACTUAL USE INCLUDING BLUETOOTH ENABLED ECG MEASUREMENT DEVICE .....	33
FIGURE 8: THE PROPOSED SYSTEM WITH THE SIMULATOR .....	34
FIGURE 9: ECG MEASUREMENTS FROM BULKY TO PORTABLE.....	36
FIGURE 10: ZERO-CROSSING ALGORITHM OVERVIEW .....	38
FIGURE 11: OVERVIEW OF E-C-A EVENT PROCESSING MODULE .....	39
FIGURE 12: EVENT PROCESSING INTERACTION SYSTEM.....	40
FIGURE 13: BLUETOOTHMANAGER AND SERIAL PORT EVENT LISTENER .....	42
FIGURE 14: ALGORITHM AND DATA AVAILABLE LISTENER .....	43
FIGURE 15: BEAT EVENT PROCESSING AND CONTEXT INFORMATION COLLECTION.....	44
FIGURE 16: UML SEQUENCE DIAGRAM OF THE DESIGN PROTOCOL – SAECA .....	56
FIGURE 17: UML CLASS DIAGRAM FOR THE TRANSPORTATION STRUCTURE.....	59
FIGURE 18: AN OUTLINE OF THE TYPE OF ECG FRAME USED IN THIS WORK.....	62
FIGURE 19: ECG DATA PARSING AND FRAME INFORMATION INSERTION .....	64

FIGURE 20: ECG DATA RECEIVING PROCESS IN PDA .....	65
FIGURE 21: THE SIGNALS IN THE FEATURE EXTRACTION STAGE.....	69
FIGURE 22: A PORTION OF ECG SIGNAL THAT CONTAINS BASELINE DRIFT AND NOISE .....	70
FIGURE 23: A 26 <sup>TH</sup> ORDER LINEAR PHASE FIR FILTER .....	71
FIGURE 24: COMPARISON BETWEEN FILTERED AND ORIGINAL ECG SIGNAL .....	72
FIGURE 25: COMPARISON BETWEEN FILTERED AND NON-LINEAR FILTERED ECG SIGNAL.....	73
FIGURE 26: ADD HIGH FREQUENCY SEQUENCE INTO THE SIGNAL .....	76
FIGURE 27: FEATURE SIGNAL WITH MULTIPLE BEAT EVENTS DETECTED.....	77
FIGURE 28: THRESHOLD AND MULTIPLE-EVENT DETECTION .....	78
FIGURE 29: ALGORITHM OVERVIEW .....	80
FIGURE 30: FLOWCHART OF THE ZERO-CROSSING ALGORITHM.....	81
FIGURE 31: SIMULATOR WORK PROCESS.....	85
FIGURE 32: THE WORK OF DATA RECEIVER.....	88
FIGURE 33: CONTEXT COLLECTION PROCESS.....	90
FIGURE 34: A FLOWCHART OF THE CONTEXT COLLECTION PROCESS .....	91
FIGURE 35: EVENT PROCESSING .....	92
FIGURE 36: CONDITION EVALUATION.....	94
FIGURE 37: USER INTERFACE.....	96
FIGURE 38: INFORMATION LOGGER .....	97
FIGURE 39: LABORATORY ROOM .....	102
FIGURE 40: THE CAMPUS.....	103
FIGURE 41: ALONG THE HALL INSIDE THE BUILDING .....	104

FIGURE 42: THE OUTPUT OF THE FILTERING PROCESS .....	107
FIGURE 43: THE OUTPUT OF THE NON-LINEAR PROCESS .....	108
FIGURE 44: COMPARISON OF THE OUTPUTS OF THE HIGH FREQUENCY SEQUENCE ADDITION PROCESS ...	108
FIGURE 45: THE OUTPUT OF THE FEATURE SIGNAL EXTRACTION PROCESS.....	109
FIGURE 46: THE OUTPUT OF THE THRESHOLD DETERMINATION PROCESS.....	109
FIGURE 47: BEAT IS DETECTED AT THE SAME LOCATION AT 382 .....	110
FIGURE 48: THE MEMORY USAGE WHILE ANALYSING 100 SECONDS ECG DATA.....	111
FIGURE 49: THE DETECTED BEATS FOR 100 SECONDS OF ECG DATA .....	112
FIGURE 50: A USER INTERFACE – POP UP WINDOW APPROACH.....	113
FIGURE 51: INTERFACE WINDOW FOR 1 SUSPICIOUS BEAT INTERVAL.....	115
FIGURE 52: A LOG FILE FOR 1 UNUSUAL BEAT INTERVAL DURATION.....	115
FIGURE 53: MANUALLY INPUT EXTRA INFORMATION INTO THE LOG FILE.....	116

## TABLE OF TABLES

TABLE 1: KVM COMPARISON .....	48
TABLE 2: CHOSEN DEVELOPMENT PLATFORM AND TOOLS.....	49
TABLE 3: DETECTION ALGORITHM COMPARISONS.....	67
TABLE 4: THE ROLES OF OBJECTS IN THE SIMULATOR .....	84
TABLE 5: TEST RESULT OF THE DATA COMMUNICATION.....	105
TABLE 6: INTEGRATION TESTING RESULT - FAILED.....	117
TABLE 7: INTEGRATION TESTING RESULT – SUCCESSFUL.....	118

# CHAPTER 1

## INTRODUCTION

“*No man is infallible*”, people make mistakes, but why do they make mistakes? Very often the reason is that they do not have all of the relevant information. Consider what would happen when a student smoking in a bathroom in college triggers a fire alarm. Because this is not a life-threatening fire event, evacuation is not necessary. If the health and safety person is aware of the cause of the alarm, a time-consuming evacuation can be avoided. In general we can say that the context information that accompanies an event can in many cases have a great impact on the outcome of how the event is processed. This is true for the fire alarm scenario presented above. It is also true for the collection and use of health information. *Context* information is required to add meaning to health measurements and allow them to be used as the basis for clinical judgement.

### 1.1 CORE CONTEXT INFORMATION

“Context” means “*the set of facts or circumstances that surround a situation or event*”-WordNet [7]. In the example above, the event is that the fire alarm is triggered, and the context information is that a student is smoking in a bathroom. So what sort of context information is required to support a health measurement?

The core supporting context for a health measurement value can include the units of

measurement, the timestamp, the frequency and accuracy of the measurement, the identity and health status of the subject of care (patient), the identity of the measuring device and the location of the patient among other properties. This core context information is usually static for the duration of the measurement and is required in order that the measurement value can be used safely. Another type of context information relates to the patient and can change during the course of the measurement. This type of context information is associated with the patient's activities of daily living (ADLs) - a set of activities (e.g. eating, sleeping, walking, stairs climbing, meeting, talking) that people carry out in daily living [8].

In the case of the fire alarm scenario described above, information relating to the time duration and location of the fire alarm are core context material. The fact that a student smoking in a bathroom triggered the alarm can be considered as the dynamic context information. Taking another example from the health domain, when a person is ascending a long staircase, their heart rate will rise. If the person's heart rate is monitored while they are carrying out this activity of daily living, and the context information about stairs climbing is not noted, the monitoring system may misinterpret the raised heart rate as being a matter of concern so it will be useful for the ECG system to be aware of the ADL context and react to it. As Jones [9] suggests *“Timely processing and transmission of such multimedia clinical data in a distributed mobile environment requires smart strategies.”* Clearly the most effective smart strategies are the well-informed ones and so context awareness becomes important.



Much of the available research in health context-awareness focuses on available resources (e.g., type of network and device characteristics), user preferences, user environment (including location) and situation [9,10]. For instance, Kim [11] introduced a context-aware Traveler Healthcare Service (THS) system, which can supply pertinent context information such as location of the patient, sex, age or important medical history to avoid unnecessary additional emergency treatment or improper treatment because of insufficient personal health information. A number of systems [12,13] have attempted to use *accelerometers* to gather ADL context information without direct human input. An accelerometer is a MEMS sensor which measures acceleration and can be used to measure human motion [14]. MEMS - *micro electro mechanical sensor* technology is the integration of mechanical elements, sensors, actuators, and electronics on a common substrate through the utilization of microfabrication technology or “microtechnology” [15]. MEMS accelerometer based approaches for ADL context collection have the advantage that the user is not required to interact directly with the monitoring system. However, they will not be able to detect ADL activities that involve little movement, nor will they be able to accurately discriminate between different types of motion.

Health professionals routinely refer to the ability or inability to perform activities of daily living as a measurement of the functional status of a person [16]. This is not surprising as ADLs will continuously influence measurements whether they are biosignals such as ECGs, or biochemistry readings such as blood glucose. It is the

author's opinion that some attempt should be made to collect this useful information as an ambulatory ECG is being recorded.

This project aims to present a prototype mobile expert system that not only detects the presence of abnormal heartbeat intervals but also requests context information about the activities undertaken by the subject upon detecting changed heart rate. In this way, the recorded biosignals can be related to an additional record of current activity of the subject of care. This additional contextual information can then be used to supplement and enrich the original biosignal record. This research uses human computer interaction to support activity context awareness in ambulatory ECG monitoring within a personal area network [17]. Let us now consider ambulatory ECG monitoring.

## **1.2 HOLTER MONITORING – AMBULATORY ECG MONITORING**

Many heart problems become noticeable only during activity, such as exercise, eating, sex, stress, bowel movements, or even sleeping [18]. Therefore ambulatory electrocardiography (ECG) has been introduced as a diagnostic tool [19] to deal with those health crises.

Ambulatory electrocardiography is also called Holter monitoring, ambulatory ECG or ambulatory EKG. For Holter monitoring, a patient wears a small recorder called a

Holter monitor as he or she goes about normal daily life [21,8,9,20].

Ambulatory ECG is mainly used to document and describe abnormal electrical activity in the heart. This can be random, spontaneous, sleep-related or caused by emotion or stress [22]. The capture and association of symptoms with disturbances in rhythm during daily activity, requires a record of the heart's electrical activity during that time [24,23,21]. This must be done continuously over time as a person goes about normal daily activities [18,25].

There are two basic types of ECG recording devices: continuous and intermittent recorders. Continuous recorders [26,27,28] monitor about 100,000 heartbeats in 24 hours and are likely to find any heart problems that happen with activity. Intermittent recorders [29,30] are used over periods of weeks or even months. During this time they provide brief, intermittent recordings. They are used when symptoms of an abnormal heart rhythm do not occur very often. An intermittent recorder can be used for a longer time than a continuous recorder. The information collected by an intermittent recorder can often be sent over the phone to a doctor's office, clinic, or hospital [31,27].

### **1.2.1 Current state of Holter monitoring technology**

As we shall see in the next chapter, some of the more sophisticated contemporary Holter monitoring systems can record and respond to ECG signals locally [27,32],

These systems in some cases incorporate an ECG analysis algorithm implemented in an embedded system or PDA that detects any abnormal heart activities and sends an alarm to the user and the remote monitoring centre. The workload at the remote monitoring centre could be reduced by introducing the context awareness into the Holter monitoring system.

### **1.2.2 The approach taken within this work – Local analysis with context awareness**

Some contemporary systems [27,32] also analyse the ECG locally and inform the patient about symptoms, thus allowing the patient to deal with the problem while issuing an alarm message to a control centre so that cardiac specialists can review the recorded abnormalities. These current systems [27,20,32] are ‘not aware of’ the daily activity context of the patient when an event is detected, only the ECG data. It is intended that the expert system proposed by the author will as much as possible facilitate better informed healthcare by collecting context information within a *personal area network*, which is introduced in the next section.

## **1.3 PERSONAL AREA NETWORK**

New terminology appears regularly in the technology domain, and wireless communications is no exception. One of the relatively recent additions is PAN, or Personal Area Network [33]. “A *personal area network (PAN)* is a computer network used for communication among computer devices (including telephones and personal digital assistants) close to one person” [33]. PANs can be used for communication

between personal devices (intrapersonal communication) over short ranges, or for connecting or *up-linking* [34] to a higher level network and the Internet [35]. A wireless personal area network (WPAN) can be realized with network technologies such as Zigbee [36], Bluetooth [5], IrDA [140] and Wi-Fi [37].

Clearly, at the time of writing, there are a number of candidate wireless technologies that could be used as the communications medium between an ECG device and the PDA based application that is the subject of this work. The choice is made easier by the fact that the author's work follows previous research at the School of Electrical Engineering Systems at DIT that was carried out by Cronin [6] where Bluetooth was chosen as a transmission medium in medical scenarios such as ambulatory ECG analysis and diagnosis in a body area network. Nevertheless, some of the available wireless transmission options at time of writing are compared below.

The HP iPAQ H5550 *Personal Digital Assistant* (PDA) employed in this research supports WLAN (802.11b), IrDA and Bluetooth communication. In the author's opinion, The most feasible alternative medium for the wireless transmission in this research is Zigbee [38], and it is often applied in large-scale Wireless Sensor Network applications [39] because of its low power consumption characteristic [40]. However, ECG devices so far seem to have moved towards Bluetooth as the preferred communication medium [41]. The Bluetooth enabled ECG monitors referred to above process medical signals that require a high level of security. Some level of security

has been implemented by the different vendors within the Bluetooth SIG [10]. In contrast, Zigbee has no standard security implementation. [36] For this reason, Bluetooth is preferred by the author to Zigbee in this particular application. At the time of writing Zigbee has not found widespread support in PDA. However, it has some purported advantages including lower power requirements [6] than the other options and so it may only be a matter of time before it is as common as the other wireless technologies. Zigbee [36] has been designed specifically for discrete data that is sent occasionally. Hence the transportation of continuously sampled ECG data from a Holter device would be likely to drain the wireless module's battery. To complete a packet transfer, multiple retransmissions can be expected on occasion [42]. This would also be a major concern for medical applications that necessitate reliability and especially vital signals like ECG that could be time critical.

IrDA [43] is not a suitable option, as it requires line-of-sight for transmission. IrDA supports only point-to-point, narrow angle ( $30^\circ$ ); this greatly restricts the amount of applications where this mode of transmission is feasible.

Wi-Fi [37] is not really a suitable selection for this application. Patient privacy is a big issue; a patient might feel self conscious if the measuring device is too bulky. Wi-Fi consumes a significant level of power. Wi-Fi might be employed once a block of medical data had been collected and needs to be transported via the Internet to an online patient database or porthole.

A Holter type instrument uses either two or three leads to measure the ECG signal in a continuous manner but only transmits brief snapshots of ECG recording (e.g. 5 to 300 seconds) at predefined regular intervals [44]. Although Bluetooth has shortcomings such as discovery time (10.24 seconds [6]) and data packet re-transmission, its plug-and-play capabilities, lower power consumption and reasonable bandwidth make it the most suitable medium for the personal area network application [6]. Therefore a Bluetooth-enabled Holter with the context collection that is described in section 1.2.2 corresponds to the system overview in Figure 1.

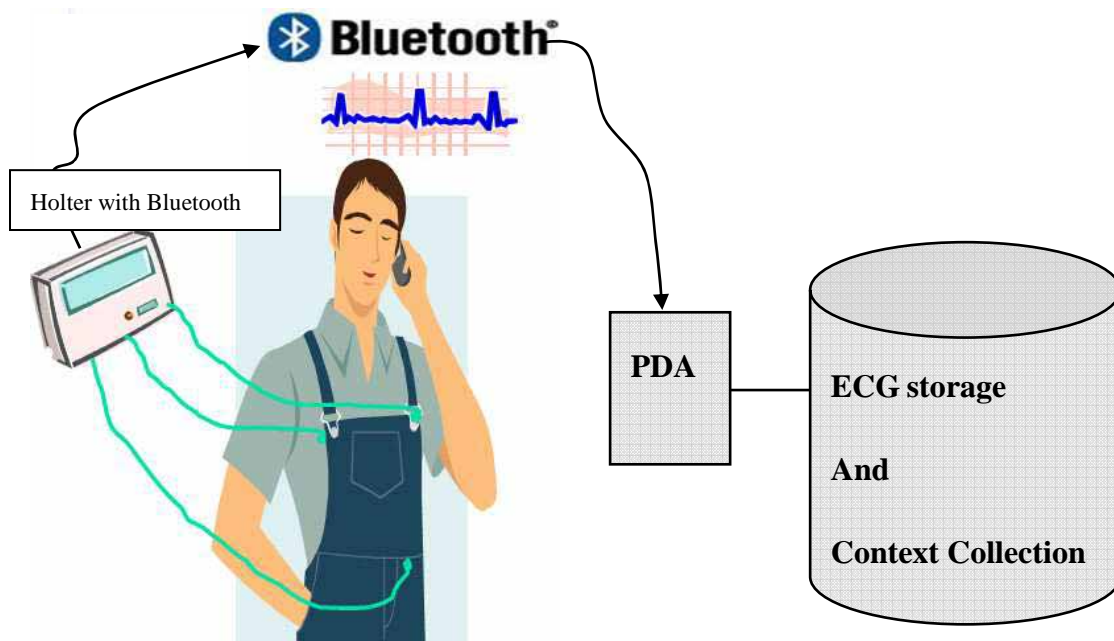


Figure 1: Overview of the system

## 1.4 OBJECTIVES AND RESEARCH QUESTIONS

The main objective is the construction and analysis of a PDA-based prototype interactive expert system for ambulatory ECG beat detection. In particular the author

will investigate whether a memory limited device such as a modern mobile phone or (PDA) can collect the context information while recording and analysing the wirelessly received ECG signals.

This research is guided by the following questions:

*Q 1: Can a resource limited device such as a PDA receive, analyse and react to ECG signals while simultaneously requesting context information from the user in an acceptable manner?*

*Q2: How much ECG data is it reasonable to expect a PDA to record for future examination?*

*Q 3: Can real-time ECG analysis algorithms be implemented successfully on a memory limited platform while simultaneously requesting context information from the user?*

*Q 4: What are the minimum hardware specifications that will give acceptable performance?*

The author has now given an introduction to the work; the remainder of this dissertation is laid out as follows.

## **1.5 OVERVIEW OF THE REMAINING CHAPTERS**

CHAPTER 2 MEASURING CARDIAC SIGNALS briefly describes the working of



the human heart and the creation and measurement of ECGs.

### CHAPTER 3 METHODOLOGY AND OVERVIEW OF PROPOSED SYSTEM

reveals the motivations, rationale, aims and scope of the work as well as providing a short explanation of some of the methodologies used.

CHAPTER 4 ANALYSIS AND DESIGN describes the analysis and design aspects of the work.

CHAPTER 5 BEAT DETECTION ALGORITHM describes the working of the zero cross beat detection algorithm.

CHAPTER 6 CONTEXT COLLECTION details the working of the event processing part of the application using the E-C-A approach.

CHAPTER 7 SYSTEM TESTING recounts a number of tests that were performed on the system.

CHAPTER 8 RESEARCH EVALUATION gives a summation with the authors final thoughts on the project outcomes, the major issues encountered and suggested future work.

## **CHAPTER 2**

# **MEASURING CARDIAC SIGNALS AND OTHER TOPICS**

The first chapter provided a rationale for this project and described the aims and scope of the work. It also introduced some of the basic health concepts underpinning this work. This chapter will provide an introduction to the working of the human heart and the electrical signals that accompany cardiac activity. The state of the art in Holter monitoring systems will be compared, introducing their characteristics, requirements and setup. In order to help the reader better understand the general description of the system in chapter three, the end of this chapter includes preliminary descriptions of some of the technical concepts that are used repeatedly in this work.

## **2.1 THE ELECTRICAL ACTIVITY IN THE HEART AND THE ELECTROCARDIOGRAM**

We have already introduced some of the basic terms that are used in this work in order to explain it at a high level. However, a number of additional topics from the medical and information technology domains should be introduced before moving on to the system overview in Chapter 3. The first of these topics is flow of blood and the corresponding electrical activity within the heart.

### **2.1.1 The human heart**

The heart is composed of four chambers, two upper chambers, the right and left atria,

and two lower ones, the right and left ventricles. The blood flow through the heart is shown in Figure 2.

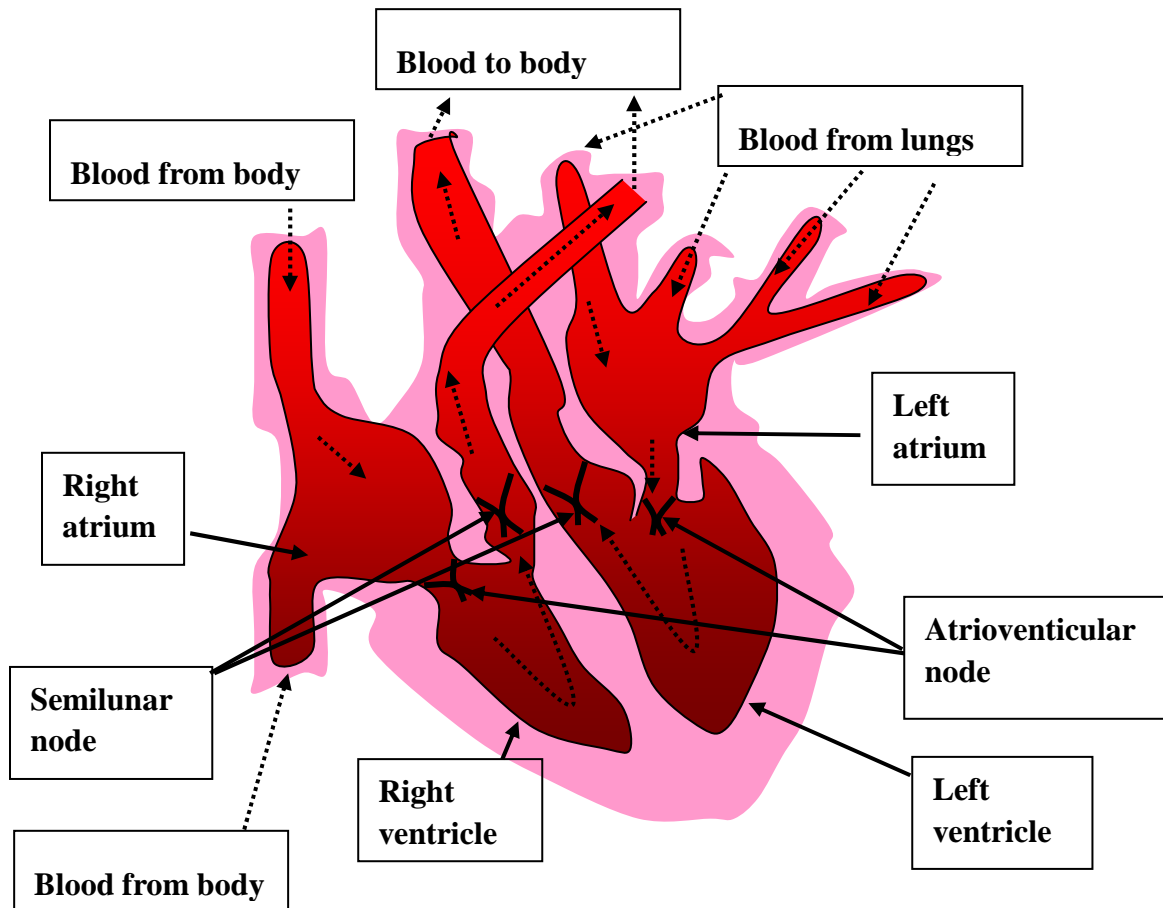


Figure 2: Blood flow through the heart [45]

The blood flows into the heart's upper chambers from the body and lungs, and then the atrioventricular nodes are opened by the contraction of the atria. The blood flows into the ventricles, when the ventricles are full of blood, the atrioventricular nodes are closed and the semilunar nodes are opened by the contraction of the ventricles and then the blood is pumped back into the body and lungs [45].

The average resting heart rate is 60 to 100 beats per minute [1]. In general, if a person

experiences a heart rate of less than 60 beats per minute the person may suffer from bradycardia [2]; If an adult experiences a heart rate of between 100 to 400 beats per minute the patient may suffer from tachyarrhythmia [3]. Based on this fact, four personalized rules for normal heart rate boundaries are created in Chapter 6.

### 2.1.2 The conduction system of the heart

The conduction starts from the *SA node* (Sinus node) - the dominant natural pacemaker of the heart, which is located in the upper part of the wall of the right atrium (the right upper chamber of the heart). A depolarization wave is initiated here and spreads out so that both atria contract and the resulting electrical depolarization and the simultaneous contraction of the atria are represented by the relatively shallow P wave in the Electrocardiogram (ECG).

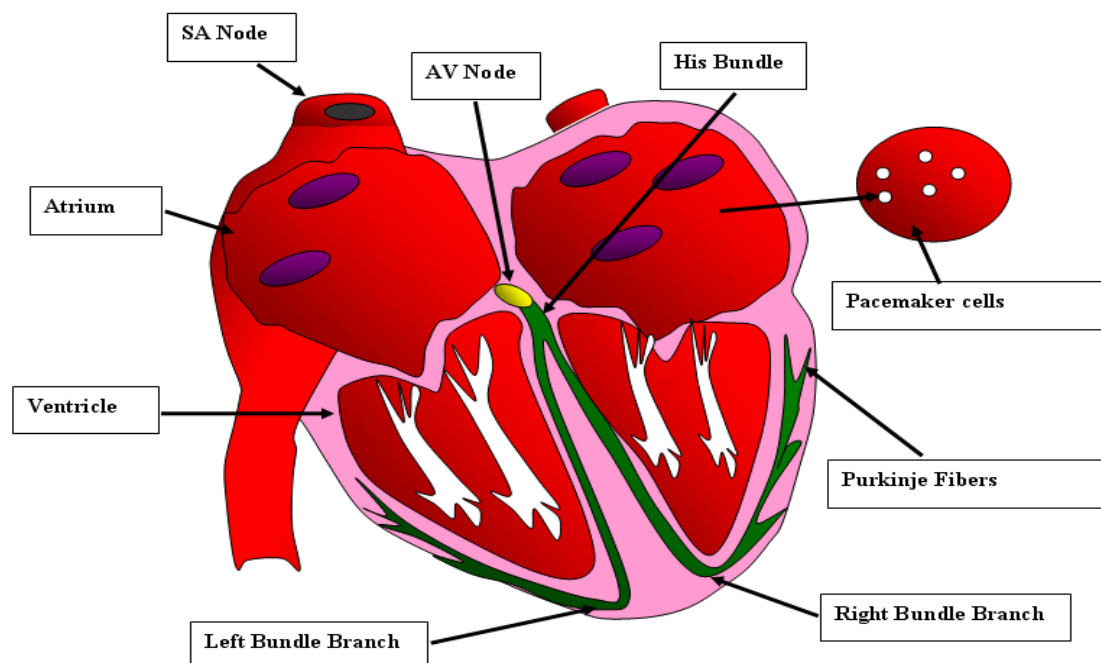


Figure 3: The conduction system of the heart [1]

The atria and the ventricle are separated by the AV node (Atrio-ventricular node), which serves as a “gateway” that impedes the electrical current before the electrical signal is permitted to pass down through to the ventricles. This happens in such a way that a brief pause in electrical activity occurs, which leaves a short piece of flat baseline after each P wave in an ECG. This pause also ensures that the atria have a chance to fully contract before the ventricles are stimulated. Next the depolarizing wave passes slowly through the AV node and then rapidly shoots throughout the bundle of HIS (explained in Glossary) and the Left and the Right Bundle Branches and their subdivisions.

The terminal filaments of the Purkinje Fibers rapidly distribute depolarization to the ventricular myocytes. Depolarization of the entire ventricular myocardium produces a surge in electrical activity which is observed as a QRS complex portion of the ECG. Within 75ms of leaving the AV node, the contraction signal has been transmitted to all of the ventricular cardiac muscle cells. Then the ventricular repolarization occurs to make the heart ready for the next contraction and it is represented by the ST segment and T wave on ECG.

Now the heart has completed one full heart beat, which includes P, Q, R, S, T waves and ST segment. A heart beat is divided into two phases – “diastole” and “systole”. During the first phase (diastole), the heart relaxes and fills with blood (ST segment and T wave). This results in a period of low *diastolic* pressure for the circulatory

system. During the second phase (systole), the heart contracts and pumps out the blood (P, Q, R, S wave). This second period corresponds to a period of high *systolic* pressure in the circulatory system [46]. The heart typically spends about 5/8 of its time in diastole and 3/8 of its time in systole [47]. Keeping this activity well timed is the job of the heart's conduction system.

### 2.1.3 Electrocardiogram - ECG

The term Electrocardiogram (which is also called ECG or EKG) is abbreviated from the German word “Elektrokardio-gramm”, which consists of a graphical representation of the myocardial depolarization and repolarization [1]. Every normal cardiac cycle (heart beat) is composed of P, Q, R, S, T waves, as depicted in Figure 4,

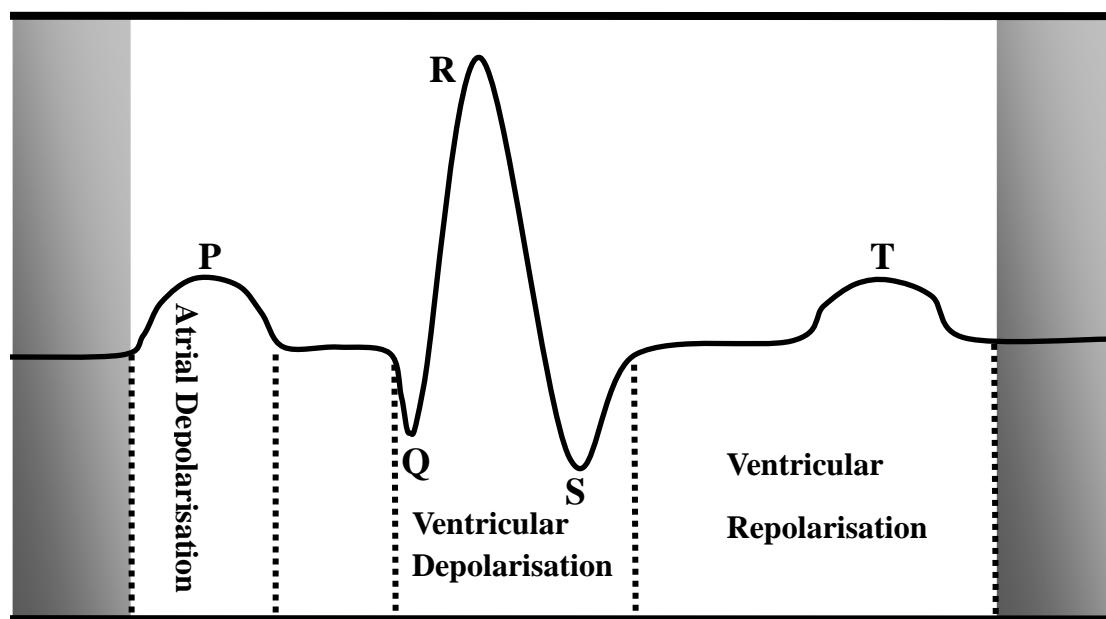


Figure 4: A complete cycle of a heart beat in ECG form with indications of the accompanying electrical changes [3]

In a standard ECG recording there are five electrodes connected to the right arm, left arm, left leg, right leg and chest of the patient. Depending on how the electrodes pairs are connected to the ECG sensor different waveforms and amplitudes can be obtained. Each pair contains unique information about the heart activity of the subject that cannot be obtained from another pair of leads [3].

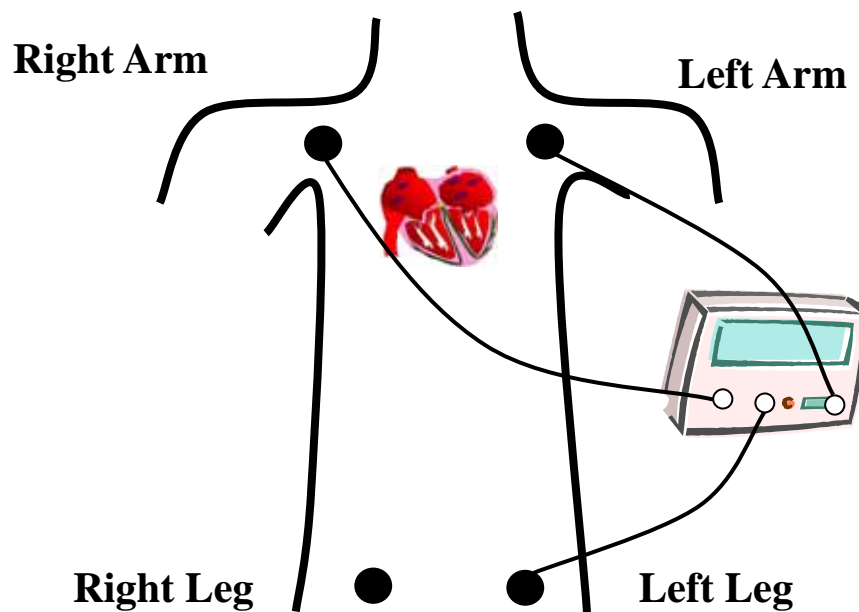


Figure 5: Different points for ECG detection [2]

#### 2.1.4 QRS complex

Although in general the term *heart beat* refers to the entire ECG cycle of PQRST, the QRS complex is taken as the portion of an ECG signal that represents the instant that a beat occurs in this project. The QRS complex portion of the ECG is the most distinctive resulting in an easier identification and the duration of the QRS complex is normally less than or equal to 100 milliseconds [2]. Figure 6 shows a time versus voltage graph of the QRS complex portion of the ECG,

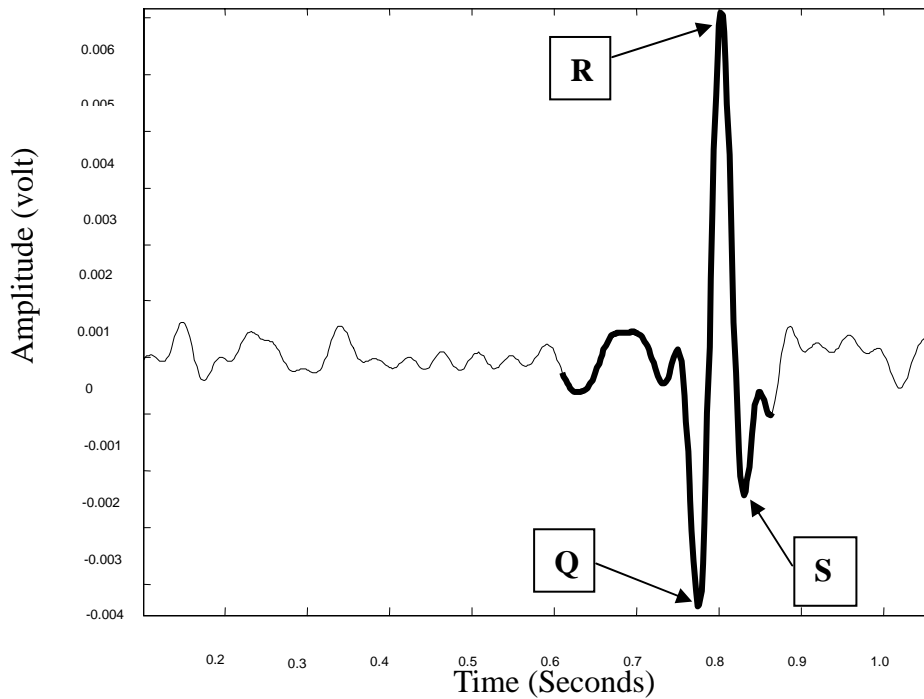


Figure 6: QRS Complex, a portion of ECG [1]

The R wave is the most prominent wave in an ECG, and this project mainly concentrates on finding the R waves in a stream of ECG data, reacting to the detected R waves and collecting the context information of the daily activities.

## 2.2 CURRENT HOLTER MONITORING SYSTEMS

Chapter one provided a brief introduction to the Holter monitor [25] [19] [48], an ambulatory device that records ECG signals over long intervals. A Holter monitor is designed to support the identification of infrequently occurring cardiac arrhythmias. In the traditional use of a Holter, subjects who wear a Holter monitor are asked to keep a timed paper record of their activities of daily living so that it can be related to



the ECG signal. A Holter takes 3 to 12 leads for the ECG recording depending on the requirement.

A classification [32] for the Holter monitoring systems is made on the following:

a) dumb Holter systems; b) cardiac event monitors - recording with remote classification; c) recording plus local ECG classification but no context awareness other than the handwritten record; d) recording plus awareness of the level of physical activity through accelerometers.

### **2.2.1 Type one – Dumb Holter**

These type of systems record signals and perform post-processing off-line. A set of electrodes [49] are placed on the patient's chest with the leads connected to the Holter. The monitor is configured to continuously record ECG signals for long periods of days, weeks or months. When the recording is finished, a clinician will analyze those recorded ECG signals and try to discover whether any abnormal beats or rhythms occurred during this recording period. The Medtronic Reveal® Insertable Loop Recorder [50] can perform up to 14 months ECG recording. This device can capture the abnormal activities of the heart because of its long time recording capabilities, but the downside cannot be ignored. In the absence of real-time processing, if the patient's heart problem appears irregularly, there is no way to pinpoint it until the recorder is returned to the clinician some time after the event occurred, which may leave the patient's life in danger.

### **2.2.2 Type two – Cardiac Event Monitor - Recording with remote classification**

In order to overcome the restriction described above, the second type of recorder performs remote real-time classification while patients continue living a normal life. Some examples of this type of ECG recorder are mentioned below. There are many examples and a short list of commercially available devices of this type is provided. Vitaphone [52] commercializes a card that can transmit ECG data by infrared to a mobile phone that automatically transmits the ECG to a service center where the ECG analysis can be made. QRS Diagnostic [51] supplies the EKGCard that can convert any computer (PC, laptop or PDA) into an electrocardiograph that allows the visualization and storing of ECG data. They also provide software which can be used to analyse ECG signal that runs on PCs or laptops but not on PDAs, although the result of the analysis can be made available on a PDA for reviewing purposes. Cardio Control [53] commercializes a product that allows the visualization and recording of ECG signals on a PDA. The signals recorded by this device can be transferred to a workstation where they can also be analyzed and printed. Pulse Medical Limited [54] have commercialised a product, MeditSense, that is a complete 12 lead ECG system designed for mobile and stationary use where ECG data can be recorded in a Tablet PC (but not in a PDA). The MeditSense system also provides interpretation of the ECG which can be used as a guide for diagnosis conclusions. MobiHealth project [55] has developed a vital signs monitoring system based on a body area network and a mobile-health service platform that can transmit sensor measurements via UMTS or GPRS to a backend system, where a remote detection of emergencies is performed.

### **2.2.3 Type three - Recording plus local ECG classification but no ADL context awareness**

The systems described in section 2.2.2 perform remote real-time ECG monitoring and overcome the previous limitations. Most of them make use of mobile telephones and/or PDAs to capture the ECG signal and send it to a monitoring center where the real-time classification is performed. ECGs are continuously transmitted to the remote monitoring station through the wireless network (GSM [57] or 3G [56]). Because those signals are sent continuously, this implies a comparatively large cost for network transportation and possibly loss of ECGs which could potentially fail to report key features in the ECG as they occur.

The third type of recorder provides real-time classification by using an architecture that includes an intermediary local computer between the sensors and the control center. Those computers perform some local real-time monitoring in order to detect some anomalies and send alarms to a control center or a hospital. Research projects such as @Home [58], TeleMediCare [55] or PhMon [59] included wireless bio-sensors that measured vital signs such as heart rate, blood pressure, insulin level, etc. Typically in this approach, the health monitoring system is carried by the patients and it controls these sensors and performs some analysis.

### **2.2.4 Type four ADL context awareness through accelerometers**

It has been mentioned at the start of this section that the wearer of a Holter monitor is

requested to use a paper based record called an *event diary* to accompany the recording which records the activities of daily living. Some Holter monitors [13] [12] have also taken the approach of intelligently sensing the patient's activities. For instance, Healy [13] proposed an accelerometer-based activity monitor to detect the wearer's energy usage every day by combining the ECG data to facilitate the doctor to better interpret the ECG data in context. Because Healy's context collection is automatic detection, the collected information can not cover all the context circumstances that are relevant for the detected heart events. However it opens a great vista for future context awareness projects.

Among the sensors used in the projects mentioned above we can find ECG sensors connected via Bluetooth to a smart phone or a PDA that act as a "base station". The base stations analyse the signals and in cases where a parameter exhibits a critical characteristic, the patient is informed. Moreover, the base station, via a mobile communication network, keeps in contact with a central electronic patient database and a medical call center where the acquired signals and vital parameters are reviewed by doctors.

The previous sections have described the electrical activity in the heart and how it is measured by various classes of ECG recorder. Before providing an overview of the proposed system, it is useful to describe the technological and theoretical background for the work. The next few sections will introduce some of the more detailed concepts

that are used in this work.

## 2.3 BLUETOOTH

Bluetooth is named after Harald "Bluetooth" Gormson [62], who was born around 910 AD and died in 985 or 986 AD having ruled as King of Denmark from around 958 and King of Norway for a few years, probably around 970 AD. Its specification was developed in 1994 by Sven Mattisson [60] and Jaap Haartsen [61], who were working for Ericsson Mobile Platforms in Lund, Sweden.

Bluetooth is a specification that supports wireless communication within a short range [5] and it has been selected to implement this project [6]. Bluetooth is an industrial specification for wireless personal area networks (PANs), also known as IEEE 802.15.1 [4]. Bluetooth provides a way to connect and exchange information between devices such as PDAs, mobile phones, laptops, PCs, printers and digital cameras via a secure, low-cost, globally available short range radio frequency [5].

Bluetooth is the most popular short-range wireless medium [5], and its robust spread spectrum uses a technique known as *frequency hopping* [63]. Frequency hopping is known to have the advantage that the associated signals are highly resistant to narrowband interference and difficult to intercept, and spread-spectrum transmissions can share a frequency band with many types of conventional transmissions with minimal interference. As a result Bluetooth has comparatively reliable performance [5]

which helps make it a suitable channel for medical communications [6].

Java Bluetooth applications are usually developed with the employment of the Standard Java Bluetooth API and the *Serial Port Profile*, a part of the Bluetooth stack that makes a Bluetooth channel behave as a simple wireless serial link. The standard Java Bluetooth API called *JSR82* has been implemented by many vendors such as Nokia Corporation, Motorola, and Rococo Software [64]. Rococo's implementation for JSR82 suits the need of this project but is expensive to buy [65]. The Serial Port Profile (SPP) is based on the ETSI [66] TS07.10 specification and uses the RFCOMM protocol. As mentioned above, it emulates a serial cable to provide a simple-to-use wireless replacement for existing RS232 based serial communications applications. This functionality is supported either natively or by add-on drivers by many Java Virtual Machines. By providing interfaces to the communication ports using the Serial Port Profile, it is possible to build a Java application for short range communication usually at no extra cost. The disadvantage is that serial port communication requires manual intervention on the part of the user and other services of Bluetooth will not be available without the support of JSR82.

This option, while attractive for this work, would not be very suitable in a commercial situation.

## 2.4 EVENT – CONDITION – ACTION (E-C-A)

Event-Condition-Action (E-C-A) rules are a mechanism first developed by Oracle [67] for use in active databases and they are a natural method for supporting reactive functionality [68], [69]. The important features of an E-C-A language are its reactive and reasoning capabilities, the possibilities to express complex actions and events and declarative semantics [70], [71]. The form of E-C-A is **On** [Event] **IF** [Condition is TRUE] **DO** [Action], which means that when Event occurs, if Condition is verified, then execute Action. E-C-A systems receive inputs (mainly in the form of events) from the external environment and react by performing actions that change the stored information (internal actions) or influence the environment itself (external actions)[71]. Conditions can range from simple to compound logic statements.

This work proposes to detect cardiac arrhythmias within a stream of sampled ECG data which is received by a PDA through a Bluetooth link. The occurrence of a heart beat as detected by a beat detection algorithm is considered an event with respect to the scope of this project. The following format briefly illustrates the event handling process,

### **When Beat Occurs...**

**If** Condition 1  
**If** Condition ... and **Context Information** -----► **Do** {Actions...}  
**If** Condition n

## 2.5 KVM

A KVM (kilobyte virtual machine) is a small footprint Java Virtual Machine for use in an embedded system or other memory limited device. A KVM [72] is designed to support uniform and standardised deployment on devices with approximately 128kB of available memory. It supports the core Java features and Java APIs called for by the Java 2 ME architecture. However, products supporting the KVM specification can omit some of the optional functionality from initial device configurations, depending on the features and capabilities of the device [142]. The Java 2 ME architecture includes a small number of configurations that are specified and standardized by Sun. The KVM specification provides the guidelines that Sun uses in defining configurations. It details which features can be omitted from an initial configuration.

## 2.6 JAVA ARCHIVE FILE (JAR)

The application will be compiled into a jar file that will run in the iPAQ 5550 PDA. The Java Archive (JAR) file format enables the developer to bundle multiple files into a single archive file. *“A Jar file is nothing more than a Zip file whose file entry is a specially named manifest file that contains attributes and digital signature for the Zip file entries that follow it”* [143].

Typically a JAR file will contain the class files and auxiliary resources associated with applets and applications.



## **2.7 FINAL REMARKS**

This chapter has described ECG briefly and some of the relevant fundamental concepts that are involved in this project. The introduction of those concepts imply a Holter monitor system, which will employ Bluetooth as the transportation method and E-C-A will be the backbone of the detection of the heart abnormalities and the context collection process. An overview of the proposed system is detailed in the next chapter.

## **CHAPTER 3**

# **METHODOLOGY AND OVERVIEW OF PROPOSED SYSTEM**

### **3.1 INTRODUCTION TO THIS CHAPTER**

This chapter gives a brief description of the main methodologies which were utilized in this work. It will also illustrate the proposed system by decomposing it into 3 parts, which are *data transportation*, *beat detection* and *reaction to events* (context information collection using the E-C-A mechanism). This is followed by a description of the interaction within and between those modules. Finally the chapter deals with issues of technology selection that lead up to the description of the implementation in Chapter 4. First of all, the constraints identified for the proposed system are shown in section 3.1.1, 3.1.2 and 3.1.3.

#### **3.1.1 System constraints that are imposed by the signals**

As discussed in section 4.4.2, the sampled ECG data of the type recorded by Holter monitors has the following characteristics which will impact on the system.

1. Sampling frequency is up to 500Hz
2. Data resolution of 12 bits that can be extended to 16 bits for convenient transmission
3. At least one channel for receiving the ECG data.

### **3.1.2 System constraints relating to the wireless transmission**

From the above description, the minimum data rate that the PDA should have for single channel operation is  $500 \times 16 \times 1 = 8\text{k}$  Baud. However, serial communications employ certain agreed Baud rates and the lowest such Baud rate above 8k Baud is 9,600 Baud. On the other hand for two-channel operation, which would be typical for a Holter monitor, the minimum Baud rate would need to be 19,200 Baud.

### **3.1.3 Platform constraints**

The system will be implemented using Java on a KVM on an iPAQ 5550, which has the following characteristics.

1. Only Wi-Fi or Bluetooth transmission is natively supported
2. 128 MB of RAM
3. Battery capacity 5V DC (2A)
4. Display size 3.8 inch with 64k colors
5. Processor Speed 400MHz

## **3.2 METHODOLOGIES USED IN THIS PROJECT**

### **3.2.1 Open Distributed Processing and its use in this dissertation**

Before proceeding with the high level description of the proposed system, it is useful to explain the framework which is used to describe it. One useful methodology that is used to document the multiple facets of technological solutions is the ISO Open

Distributed Processing (ODP) approach [73]. The main goal of a distributed computing system is to connect users and resources in a transparent, open, and scalable way. A good example of the use of the ODP methodology is given by Sinnot [74]. The ODP approach features five different views on the solution. It has been decided to use this approach in a flexible way to describe the proposed system. The five ODP viewpoints are described below.

The **Enterprise Viewpoint** concentrates on the business environment in that a system has to operate and is intended as a manager's view of the system. A clear view of the intended use of the system is required in this view, i.e. the constraints of the system must be laid out. This view of the proposed system is covered in the **first part of this chapter**.

The **Information Viewpoint** focuses on the information and associated processing of a system. The information viewpoint uses schemata and information models of the type that can be expressed by the Unified Modeling Language to specify the way that information may be transformed. This view is described in **Chapter 4**.

The **Computation Viewpoint** focuses on the detailed software engineering aspects such as application programming interfaces and algorithms. This material can be found in **Chapters 5 and 6**.

The **Engineering Viewpoint** focuses on the deployment aspects of a system and provides the mechanism for realizing the computational model. Deployment is not a prominent part of this project, but a brief description of the hardware and software platform can be found in **Appendix C**.

The **Technology Viewpoint** details specific technologies, both hardware and software which will be used to implement a system. This viewpoint is covered in the second half of this chapter.

Other methodologies were employed in the design and implementation of this system, including the *Unified Modelling Language* and *Throwaway Prototyping*. The next couple of sections provide a brief description of each of these methodologies.

### **3.2.2 Unified Modeling Language**

UML (Unified Modelling Language) [75] and Throwaway Prototyping [30] are employed for system diagrams and for programming, respectively. First, UML is officially defined at the Object Management Group (OMG) [76]. OMG™ is an international, open membership, not-for-profit computer industry consortium.

UML was designed to specify, visualize, construct, and document software-intensive systems [27] and thus promote understanding between software engineers and other stakeholders in the software development process. A simple guide to the nomenclature

of UML diagrams that is used in this dissertation is given in **Appendix I**.

### **3.2.3 Throwaway Prototyping – System development process**

This project employs throwaway prototyping [77] as the standard protocol to develop the whole system. It refers to the creation of a system that will eventually be discarded rather than becoming part of the finally delivered software. After preliminary requirements gathering is accomplished, a simple working model of the system is constructed to visually show the users what their requirements may look like when they are implemented into a finished system. The most obvious reason for using throwaway prototyping is that it can be done quickly. Following the throwaway prototyping, this project follows the steps shown below,

***Step 1: Beat detection in MATLAB***

***Step 2: Beat detection in Java***

***Step 3: Simulated ECG measuring device on a PC***

***Step 4: Communication***

***Step 5: E-C- A Context collection***

This project produced a mind map (**Appendix E**) to show all the background material that has to be taken into account in order to follow the steps shown.

## **3.4 ENTERPRISE VIEWPOINT OF THE COMPONENTS OF THE SYSTEM**

This system employs Bluetooth as the wireless channel to receive ECG signals and

reacts to the detected heart beats while recording the context information for better diagnosis afterwards. As mentioned in the previous chapters, the main software artifact produced by this project will be a mobile PDA-based interactive ECG recording system, which provides high mobility and appropriate user interaction to supply pre-emptive cardiac wellness monitoring while the patient is either in a stationary or mobile situation. The system will monitor cardiac rhythms from the ECG signal and upon discovering changes in heart rate will ask the subject to enter information about their current activity of daily living.

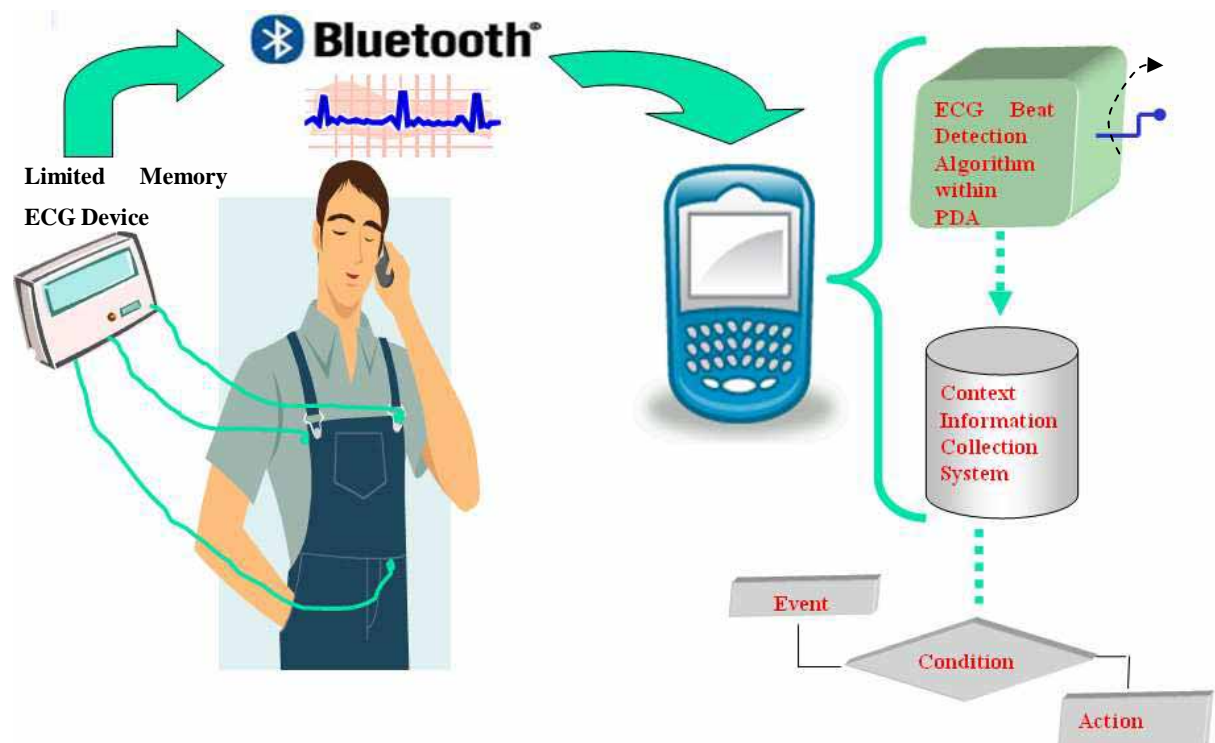


Figure 7: The proposed system in actual use including Bluetooth enabled ECG measurement device

Figure 7 shows the main components of the proposed system. The system is composed of three different parts, which are data transportation, beat detection and

reaction to events (context information collection using E-C-A rules). Ideally, one of the currently available Bluetooth-enabled 3 lead ECG instruments would record the ECG signal and send them to the PDA which is also carried by the subject of care. However this project is actually not using a real ECG measurement device and a simulator was developed by the author to read the ECG data from a file and send them to the PDA through the Bluetooth dongle in a PC. An overview of the system is shown in Figure 8.

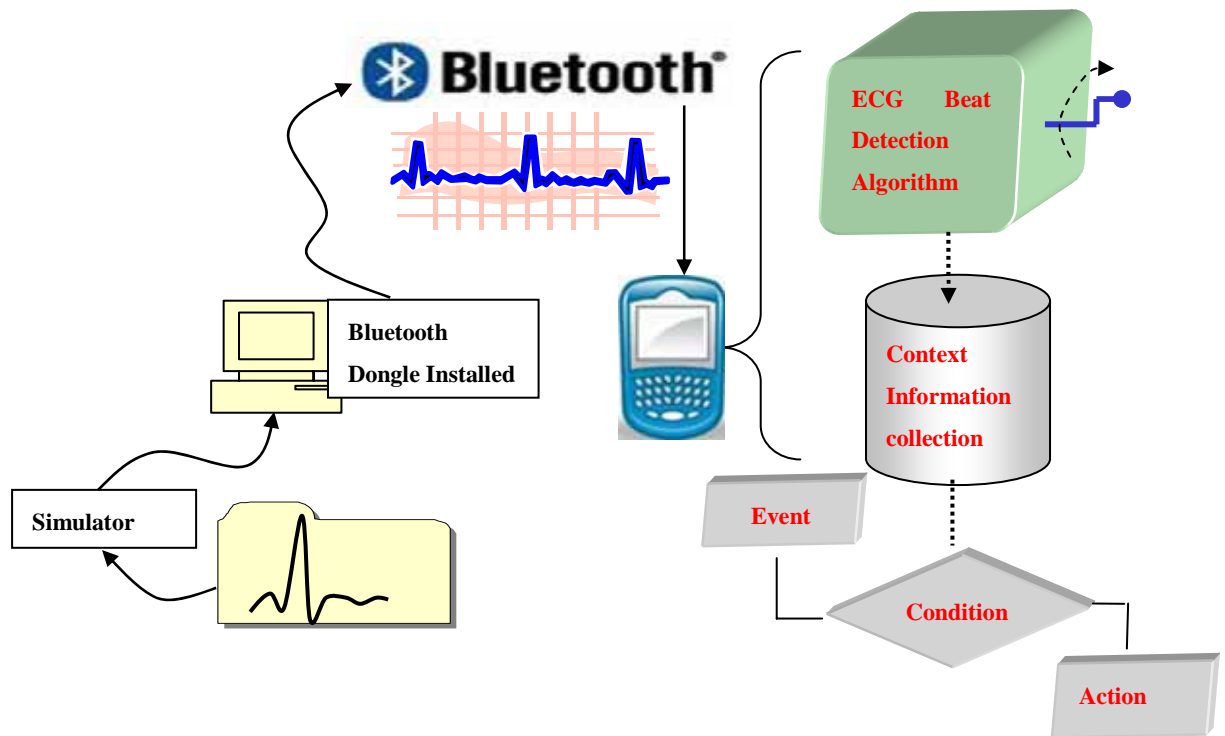


Figure 8: The proposed system with the simulator

When the ECG signals are sent to the PDA, they will be first analyzed by an ECG beat detection algorithm. Because the 26<sup>th</sup> order FIR filter of the detection algorithm, the last 27 samples of the data in the ECG frame will be missed, which could cause false beat detection or miss a heart beat. Therefore the system will retain a circular



buffer copy the last 27 samples into an array, when the next frame arrives, it will join these 27 samples with the first 473 samples in the coming frame to avoid wrong detections. Whenever a beat is detected, it will be considered as an event that will be passed into the next level - the E-C-A rule-base (reacts to the beat events generated by the beat detection algorithm), which will do a series of analyses of the generated events, and then evaluate the conditions and acquire the appropriate ADL context information, once these two are fulfilled, the context collection system will trigger the corresponding actions against the patients' current heart situations. It has been mentioned that one constraint of the system is the low memory of the handheld device – PDA. For this reason, the data communication, algorithm detection and context collection could occupy a high percentage of the PDA's resources. The following sections give an overview of the three components.

### **3.4.1 Data transportation module**

This system is intended to form part of a wireless personal area network (WPAN), where a WPAN is a low-cost networking scheme that enables computing devices such as PCs, laptop computers, printers and personal digital assistants (PDAs) to wirelessly communicate with each other over short distances [78,79]. It could serve to interconnect standard computing and communicating devices, or it could serve a more specialized purpose such as allowing the surgeon and other team members to communicate during an operation [73]. This project employs Bluetooth as a communication method for data transportation to increase the mobility of the

healthcare such that the patient can move closer to the centre of care. It is known that the traditional medical measurement device typically has multiple wires that limit the mobility of the patient. The system presented here will not eliminate the wires of the ECG instrument, but it will partially decouple the ECG recording instrument and ECG analysis system, which will make the pair of devices more comfortable to wear. This is illustrated in Figure 9.

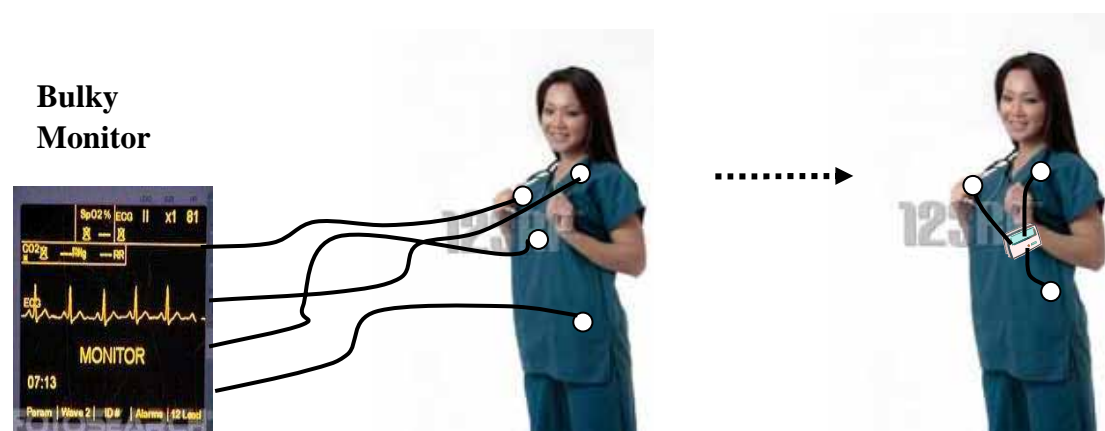


Figure 9: ECG measurements from bulky to portable

The ECG signal can be sent through different transportation channels, such as Bluetooth, Network, or serial cable etc. The current intention is to apply Bluetooth for clinical monitoring, but as it is not guaranteed that Bluetooth will prevail forever, another technology will replace Bluetooth in the future. For better adaptation and easier upgrade of the whole system, the modular structure of the information transportation part of the system will allow it to be adapted to other communication channels as they become available. In any case, combining the wireless technologies with the ambulatory ECG could make the monitoring process more convenient.

### 3.4.2 ECG beat detection module

Several ECG beat detection algorithms were investigated first by research and then using MATLAB [Appendix A]. This process will be discussed in Chapter 4 as well as the justification for choosing the Zero-Crossing algorithm [80] for Java Implementation. The Zero Crossing approach is based on a feature obtained by counting the number of zero crossings per segment in a “chopped” version of the ECG wave form. It is well-known [80] that zero crossing methods are robust against noise and are particularly useful for finite precision arithmetic. The excellent performance of the algorithm was confirmed in other work by a sensitivity of 99.70% (277 false negatives) and a positive “predictivity” of 99.57% (390 false positives) against the MIT-BIH arrhythmia database, where 135 MIT-BIH ECG records with 91006 real beats or nearly 24 hours of ECG recordings [80] were tested under this algorithm.

Figure 10 illustrates the Zero-Crossing algorithm. The corresponding components are as follows:

- a. A portion of ECG Signal;
- b. 26<sup>th</sup> Order FIR Filter;
- c. Comparison between the original ECG and filtered ECG signal;
- d. Signal Processing Block 1 that calculates a low passed “mirrored” digital signal;
- e. Signal Processing Block 2 that generates a feature signal  $D(n)$ ;
- f. Threshold value determination based on feature signal;

g. Decision Logic that detects beats.

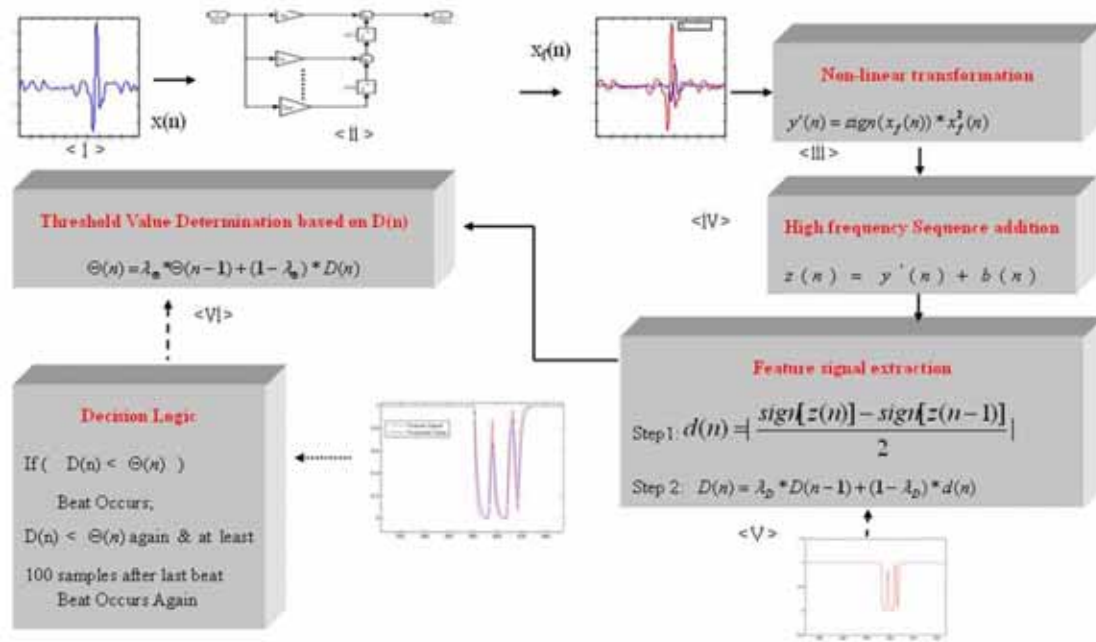


Figure 10: Zero-crossing algorithm overview

The algorithm is discussed in detail in Chapter 5.

### 3.4.3 E-C-A Event processing module for Context information collection

When the beat event is generated from the last step “Zero-Crossing”, it will be loaded into the E-C-A Event Processing Module. A raised **Event** triggers a number of calculations, and the calculated results will be processed by evaluating the **Condition** component of each applicable E-C-A rule. Upon receipt of anomalous or raised heart rate, the **Condition** component of the event processing module will issue the **Action**. In this case the Action will call the user interface to request the subject to input the appropriate context information. An overview of this part of the system is shown in Figure 11..

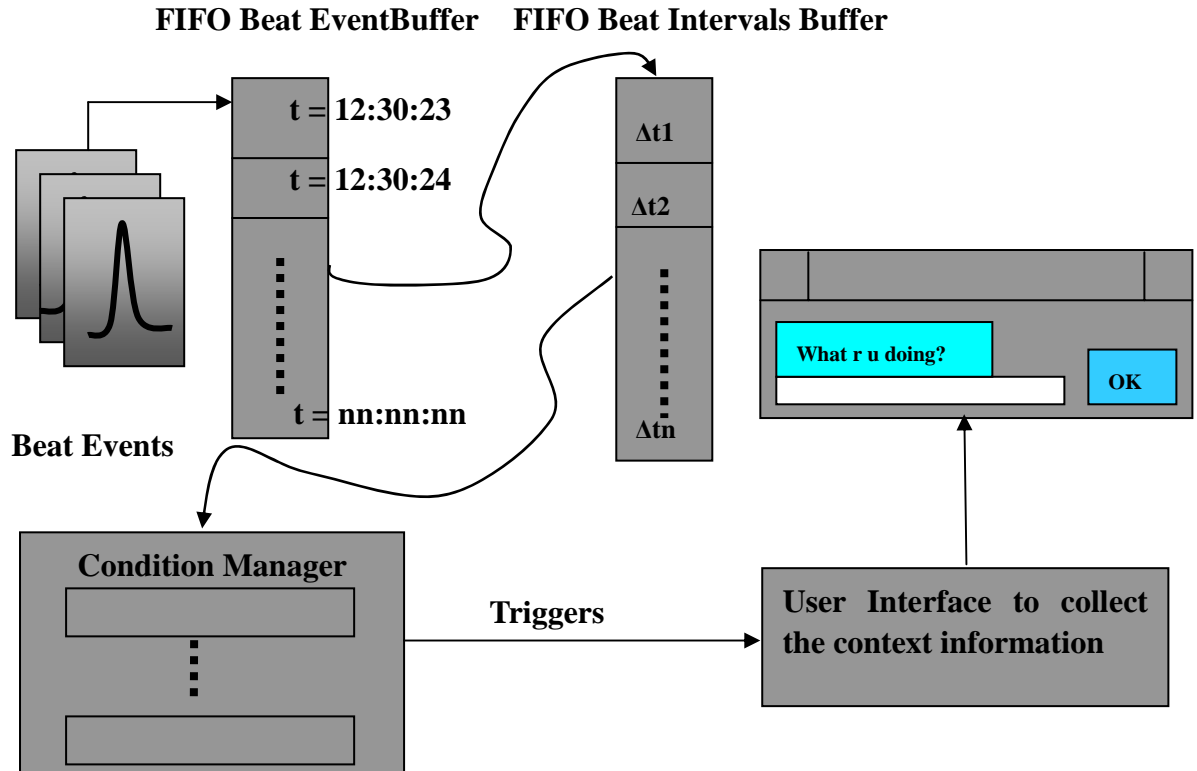


Figure 11: Overview of E-C-A Event Processing Module

The availability of Bluetooth-enabled ambulatory ECG measurement equipment has made it feasible to implement such a mobile and interactive system. Although it could be argued that the serial port profile approach and the associated security issues would need to be solved to make the system practical [145], the work satisfies the project's scope. A more detailed description via ODP information and computation view of the three modules will follow in Chapter 4.

### 3.4.4 Bluetooth enabled Electrocardiograph Simulator for a PC

The scope of the project does not permit the author to work on the measurement of ECGs. However, to simulate a real ECG instrument that can send ECG signals

through a Bluetooth channel, the author developed a simple java application for a PC that can read an ECG file. The electrocardiograph simulator application transmits the ECG data through a Bluetooth dongle on a PC via a Bluetooth serial profile link to the PDA where it is processed by the main Java application.

The “Interaction” is based on Event-Condition-Action rule that was first developed by Oracle [67] for active databases. It is implemented through the Java Event processing paradigm. There is listener associated with each communication channel. Figure 12 gives an overview for the process.

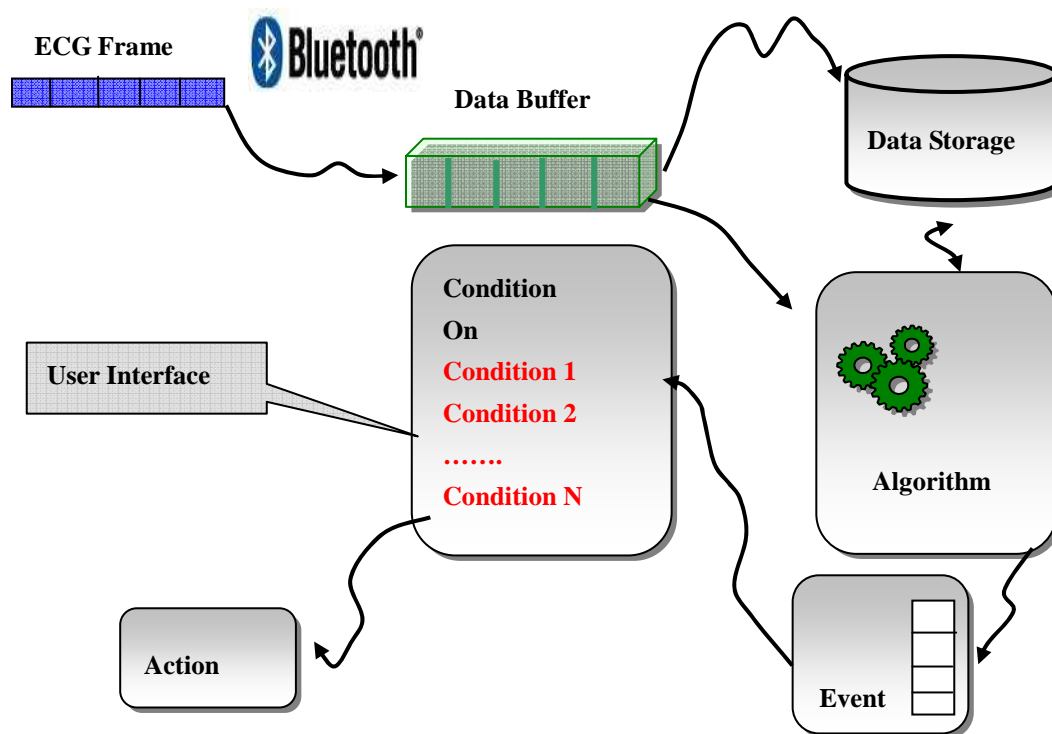


Figure 12: Event processing interaction system

### 3.5 HOW THE SYSTEM WORKS – AN ENTERPRISE VIEW

Having described the components of the context collection system, it is now possible

to describe at a high level how the parts of the system cooperate to detect and respond to cardiac arrhythmias. In the following sections, the terms indicated by “courier new italic” are objects or methods of object.

### **3.5.1 From transmitted ECG sample to sample for beat detection**

The simulated Electrocardiograph transforms each ECG sample from the file to a series of pairs of 16-bit integers, which is commonly called a frame in communications engineering and transmits it across the Bluetooth serial link. Figure 13 describes the first step of the process of parsing and responding to the ECG frames in more detail. The first interaction with incoming “frames” of pairs of 16-bit ECG samples (1) which requires 32 bits per pair of samples recorded at each sample interval from the simulated Bluetooth enabled ECG measurement device, is with a Serial port event listener (2). This object is delegated by the Bluetooth Manager to monitor the Bluetooth Serial input stream continuously. Whenever an ECG frame arrives on the Bluetooth serial port on the PDA, the Serial port event listener will call a method from the *BluetoothManager* (3) to request that it retrieves the frame (4) unpacks the payload of ECG samples and stores these samples in a small FIFO data buffer that can hold exactly one ECG frame. This buffer is used to pass the samples on to the next stage and also store them into a file (5) for later data retrieval.

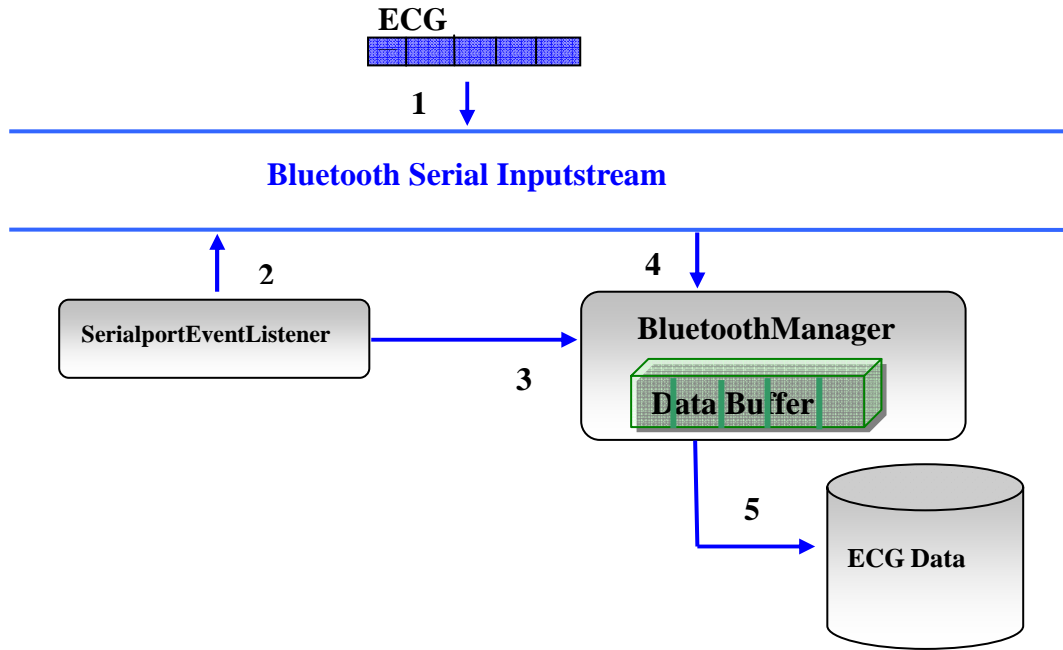


Figure 13: BluetoothManager and Serial Port Event Listener

Every ECG frame contains 500 samples along with patient ID, sampling frequency and the frame number. Patient ID and sampling frequency are used to identify whether the ECG data is from the right source. The frame number is used to determine if there is any frame loss during the data transportation. The job of the simulator is to combine ECG data samples, patient ID, sampling frequency and the frame number together and send the frame down to the Bluetooth serial input stream. There could be more parameters to be put into the frame for better monitoring. For instance, Instrument ID is to identify the type of the instrument and time stamp is to store the recording time.

Once an incoming frame has been stored, a Java object called *DataReceiver* copies the contents of the Data Buffer of *BluetoothManager* that was described in the



previous section and transforms and scales the two-byte data points into the Java double data type that is used for further data processing and stores the transformed data into its own array. A second listener object called *DataEventListener* monitors the *DataReceiver*'s array. Whenever this array is full, it will invoke the object *Algorithm* to calculate whether a QRS complex exists within the corresponding sampled data as shown in Figure 14.

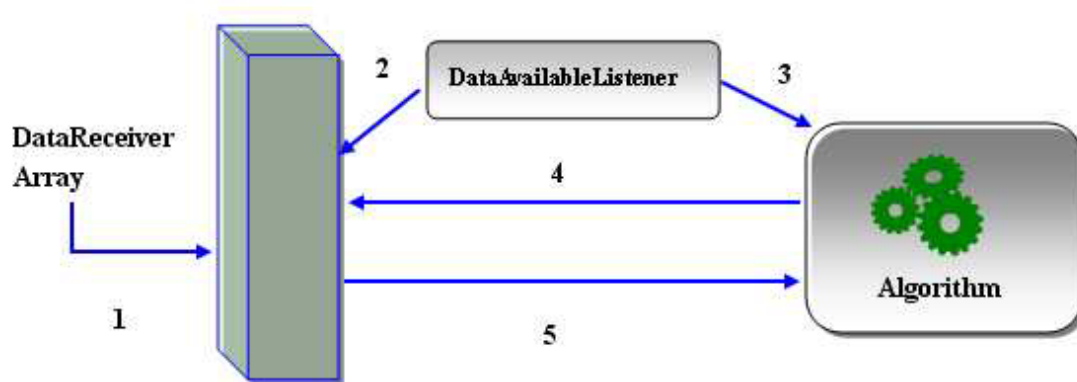


Figure 14: Algorithm and Data Available Listener

### 3.5.2 From sample in algorithm to beat event

The *Algorithm* will continuously analyze the incoming ECG frame. Whenever a heart beat is located, an event object called *BeatEvent* will be instantiated as a manifestation of the identified heart beat. This object also holds the associated time information, wraps them together as an event and those events are continuously stored for further processing under the supervision of the object *BeatEvent*.

### 3.5.3 From beat event to action

Once the beat interval is calculated, a listener object called

*ProcessedEventsListener* will inform the *ConditionManager* object. A set of rules of *the ConditionManager* are evaluated for the processed results from the *BeatEvent* stage. A user interface screen, managed by the *UserInterfaceManager* object, supplies the appropriate context information for the *ConditionManager* to trigger the corresponding actions against the current heart condition from the *ActionManager* object.

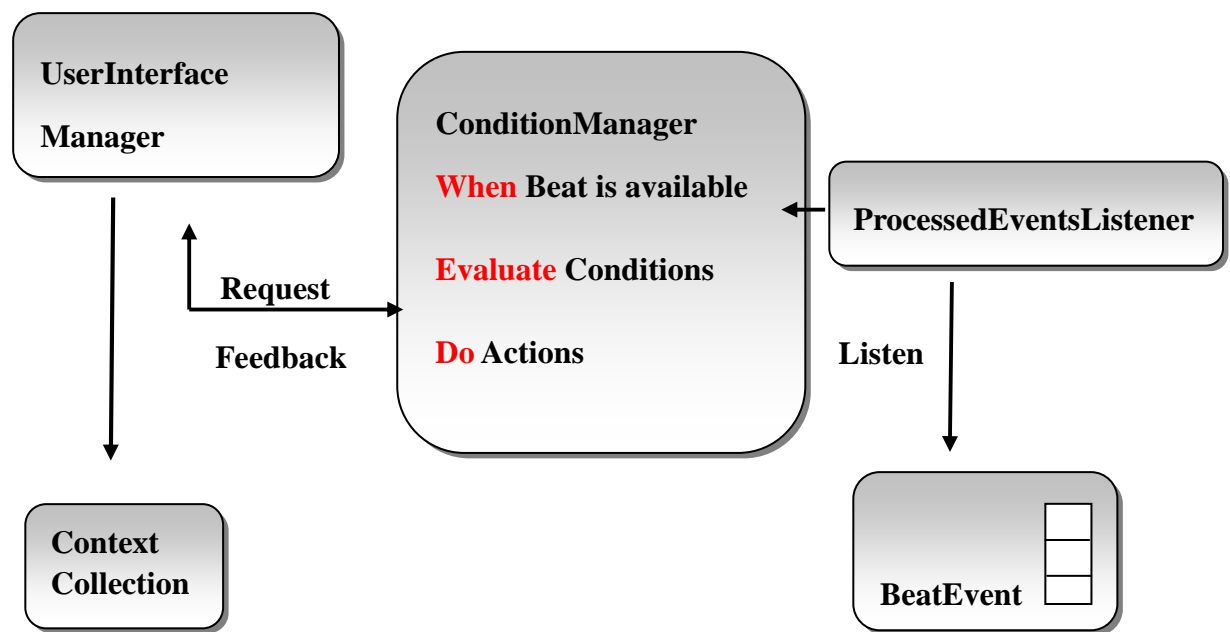


Figure 15: Beat Event processing and context information collection

### 3.7 TECHNOLOGY VIEWPOINT – TECHNOLOGY CHOICES

Following on from initial discussions in Chapters 2 and 3, the remaining sections of this project correspond to the technology viewpoint of the ODP approach. The technologies, tools and setup issues associated with the implementation of the system are detailed and discussed here.

### **3.7.1 Choice of Development and Testing Platform**

Java was chosen as the implementation language for the system because of the relative simplicity of developing in Java, because of the low cost and its standard support for Bluetooth and its popular advantage “Write once, run anywhere” [41] which it was hoped would really ease the development of applications.

This project employs Java programming language for the handheld device, i.e. HP iPAQ Java™ Platform, Micro Edition (Java ME) or Java Wireless Toolkit. Java provides a robust, flexible environment for applications running on a broad range of other embedded devices, such as mobile phones, PDAs, TV set-top boxes, and printers [81]; however in order to implement the nice features of Java, the handheld device must be Java enabled. In other words, it requires a Java Virtual Machine.

There are many Java Virtual Machines for handheld devices, For instance, both NSICOM’s CrEme and IBM’s J9 have been successfully deployed on PocketPCs. There are also other Java Virtual Machines on the market, but many of them suffer from the following problems: not supported anymore (Sun's PersonalJava), not fully Java compatible (SuperWaba), not compatible with Java Bluetooth Development Kit (Esmertec Jbed), or out of date (Insignia Jeode) [82].

Different Java Virtual Machines have different specifications and features, which may suit the requirements of this project to varying degrees. In order to find out which

Java Virtual Machine best suits the project, the following areas need to be compared: Supported Operating System, Compatibility with J2ME, Supported Processor, Java Bluetooth Compatibility, Speed, Extra Supported Packages, Known Bugs and Cost. Many areas have been compared among them though, the main factor is whether the Java Virtual Machine is suitable for the purpose of this project, such as whether it supports J2ME, Bluetooth Profile (JSR 82) or Serial Profile and has a higher speed for running Java applications. Considering these factors, NSICOM's CrEme Java Virtual Machine will be the first choice for this project.

NSICOM's CrEme supports the most popular handheld device platforms Windows CE and PocketPC, and it supports many processors such as SH3, StrongARM, XScale. Because the PDA that is employed in this project is the HP iPAQ 5550, its processor is Intel's XScale and its platform is PocketPC 2003. Therefore NSICOM's CrEme satisfies the first requirement.

The compatibility with J2ME is a prime factor for the comparison. If it does not fully support J2ME, it may cause problems with the programming part of the project. Table 1 shows clearly that only Esmertec Jbed, NSICOM and IBM's Java Virtual Machine fully support J2ME. Others like PersonalJava virtual machine do not support J2ME at all, Kada, Mysafu, and Jeode [83] only supports part of J2ME. Therefore the choices left are Esmertec Jbed, NSICOM and IBM's Java Virtual Machine.

One of the aims of this project is to build a Bluetooth Enabled Personal Area Network, so the Java Bluetooth development kit compatibility has to come into play. According to the customer support section from the corresponding website [84], Esmertec Jbed does not support Bluetooth profile (that is JSR 82). IBM's J9 is compatible with aveLink [85] however aveLink costs over €1000, which exceeds the funding of the project. NSICOM's CrEme was built up with Avetana [86] and costs only €25. Beside that, NSICOM's CrEme also supports many extra packages such as Swing, COMM, CORBA, SQL, and J2SE. The COMM package will support the serial profile that suits this project. To summarize the above discussion, a comparison table is given below to explain why NSICOM's Crème is chosen.

<b>Names /Versions</b>	Supported Operating system	JVM Compatibility	Supported Processors	Supported Hardware	Java Bluetooth Development Kit Compatibility	Speed	Extra Supported Packages	Known bugs	Cost
<b><u>NSIcom CrEme 4.11</u></b>	WinCE; PocketPC	J2ME	SH3, StrongARM, Xscal	PocketPC, Palm	Aventana	Fast	Swing, COMM, CORBA, SQL, JSSE	---	Contact NSIcom
<b><u>SunPersonalJava</u></b>	WinCE	PersonalJava	SH3, StrongARM, Xscal	PocketPC	Rococo & avalink	Fast	---	Development terminated	Free
<b><u>Jeode</u></b>	WinCE; Linux; PocketPC	CDCFoundation Profile	StrongARM, Xscal	Dell Axim X5, PocketPC	Zucotto	Fast	---	---	\$ 49.95
<b><u>IBM_J9</u></b>	PocketPC 2002 and 2003	J2ME	ARM, SuperH, PowerPC, Xscale, X86 & MIPS	HP iPAQ 5550, 2210 & more	avaLink	Fast	---	---	\$599 or \$6 per device
<b><u>Mysaifu</u></b>	Windows Mobile and PocketPC	Part of J2SE	StrongARM & XScale	PocketPC, Palm, Dell	---	Medium	---	---	Free
<b><u>Esmertec Jbed</u></b>	Windows CE/Mobile, Linux, Qtopia, VxWorks	CLDC 1.1, MIDP 2.0 & JTWI	ARM, StrongARM & XScale, TI OMAP	Many	No	---	JSR 36, 62, 129, 46, 66 & 169	---	Contact Esmertec
<b><u>Kada</u></b>	WinCE 2.11/3.0	MIDP 2.0	StrongARM & XScale	PocketPC	---	Fast	---	---	Contact Kada

--- = Only incomplete information found by the author

Table 1: KVM comparison

In summary then the development platform is presented in Table 2 below

Tools	
Software	Hardware
Java, NSICOM Crème 4.0[87], Windows XP and CE, Bluesoleil[88], RXTX[89], MATLAB[90]	PC, PDA HP Ipaq 5550 [HP]Bluetooth dongle

Table 2: Chosen development platform and tools

### 3.7.2 Choice of wireless communication medium between electrocardiograph and PDA

Two prominent characteristics of this system are that it is mobile and interactive as described previously. It has been stated earlier that Bluetooth has been chosen as the wireless communication medium. A number of reasons have also been given as to why Bluetooth is employed to increase “mobility”. As stated in Chapter 2 not only are most of the handheld devices and certain Electrocardiographs already Bluetooth enabled, this approach was inherited from Cronin’s previous research [6]. This system does not use the fully standardized JSR82 for Java Bluetooth communication for a number of reasons. This is mainly because of cost. However with the JSR82, the system will have more functions and better security. The serial port profile satisfies the system requirements, but the disadvantage is that it does not employ dynamic device discovery that forms part of the JSR 82 specification [8] and so needs manual setup for port numbers when the port number of the Bluetooth device is changed.

According to the previous research of Cronin [6], a number of different

communication methods have been explored in various healthcare scenarios, and Bluetooth is selected as the best transportation method at the moment for the body area network – BAN. But other transportation methods are also advancing. Therefore, as will be explained in the next chapter, a data transportation structure is developed based on the “Abstract Factory” Java pattern [91] to increase the adaptability and flexibility for the whole system. This structure allows the users to conveniently switch to any available built-in transportation method when required.

### **3.7.3 Other choices affecting wireless sample transmission**

For the data transportation, a very important issue is the data parsing, which directly affects the detection accuracy of the algorithm. The first question is the data resolution. As the scenario is to deal with ambulatory ECG, a 24 bit ADC (Analogue Digital Converter) is not required. Based on the results of the research from Zhou [92] and Fernández [93], a 12-bit ADC will fit for ambulatory ECG recording. Therefore appropriately scaled samples can be transmitted as a pair of bytes.

OBEX could be an option for file transfer but it will raise too many objects in the receiver side, and then the garbage collection of Java could kill the real-time process. Instead, Serial port profile is employed in this application. When sending ECG data to the PDA, a “frame” approach is introduced. A frame does not only contain the ECG data but also contains PatientID, SampleRate, and FrameNumber. Therefore during the transmission, it is easy to identify if the PDA is receiving the ECG data from the



right patient by comparing PatientID. Although it is recognised as an extremely important issue for a commercial system, this is not investigated in this project.

The third question is the data transfer speed. A double integer will take twice the time to transmit compared to a single integer. Integer arithmetic is introduced into the system to increase the transfer speed. Before the transmission, the format of the ECG data will be transferred from double into 16-bit short integer per channel. For convenience, when the data arrives in the PDA, those data will be parsed back into the original double format for the beat detection algorithm.

#### **3.7.4 Choice of beat detection algorithm**

Since the system is a Holter monitoring system that processes ambulatory ECG, the algorithm must be both fast and accurate. Besides this, the monitoring “gateway” device is a PDA with a limited memory capacity. Therefore the Java code implementation for this algorithm is required to occupy the memory of the PDA as little as possible. Various algorithms were investigated by the author using MATLAB first to explore their performance and to indicate the likely complexity of a Java code implementation.

The accuracy and the computational load were found to be about the same among Alfonso [76], Pan & Tompkins [94,95] and Kohler [80]. Kohler’s zero-crossing algorithm is best suited to Java implementation because it has only one filter and there

is no differentiation, reducing the complexities of the Java programming.

### **3.7.5 Java event model for implementation of E-C-A**

One of the convenient side effects of choosing Java as the programming language is the use of the Java event model. The context collection (E-C-A) uses Java `EventObject` and `EventListener` [81] objects. The `EventListener` listens asynchronously to the link that the `EventObject` has registered with the `EventListener`. If that particular link changes, the `EventListener` will pass the relevant information to the corresponding process to deal with those changes. Earlier in this chapter the reader was shown how when the heart beat is detected, it will be passed to an object that is called `BeatEvent`. This object processes detected beat events and then places the processed results into an array. Another object named as `ProcessedEventsListener` listens to this array all the time; if any change happens to this array, this object will pass the changed part to the next object `ConditionManager` that manages its own condition rule sets and calls the object `UserInterfaceManager` to input the correct context information for the system.

## **CHAPTER 4**

### **ANALYSIS**

Following the ODP approach, this chapter corresponds to the Information Viewpoint. It details the design overview of the system. This system contains three major parts, which are signal (data) transportation, algorithm and context collection. But first of all, it is necessary to describe the context of this research.

#### **4.1 A SCENARIO OF USE FOR BODY AREA NETWORK FOR INTELLIGENT AMBULATORY ECG RECORDING**

Recent technological advances in integrated circuits, wireless communications, and physiological sensing allow the fabrication of miniature, lightweight, ultra-low power, intelligent monitoring devices [97]. A number of these devices can be integrated into a Wireless Body Area Network (WBAN) [98], a new enabling technology for health monitoring.

Due to busy modern lifestyles, many people have become absorbed by day to day living, taking little time to take care for themselves. There have been quite a few documented cases of busy professionals with untreated diabetes who developed cardiovascular disease as a result of being unaware of their glucose levels [100]; other people with high blood pressure continue to take part in activities which could have adverse health consequences and then suffer a heart attack by not having paid

attention to warning signs such as their rising blood pressure levels [99]. While the introduction of body area networks of health sensors will not change behavior patterns, this technology, if it becomes sufficiently user friendly and unobtrusive to wear, has the potential to raise awareness among patients of health issues as they arise and thereby sustain wellness.

Smart healthcare applications need to adapt to the situation of the user in order to provide timely and tailored information in a way that is suited to the moment and to the context of use. In this project an infrastructure (S-A-E-C-A) is developed to support this type of smart context aware application.

The PDA that is developed in this research is intended to form part of an intelligent Holter system. The following scenario might apply: A person with irregular heart activities is advised by a doctor to take a Holter monitoring of their cardiac biosignals for 24 hours, or longer, to aid diagnosis. During this ambulatory ECG recording, the person may experience arrhythmias that are due to their medical condition. They may also experience a raised or modified heart rate that is due to the normal activities like running, becoming excited or agitated or even on account of the surrounding environment [101]. For instance, if the wearer is exerting themselves by running, cycling, or climbing stairs, their heart rate will rise. In the scenario presented in this work, the ECG monitoring device and Java application on the PDA will cooperate to quickly detect this change in heart rate and popup a screen on the PDA to ask the

person to input their current activities, i.e. the context information of activities of daily living. When the wearer tells the PDA the reason for the raised heart rate if it exists, the PDA will record the reason and the time that this activity occurred for future analysis. If the wearer tells the PDA that they were doing nothing that was likely to change heart rate or if an isolated irregular heartbeat occurs, the PDA will suspect that a significant artefact has occurred and will remember the time that this unexplained abnormality happened. When the responsible clinician retrieves the monitoring device, they will not only get the ECG readings and the handwritten journal of the patient but also analysed ECG results and related ADL context information. It is hoped that this will support the best decision for the patient. Therefore, the context information collection will help enhance mobile ECG recording. In this project, a 30 minutes ECG data [Record 100 in MIT-BIH ECG database] [106] is tested in the system.

## **4.2 SYSTEM DESIGN ENVIRONMENT**

The application is written in Java, therefore an appropriate KVM has to be selected to fully support the application.

As it was mentioned in Chapter 2, there are different data transportation methods such as Bluetooth, Zigbee and Wi-Fi. This research proposed to use Bluetooth following the previous research of Cronin [6]. However other technologies could make significant improvement after a future upgrade or modification; it would better have a

generalised transportation structure to be compatible with other transportation methods such that the adaptability of the system can be improved. A range of real time ECG beat detection algorithms are currently available. Some algorithms can perform additional analysis on the ECG signal at the cost of additional computational load. Because the detection of heart rate is the main computational objective in this research, a lightweight algorithm with less functionality but high accuracy for the beat detection will be selected.

### 4.3 SYSTEM DESIGN PROTOCOL – SAECA

S-A-E-C-A stands for Signal – Algorithm – E-C-A, and it is a generalized format that could also be applied to other areas. When the signal is available on the device, the algorithm will process the incoming data and then pass the calculated result into the E-C-A part to make decisions or take actions against the current situation. Figure 16 illustrates the workflow among the S-A-E-C-A system.

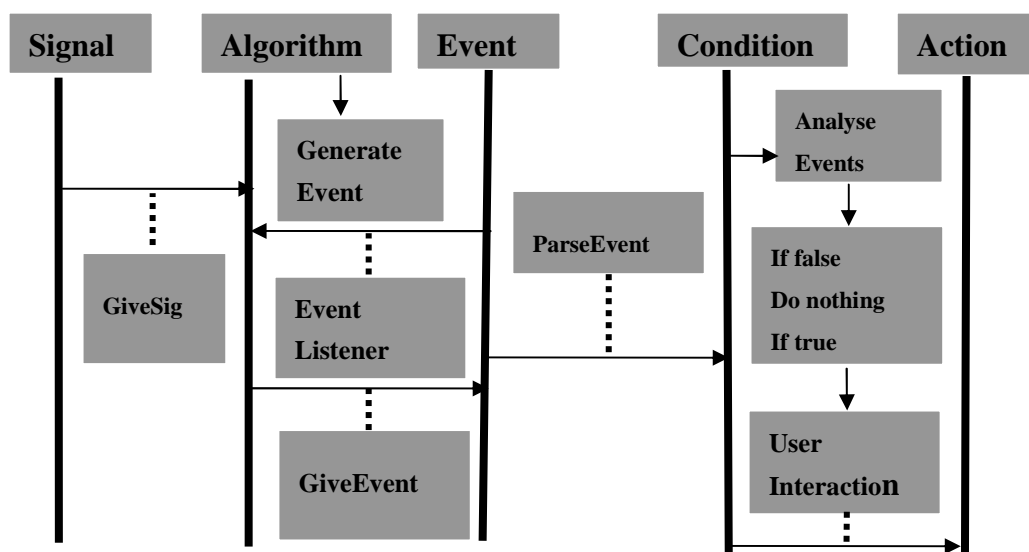


Figure 16: UML Sequence Diagram of the design protocol – SAECA

This process is Object-oriented, and every stage is interconnected through a registered listener. In Figure 16, whenever the signals are transmitted across the Bluetooth link to the system, the block Signal will catch the signal and call the method `GiveSig()` to transmit the raw signal to the block Algorithm. This block will analyze the signal to generate a computer-understandable event and Event's method `eventListener()` is listening for any event generated. Once it detects the generation of an event, it will instruct Algorithm to call its method `giveEvent()` to pass the event to the block Event. The block Event will parse the event for Condition to evaluate its conditions. After the condition evaluation, if the returned value of this method is false, then no action is taken; if the returned value is true, then it will ask the block Action to call its method to fire the corresponding user interactions to collect the context information.

Based on this design protocol, the system consists of three elements: *Signal (Data) Transportation*, *Algorithm* and *Context Information Collection*. In this project, the modified ECG data will be transported to the PDA by the ECG simulator. The receiving end will parse the digital signal first and then hand it over to the next section – the beat detection algorithm. The algorithm will analyse the incoming ECG data and 'locate' the heart beat with its corresponding time information. The identified heart beat information will be captured by Context collection that will calculate the heart beat intervals. If any irregular heart beat interval is found, the system will ask the user to input the context – ADLs information. The data transportation will be considered.

## **4.4 SYSTEM DESIGN – DATA TRANSPORTATION**

As mentioned earlier in this chapter, the system is composed of three parts: Signal (Data) transportation, algorithm and context information collection. Each followed the design protocol – S-A-E-C-A, and each part communicates with each other through the registered listeners. Therefore the coupling between the modules is very loose, which gives good performance in terms of software design.

This section will describe how the system maximizes the flexibility of the transportation methods, and simulates the real time ECG recording situation. The data resolution, transfer speed and data accuracy are considered, as is how the data is transmitted to the PDA.

### **4.4.1 Transportation structure**

If the requirement of subject mobility is ignored, the ECG signal can be sent through different transportation channels, such as Bluetooth, Wi-Fi, Wimax [102], Zigbee wired network, or serial cable. The aim in this work is to apply Bluetooth for communications that support an ambulatory ECG recording system. However, as identified in the previous discussion and for better adaptation and easier upgrade of the whole system, the transportation software structure is developed with a swappable communication adapter to deal with these issues.

The different communication channels and their management are developed with the



software Pattern called “Abstract Factory” [91]. When a user requests the establishment of a communication channel to exchange messages, it will ask the class `TransManager` that is managing different transportation channel managers such as `BTManager`, `NetManager` and `SerialManager`. Each of these three manager classes can create and manage their respective transportation channels while delegating some functionality to helper classes, so that `BTManager` creates `BTRceiver` and `BTSender` for instance. These channels also implement the java interface `CChannel` that can also be accessed by the user. Figure 17 below illustrates the relationship between the communications classes in the application.

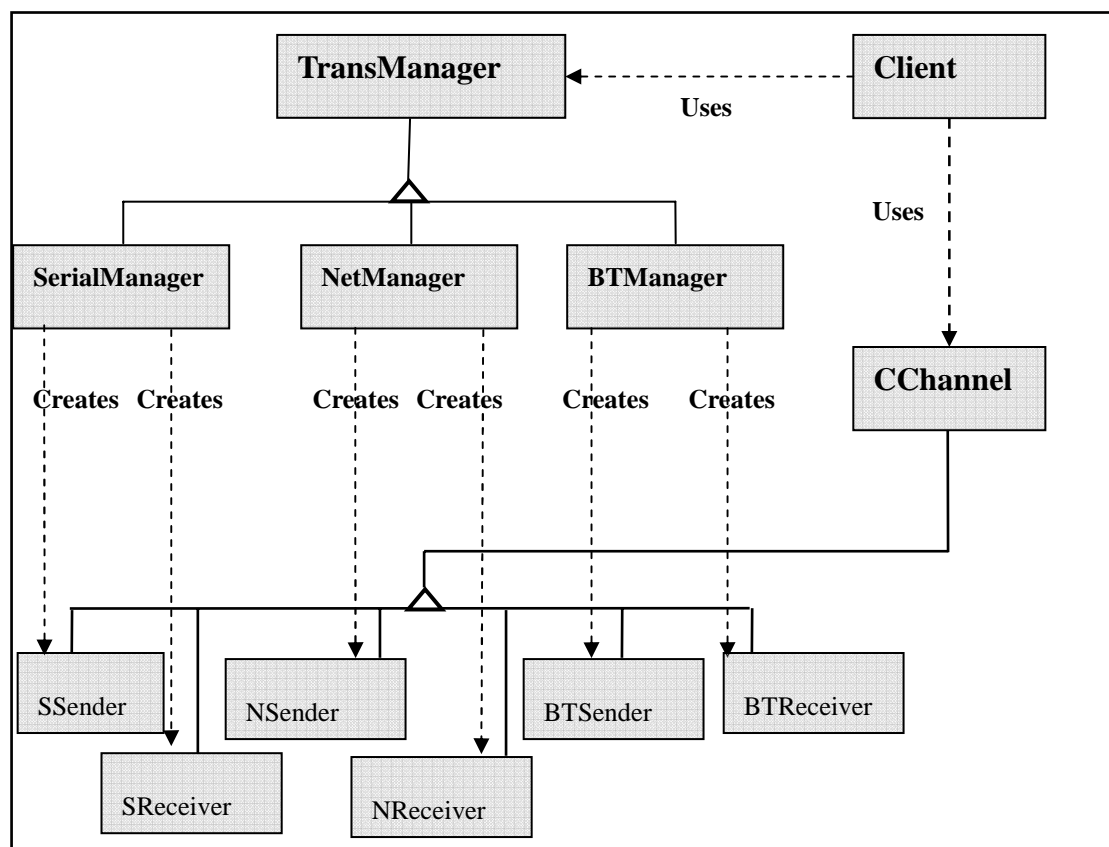


Figure 17: UML Class diagram for the transportation structure

For personal area networks, Bluetooth is the most appropriate wireless medium [6],

but Zigbee could be a future approach. Bluetooth is not necessarily an all purpose solution for wireless transmission of biosignals. For instance, there may occasionally be a loss of bandwidth due to interference resulting in a requirement for multiple re-transmissions of packets [103]. When this is accompanied by limited storage capacity of the sending device, it may result in a delay or even break in the transmitted signal. A break of this sort in the transmission of vital signals such as ECG recordings could be considered life threatening in certain circumstances. Given that the SAECA approach is intended to be applied to other problems in the future it is necessary to consider the flexibility to facilitate future different transportation methods.

#### **4.4.2 Dealing with ECG data**

This project does not employ a real three-lead ECG instrument with Bluetooth capabilities. If it did, a set of values for these parameters would be imposed by the device. In the absence of a real ECG device, the selection of data resolution, transfer speed and data accuracy, before and during the ECG data transportation is left to the author. The selection of appropriate values for these parameters is discussed in the following sections.

##### **4.4.2.1 Data resolution**

In Section 3.7.3, it was stated that a 12-bit ADC is sufficient for ambulatory ECG recording. There is an obvious difference between ambulatory and stationary ECG recording, as there is much more noise in the former. In ambulatory ECG recording,

unwanted high amplitude EMG [104] artefacts arising from muscle movement, and instrument and ambient noise are relatively high. This is particularly true for the measurement of abnormal QRS complex parameters and the detection of ventricular late potentials in patients with a history of ventricular tachycardia [13]. Therefore very high resolution (24-bit ADC) ECG recording with is not necessary for this scenario [92], and 12-bit ADC will suffice for ambulatory ECG recording[105,93]. It is worth mentioning that the downloaded short ECG data has 12-bit resolution over range from 5mV to -5mv at a sampling frequency of 500 Hz. The longer ECG data file (Record 100 in MIT-BIH ECG database) [106] has 11-bit resolution over range from 5mV to -5mv. Sample values thus range from 0 to 2047 inclusive, with a value of 1024 corresponding to zero volts.

#### **4.4.2.2 Data transfer speed**

In the Java code implementation, it will be faster to transfer the data in “integer” or “short” format rather than “double” format. The data on the input or the output stream that is set up by the Java’s virtual machine is only in “byte” format. Java uses 8 bytes to represent a “double” and only uses 4 and 2 bytes for “long integer” and “short integer” respectively.

The sampling frequency is 500 Hz, so the simulator will “sample” 500 samples of ECG data every second. 500 samples in its truncated “integer” type will take 2000 bytes in “long” integer type (or two channel “short” type) or 1000 bytes in “short”

type per channel. When the “*short*” type is employed for a single channel, 8000 bits of data per channel with extra information such as sampling rate, frame number and patient name must be sent to the PDA in one second to maintain real time operation.

#### **4.4.2.3 Data accuracy**

ECG data is transmitted and stored in a frame that corresponds to an array of bytes in Java. It is necessary to ensure that a number of ECG frames arrive in correct order and that they are from the corresponding patient. The accuracy of the medical data is critically important, and the spectrum occupied by the Bluetooth is overlapping with 802.11 technology [6]. At moment, there is no available standard technology to ensure quality of service and data accuracy when interference occurs. Although a frame approach is introduced to increase data accuracy in transmission, it does not solve the interference problem. However, it can recognize patient and ECG data information. This approach is shown in Figure 18.

<b>Patient ID</b>	<b>Sample Rate</b>	<b>Frame Number</b>	<b>Parsed ECG Data</b>
-------------------	--------------------	---------------------	------------------------

Figure 18: An outline of the type of ECG frame used in this work

It has been mentioned in earlier chapters that the “frame” – byte array contains not only the ECG samples, but also contains rudimentary context information such as patient ID, sample rate, and frame number. To ensure the maximum accuracy of the data, patient ID and sample rate has to stay the same during transmission and the frame number has to be incremented frame by frame. For instance, if a patient is

called “Peter” that is in “String” format and the sampling frequency is 500Hz, there are 500 samples in “short integer” format per channel that will be sent to the PDA. We have the following, Peter = 3 bytes; Sampling frequency 500 = 4 bytes; Frame number 0 = 4 bytes; 500 samples (2 bytes per sample per channel) = 1000 bytes per channel or 2000 bytes in all. Therefore in this frame, the first 3 bytes is the patient ID – Peter, the next 4 bytes is the sampling frequency, the next 4 bytes is the frame number and the last 2000 bytes contain the sampled ECG data which have been taken by the simulator from the ECG signal file. As each frame arrives, the receiver inside the PDA will parse this frame first to extract the patient ID, the sampling frequency and the frame number to ensure this frame of data is from the right source to avoid problems caused by interference.

If the frame number of an arriving frame is not in numerical order, the system recognises that frames of ECG data that have been lost during the transmission. Whenever this type of situation happens, the system will stop the transmission first and ask the simulator to send the signals in a new round in case the saved ECG data has been corrupted.

#### **4.4.3 ECG data transportation process**

The ECG data transportation process is composed of two phases, these are sending and receiving. Both of them involve data parsing.

#### 4.4.3.1 Phase one – sending

Before the ECG data is transmitted, the ECG data will be scaled and transformed into “Integer format” shown in Figure 4.4. When the sample is in “short” type, it will be parsed in bytes for sending over to the output byte stream of the Bluetooth serial connection. In order to improve data accuracy, the simulator will also combine the patient ID, the sampling rate and the frame number with the ECG data, as shown in Figure 19.

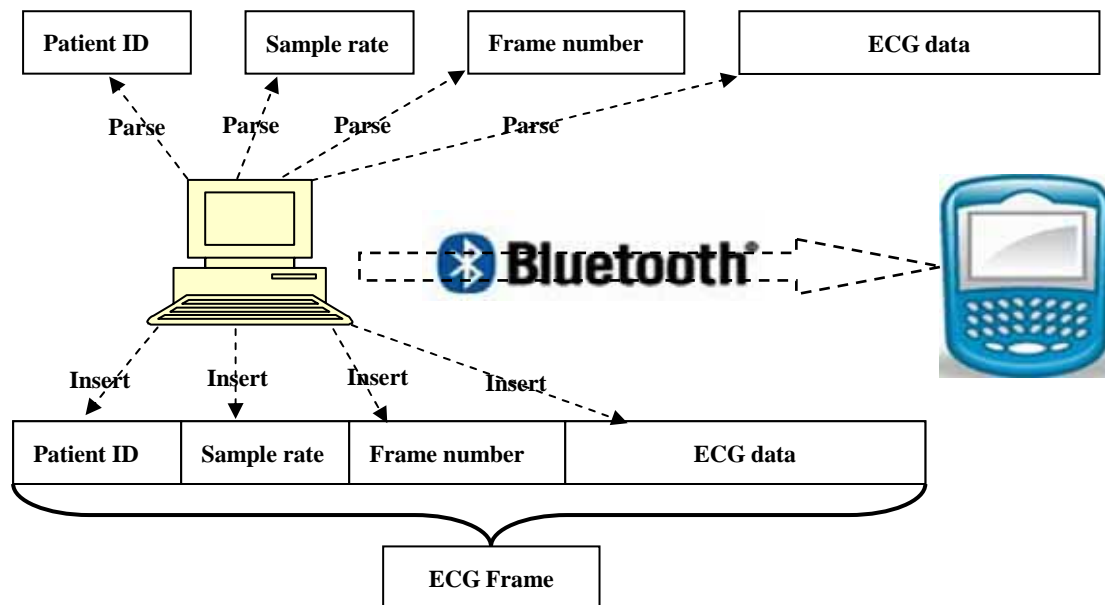


Figure 19: ECG data parsing and frame information insertion

#### 4.4.3.2 Phase two – receiving

When the ECG data arrives at the receiving end-PDA, the system provides an empty frame to store the incoming information. The system is already informed by the user’s input about the Patient ID and the sample rate. Therefore it will take the first 7 bytes and parse them back into “String” and “int” (integer) format. If the parsed results are the same as the user’s input, the system will extract the frame number and the ECG

data. The ECG data will be stored in a separate file for recording purposes and will be parsed back into the original truncated number for further analysis by the algorithm outlined in Figure 20, and discussed in 4.4.2.3.

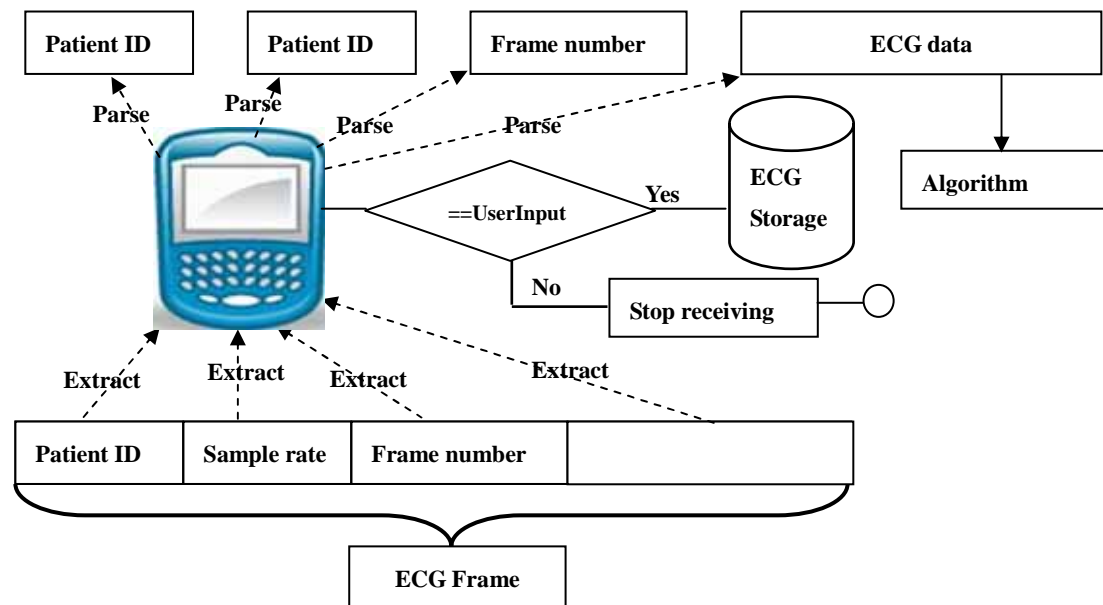


Figure 20: ECG data receiving process in PDA

Further design details of the context information collection and their impact on the system implementation can be found in Chapter 6. The beat detection algorithm is now considered in more detail.

## **CHAPTER 5**

### **ALGORITHM**

This chapter deals with a major part of the system – the beat detection algorithm. This chapter also corresponds to the ODP computation viewpoint of the beat detection algorithm.

#### **5.1 INTRODUCTION**

The heart beat detection algorithm plays a very important role in the system because it processes the stream of sampled ECG data coming from the ECG measuring device to find the heart beat that is considered as an input – the event for the next stage in the process, the Event-Condition-Action rule base. All of this activity will happen on a PDA with relatively limited resources. Therefore the algorithm must exhibit certain characteristics; for instance, it should execute in real time with a low computational load, minimum time and high accuracy so that the whole monitoring system can work effectively. For this research, the algorithm is dealing with ambulatory ECG data, so noise levels will be higher than those of the static recordings, and high detection accuracy is essential.

The principle objective for the algorithm is to find the location of the R wave in the ECG data frame (see Chapter 4) transferred from the ECG simulator, so the heart beats are located in the ECG data frame. Therefore beat events are generated for the



Event – Condition – Action stage.

Kohler [107] describes a series of QRS detection algorithms and categorizes them from a number of different perspectives, such as whether they were tested under a standard ECG database like MIT-BIH or whether the computational load is low or it has high detection accuracy (“sensitivity”). A review by the author on the results of Kohler’s work with initial implementations in MATLAB of the more promising algorithms and a subsequent investigation of the workload associated with making the transformation from MATLAB to Java, yielded a ‘short list’ of possible algorithms for this project. These algorithms are shown in Table 3.

	<b>Standard Database</b>	<b>Computation Load</b>	<b>Java Transformation</b>	<b>Sensitivity</b>
<b>Filter Banks [96]</b>	<b>MIT-BIH</b>	<b>Low</b>	<b>Medium</b>	<b>99.56%</b>
<b>Zero-Crossing [80]</b>	<b>MIT-BIH</b>	<b>Low</b>	<b>Low</b>	<b>99.7%</b>
<b>Pan &amp; Tomp[94]</b>	<b>MIT-BIH</b>	<b>Medium</b>	<b>High</b>	<b>99.3%</b>
<b>Differentiation[108]</b>	<b>MIT-BIH</b>	<b>Unknown</b>	<b>Unknown</b>	<b>92%-100%</b>

Table 3: Detection algorithm comparisons

The algorithms listed in Table 3 were implemented in MATLAB and the Zero-crossing algorithm was transformed into a Java language representation. The only realistic competitor for the Zero-crossing approach is the Filter Banks approach, but the MATLAB code for the Filter Banks approach is much more complex than for

zero-crossing approaches.

## **5.2 ZERO – CROSSING ALGORITHM INTRODUCTION**

A method of QRS detection using zero crossing counts was proposed by Kohler, Henning & Orglmeister [80]. In this approach the ECG signal is first filtered and then a high frequency triangular sequence is added to the filtered signal. The amplitude of the high frequency component is less than that of the QRS complex, and therefore the number of zero crossings is observed to be high in non-QRS segments and low during QRS segments.

Figure 21 illustrates the zero-crossing method and the following panels are the processing procedures of the algorithm.

Panel i ) the original ECG signal

Panel ii ) the band pass filtered ECG signal

Panel iv) the bandpassed ECG signal after the non-linear transformation

Panel iv) the high frequency sequence

Panel v ) the signal after adding the high frequency sequence

Panel vi) zero crossing count (exponential window).

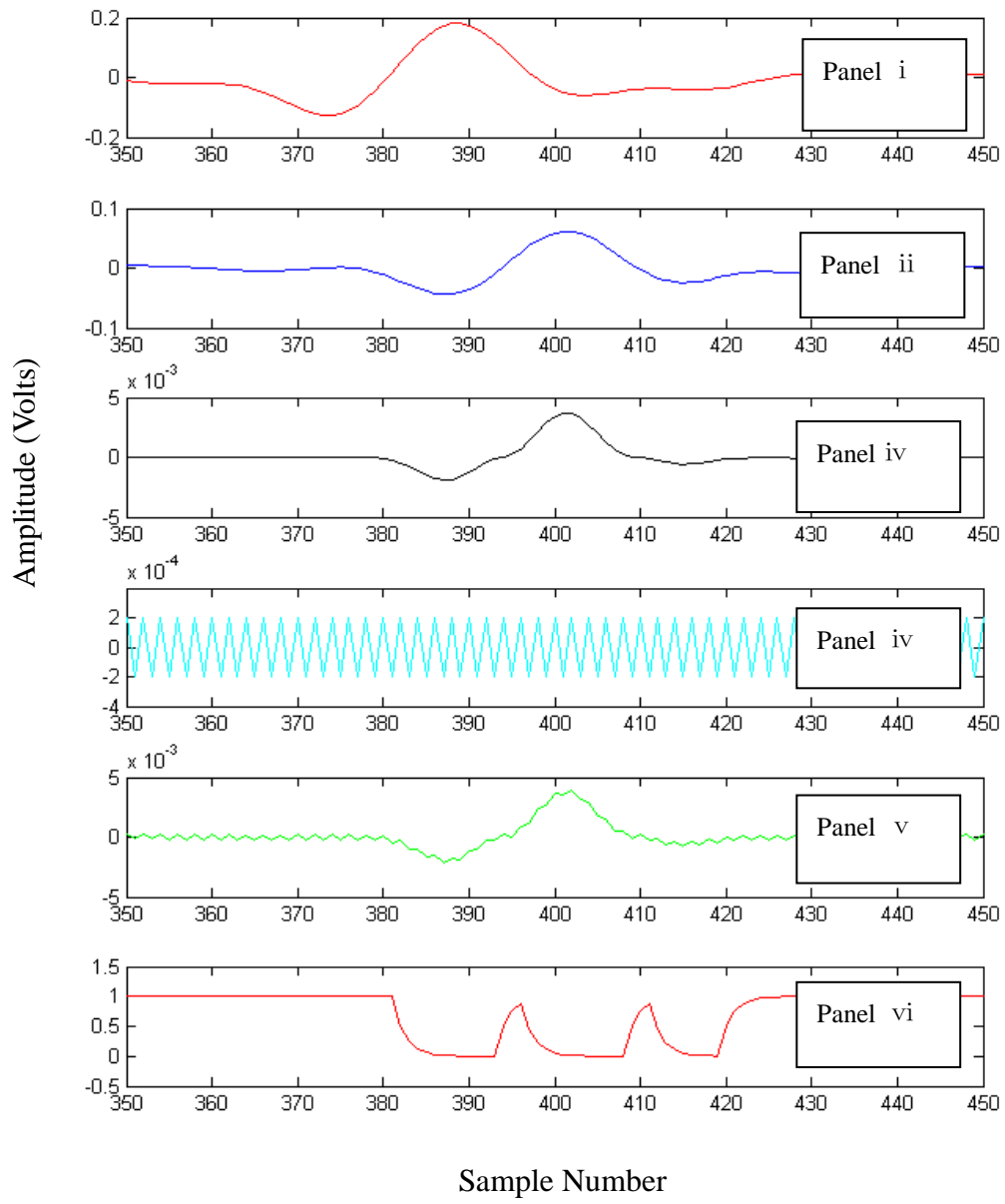


Figure 21: The signals in the feature extraction stage.

Following the steps illustrated in the above figure, it is reasonable to detail this method in the following panels.

### 5.3 ALGORITHM ILLUSTRATION

In **Panel i of Figure 21**, the original ECG signal is introduced, and components of the ECG signal can be composed of frequencies which extend beyond 40 Hz, whereas the P and T waves usually have frequency components only extending to 10 Hz [67]. Problematic ECG signals may contain baseline drift, noise and artifacts.

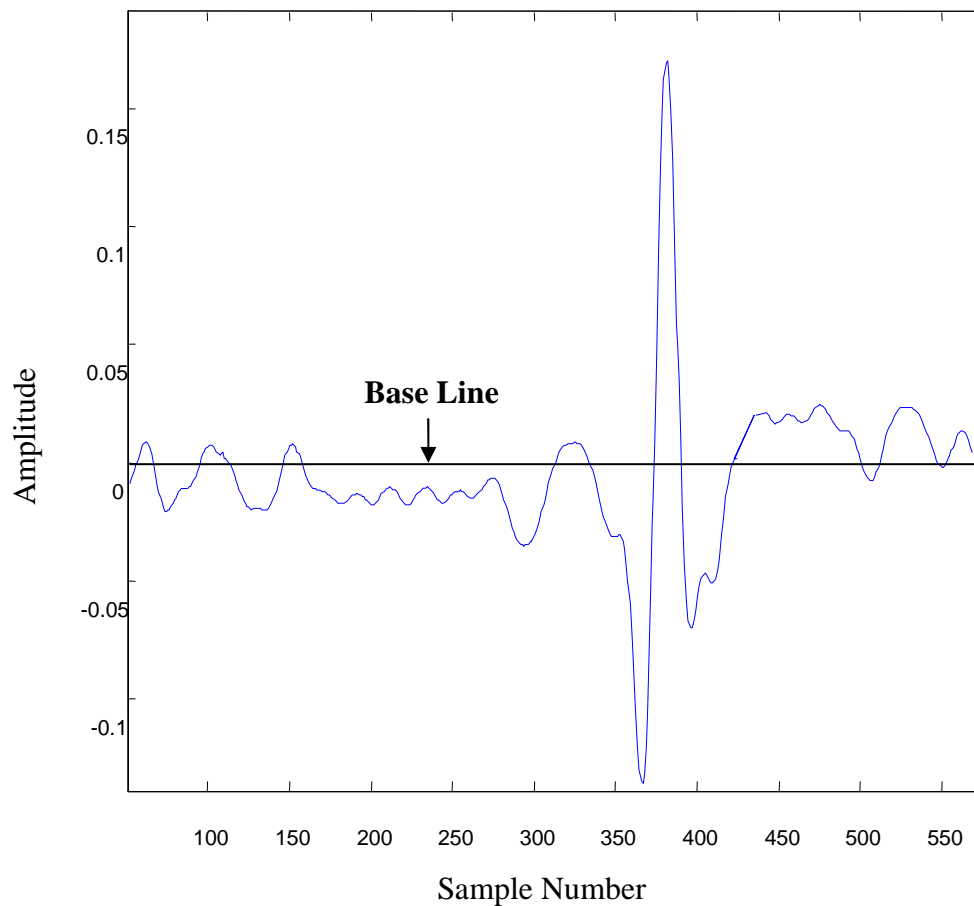


Figure 22: A portion of ECG signal that contains baseline drift and noise

Due to these spectral characteristics of the ECG components, it is reasonable to filter the ECG signal first in order to attenuate the mean, the P and T-wave, and the high

frequency noise.

In **Panel ii of Figure 21**, the original ECG signal is passed through a band pass filter with a linear phase response and cut-off frequency of 18-35 Hz to increase the signal quality. The difference equation for a 26<sup>th</sup> order linear phase FIR Filter is shown in Equation 5.1

$$y(n) = \sum_{m=0}^{26} h(m) * x(n - m) \quad \text{Equation 5.1}$$

where  $h(m)$  is impulse response and  $x(n-m)$  is the inputted ECG data delayed by  $m$  indices.

A graphical representation of the filter is illustrated in Figure 23, where  $x(n)$  is the inputted ECG signal, the term  $h(m)$  represents the coefficient of the  $m$  tap of the FIR filter and  $y(n)$  the filter output at the  $n$  sampling instant respectively.

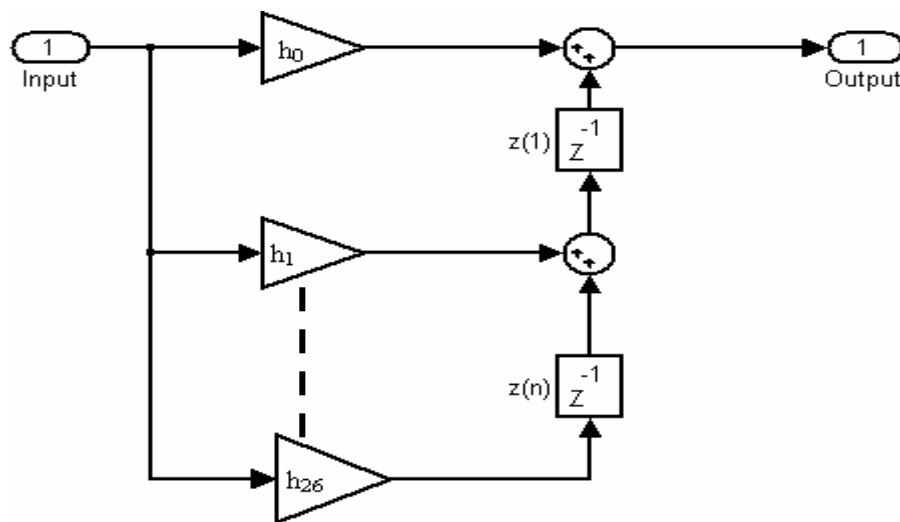


Figure 23: A 26<sup>th</sup> order linear phase FIR filter

When compared to the original ECG signal, the bandpassed ECG signal has less variations in the non-QRS segment, as shown in Figure 24, where the blue graph represents the filtered ECG signal. Although the amplitude of the filtered QRS complex is reduced, the waveform of the original ECG signal is preserved.

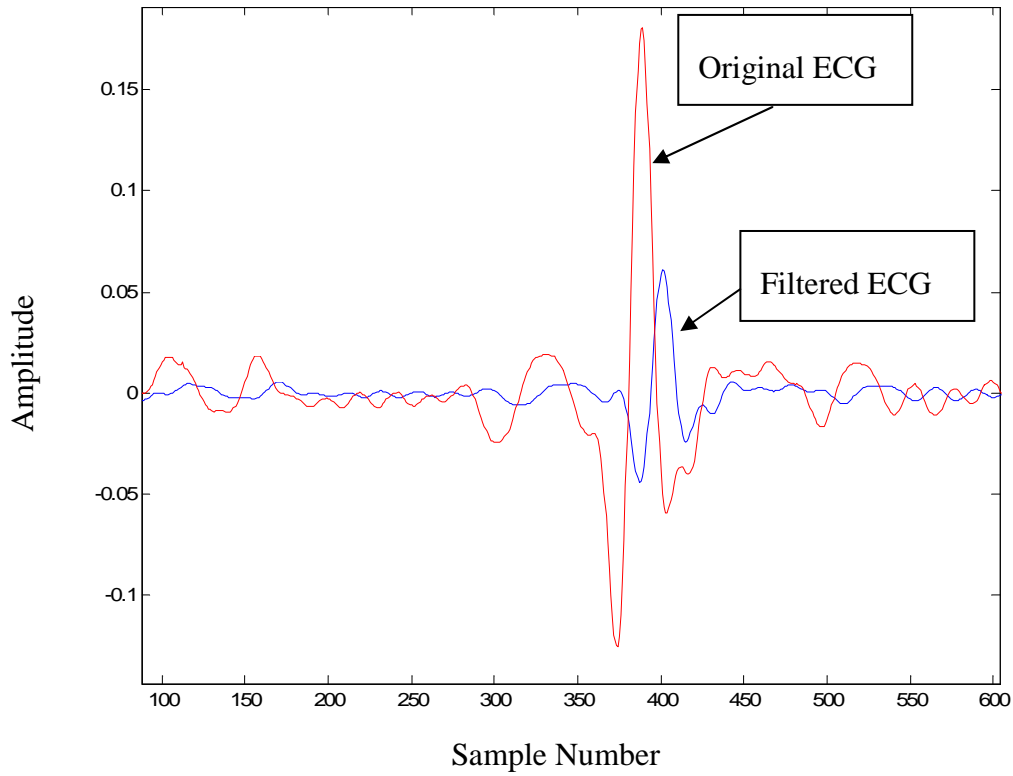


Figure 24: Comparison between filtered and original ECG signal

In **Panel iv of Figure 21**, a further increase in the quality of the bandpassed ECG signal is achieved by the application of Equation 5.2 to the filtered digital signal.

$$y(n) = \text{sign}(x_f(n)) \cdot x_f^2(n) \quad \text{Equation 5.2}$$

where  $x_f(n)$  denotes the filtered ECG signal,  $\text{sign}(\ )$  is a mathematical function which returns 1,0,-1 depending on the inputted values.

Therefore the sign of every filtered ECG sample is preserved. The squared function will increase the relative difference between the QRS complex and the non-QRS segment. Then the non-QRS segment of  $y(n)$  will become flatter than the band-passed and filtered ECG data.

Figure 24 illustrates the difference between the filtered ECG signal and the non-linear filtered ECG signal. As discussed in Chapter 2, An ECG signal contains P, Q, R, S and T waves, in addition to noise. It is shown that two graphs have different amplitudes and the non-QRS part of the further transformed ECG data has fewer variations from the baseline. Also the P and T wave are highly attenuated but not eliminated as shown in Figure 25.

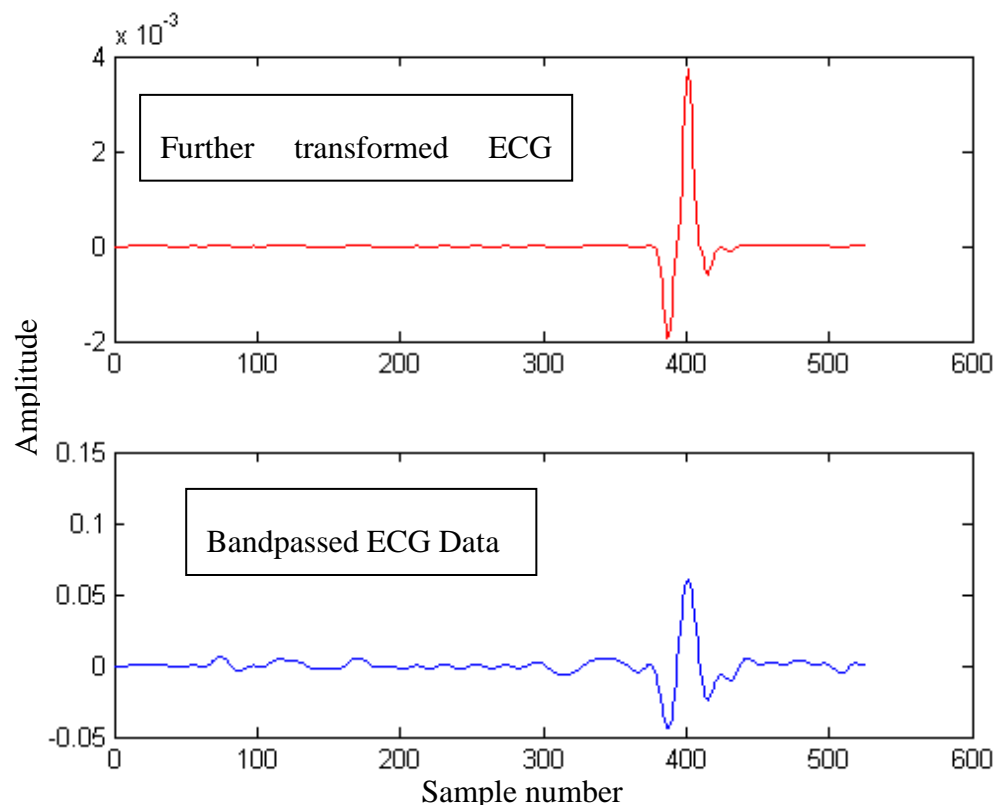


Figure 25: Comparison between Filtered and Non-linear filtered ECG signal

As the signal  $y(n)$  is from the bandpass filter, high frequency oscillations are strongly attenuated. It is necessary to add a high-frequency sequence to facilitate the occurrence of zero crossing.

In **Panel iv of Figure 21**, the high frequency sequence  $b(n)$  is introduced as described by Equation 5.3.

$$b(n) = (-1)^n K(n) \quad \text{Equation 5.3}$$

$b(n)$  is a high-frequency sequence, the  $(-1)^n$  term changes sign at every consecutive sample.

Adding the filtered signal to the high frequency sequence yields Equation 5.4

$$z(n) = y(n) + b(n) \quad \text{Equation 5.4}$$

$K(n)$  is determined from the magnitude of the signal  $y(n)$ .

$$K(n) = \lambda_K K(n-1) + (1 - \lambda_K) |y(n)| \cdot c \quad \text{Equation 5.5}$$

where  $\lambda \in (0;1)$  is a ‘forgetting factor’.

This equation results in a weighted sum between the  $K(n)$  and  $y(n)$  components of the signal. When the forgetting factor approaches 1, the amplitude of  $K(n)$  will become



smaller, resulting in a smaller amplitude of  $z(n)$ . Therefore the value of this forgetting factor can influence the amplitude variation of  $z(n)$ . If the noise level on the ECG signal is high, the forgetting factor should be set closer to 1 by the technical user such that the noise level will not be amplified, so that the probability of detecting false beats will be reduced.

The constant gain,  $c$ , plays an important role here. If  $c$  is defined to be bigger than the absolute maximum value of  $z(n)$ ,  $z(n)$  will be dominated by the signal  $b(n)$ . If  $c$  is too small, the oscillations will not be obvious and so the zero crossing detection will be less effective. Through observation, the author has found that practical values of  $c$  tend to be in the range of  $\frac{1}{100}$  to  $\frac{1}{10}$  of the maximum amplitude of the non-linear filtered ECG signal, and when in this range zero-crossings can be obtained. Following this processing of the signal, **Panel v of Figure 21** results in a signal as shown in Figure 26, the high-frequency oscillations are carried away by the suddenly changed amplitude of the ECG waveform – QRS complex. A, B and C in Figure 26 are corresponding to A, B and C in Figure 27.

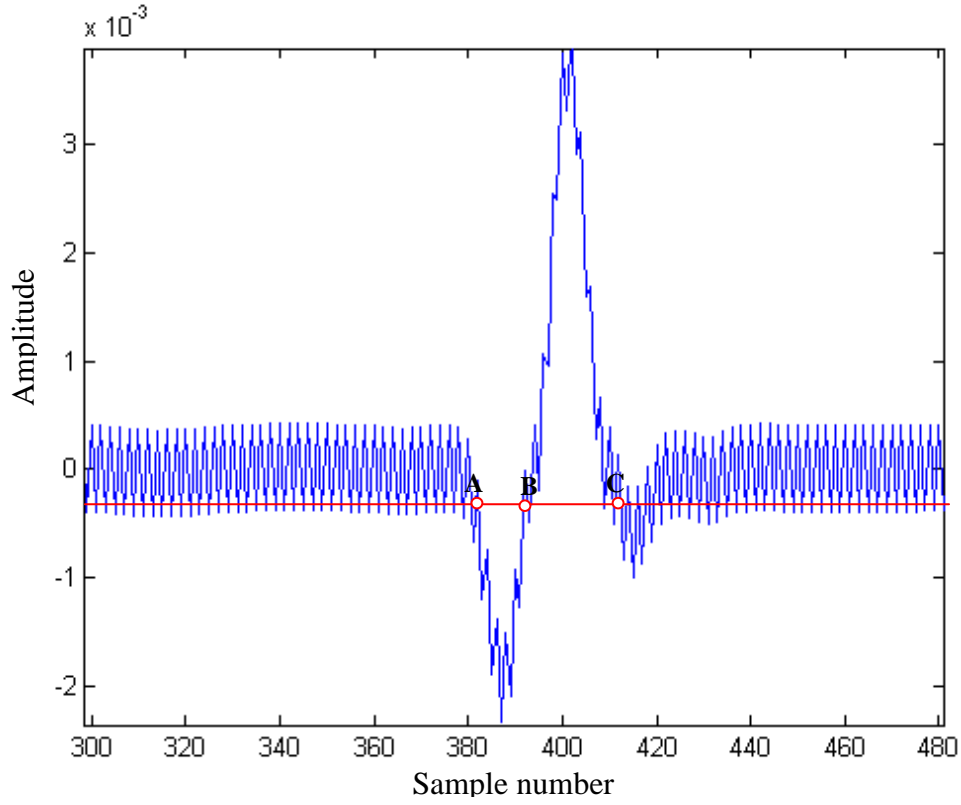


Figure 26: Add high frequency sequence into the signal

In **Panel vi** of **Figure 21**, for detecting Zero-Crossings, Kohler recommends a term  $d$  where:

$$d(n) = \left| \frac{\text{sign}[z(n)] - \text{sign}[z(n-1)]}{2} \right| \quad \text{Equation 5.6}$$

The possible values for the  $\text{sign}[z(n)]$  are  $-1$ ,  $0$ , and  $1$ , therefore the possible values from Equation 5.6 above are  $0$ ,  $0.5$ ,  $1$ . The feature signal is calculated as:

$$D(n) = \lambda_D * D(n-1) + (1 - \lambda_D) * d(n) \quad \text{Equation 5.7}$$

where  $\lambda_D$  is a forgetting factor with values between  $0$  and  $1$ .

With respect to a processor or direct hardware implementation, this calculation has the advantage of being more easily adjustable while consuming less memory [32]. The forgetting factor is configured to be adjustable for the difference between the feature signal  $D$  and term  $d$  according to different heart and context information for the patient. The feature signal is plotted in Figure 27.

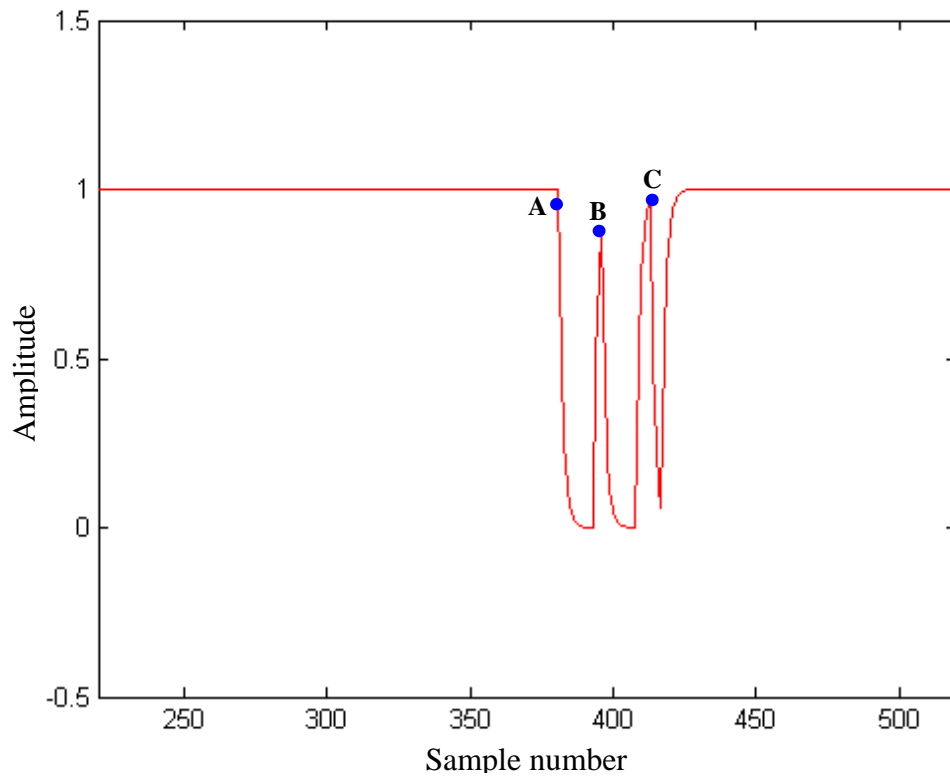


Figure 27: Feature signal with multiple beat events detected

The threshold for the feature signal is defined in Equation 5.8.

$$\Theta(n) = \lambda_{\Theta} * \Theta(n-1) + (1 - \lambda_{\Theta}) * D(n) \quad \text{Equation 5.8}$$

where  $\lambda_{\Theta}$  is a forgetting factor again which varies from 0 and 1.

This forgetting factor corresponds to a weighted sum that can be adjusted due to

different situations. For instance, if the threshold value needs to be higher,  $\lambda_{\theta}$  will be closer to zero and vice versa. The following plot illustrates both the feature signal and the threshold values with the relationship defined according to Kohler [32]: “When  $D(n)$  falls below the threshold an event is detected”,

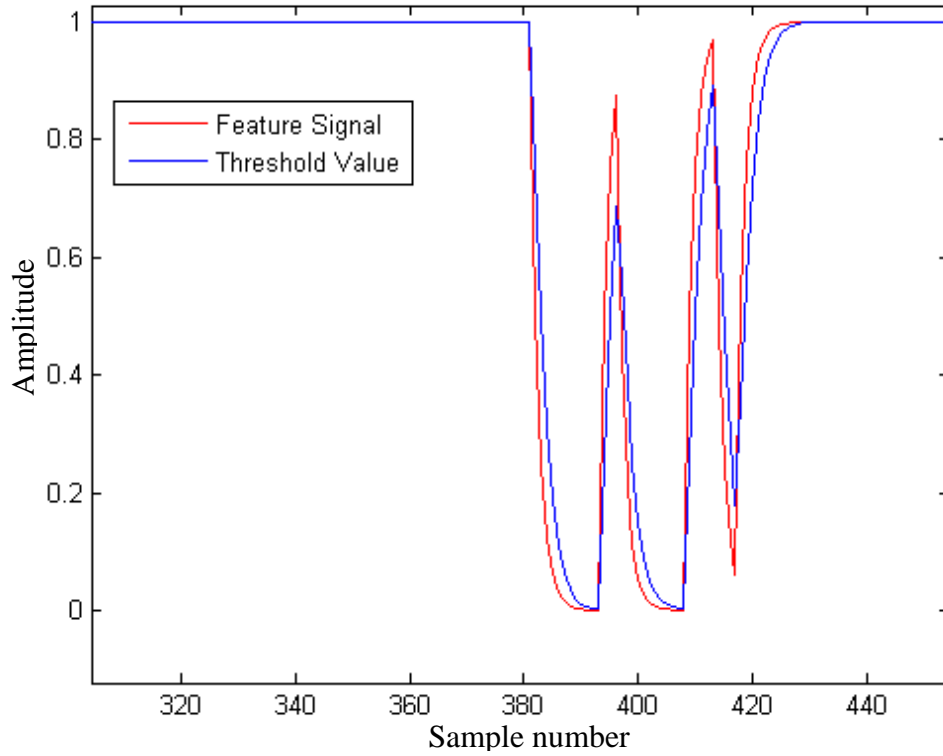


Figure 28: Threshold and multiple-event detection

From Figure 28, it is seen that sometimes the feature signal has more than one peak. In order to prevent multiple false beat detection events from occurring, the **minimum elapsed time** between 2 possible beats is defined based on the chosen sampling frequency. If the sampling frequency is 500 Hz, this interval is defined for 100 samples, equivalent to 0.2 seconds. If there are 3 real heart beats within 0.2 seconds as shown in Figure 26, heart rate would be 900 beats in a minute, which is not possible

for human subjects. The **minimum elapsed time** between 2 possible beats is varied according to sampling frequency, different heart and context information. This issue will be discussed again in section 6.3. Figure 29 gives an overview of the zero-crossing algorithm.

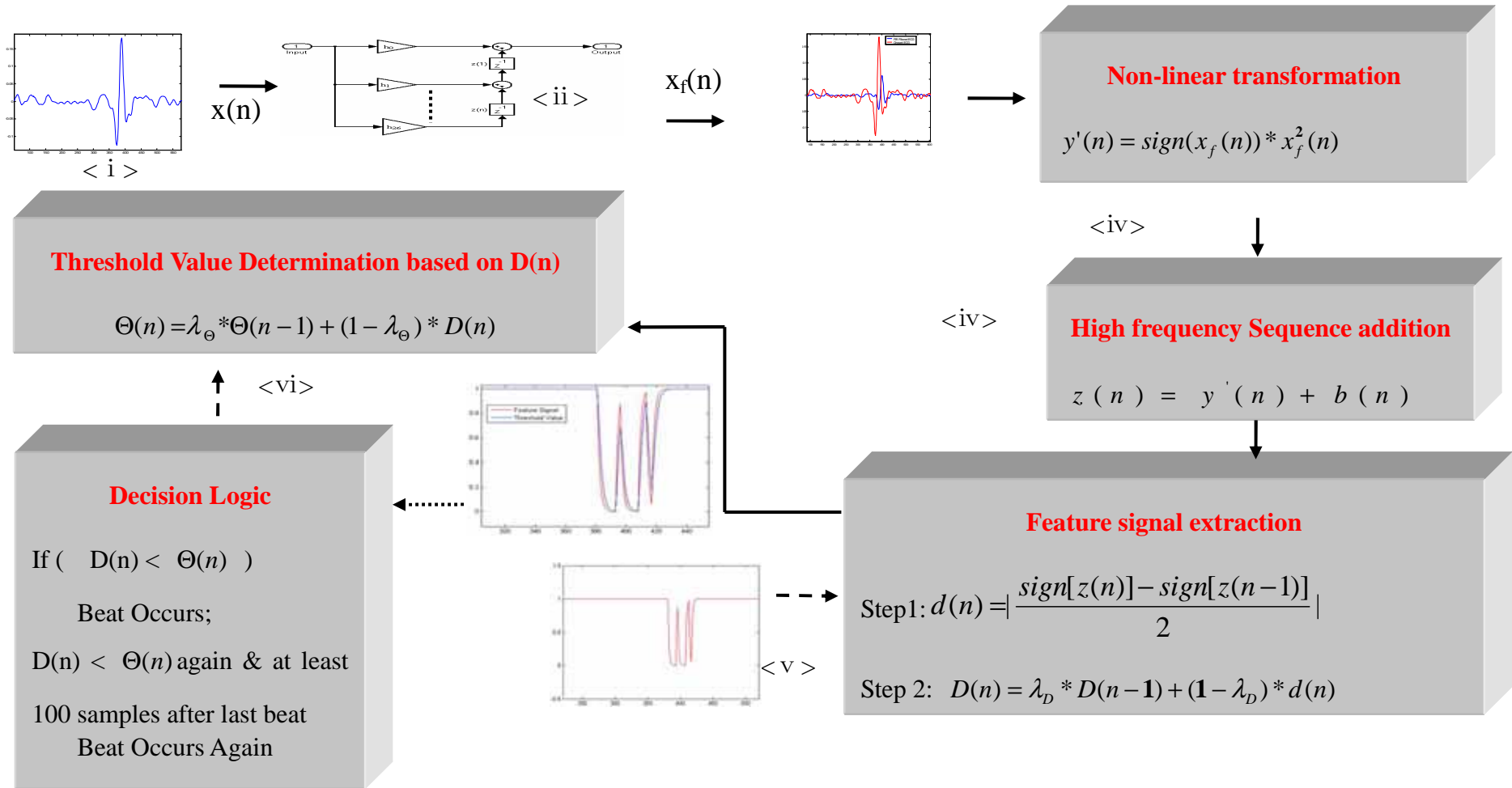


Figure 29: Algorithm overview

## 5.4 FLOWCHART OF THE ALGORITHM FOR JAVA IMPLEMENTATION

The zero-crossing algorithm [80] was first implemented in MATLAB (**Appendix A**) and then developed in Java (**Appendix A**). The flowchart diagram for this procedure is presented in Figure 30.

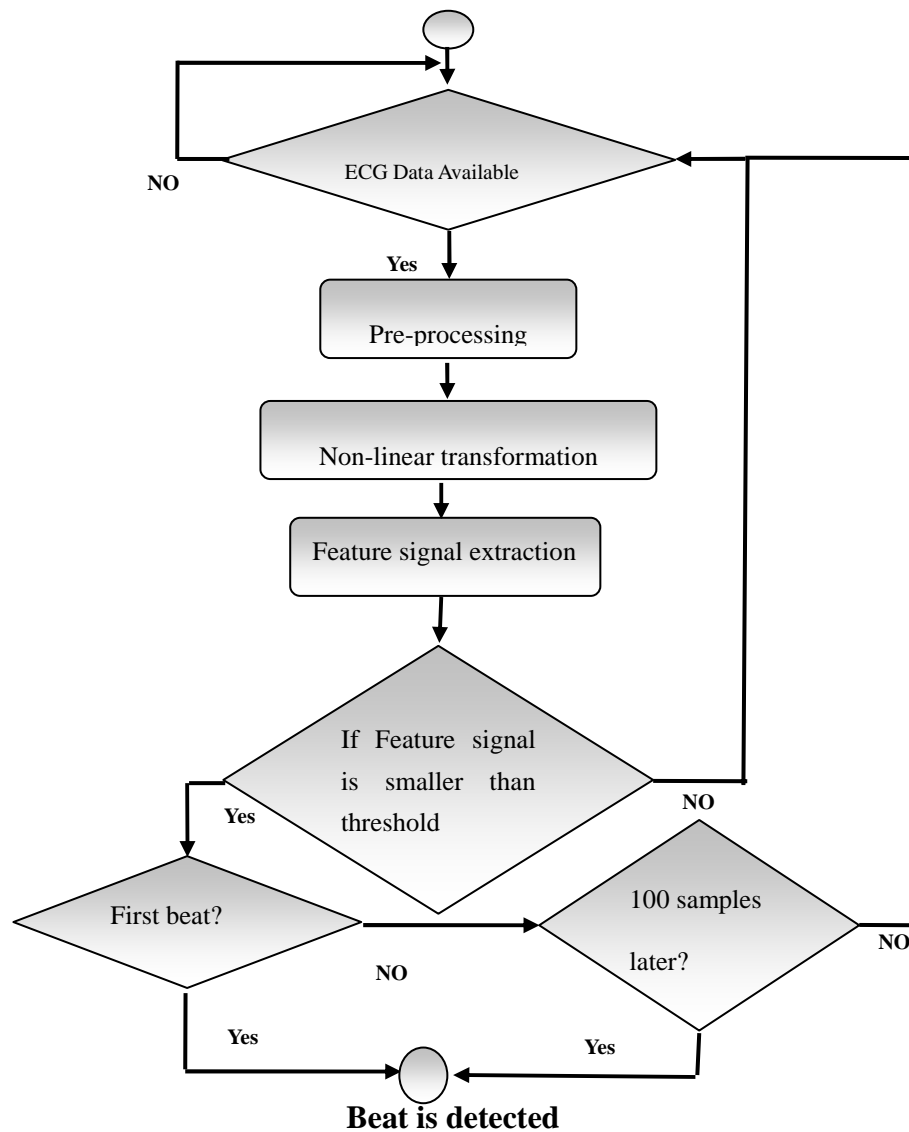


Figure 30: Flowchart of the zero-crossing algorithm

This concludes the description of the computation view point of the detection algorithm. Data transportation and context collection is presented in Chapter 6.



## CHAPTER 6

# DATA TRANSPORTATION AND CONTEXT COLLECTION

This chapter follows the ODP's computation viewpoint to further describe the data transportation and the context information collection module. In previous chapters the technique used to deal with ECG data issues was presented and the context collection module was developed using the Event – Condition – Action production rules approach, which ensures that this process is only loosely coupled with the Algorithm module such that no polling process exists in the system. Therefore the memory usage of the PDA is reduced.

### 6.1 DATA TRANSPORTATION

The data transportation has two roles: on the simulator side, it is to parse and send the ECG data; on the PDA side its role is to receive and parse the ECG data.

#### 6.1.1 Simulator - sending

In Figure 6.1 it is shown that the simulator contains three objects: `ECGInfo`, `Framer` and `BluetoothManager`. The `Framer` object will instantiate `BluetoothManager` and `ECGInfo` objects, and retain the corresponding references. The different roles played by each object are detailed in Table 4.

Object Name	Responsibilities
BluetoothManager	Sets up the serial connection between the simulator and PDA
ECGInfo	Parses the ECG data
Framer	Creates a ECG frame and send it to the PDA

Table 4: The roles of objects in the simulator

As indicated in Table 4, when the Bluetooth serial connection is set up and `ECGInfo` has finished parsing the ECG data from a file, the *Framer* object will create ECG frames that contain ECG data and other information such as the patient ID, the sampling rate and the frame number, and send the frame to the PDA.

The overview of the whole process on the sender side (Simulator) is illustrated in Figure 31, and its individual sub-processes are labelled.

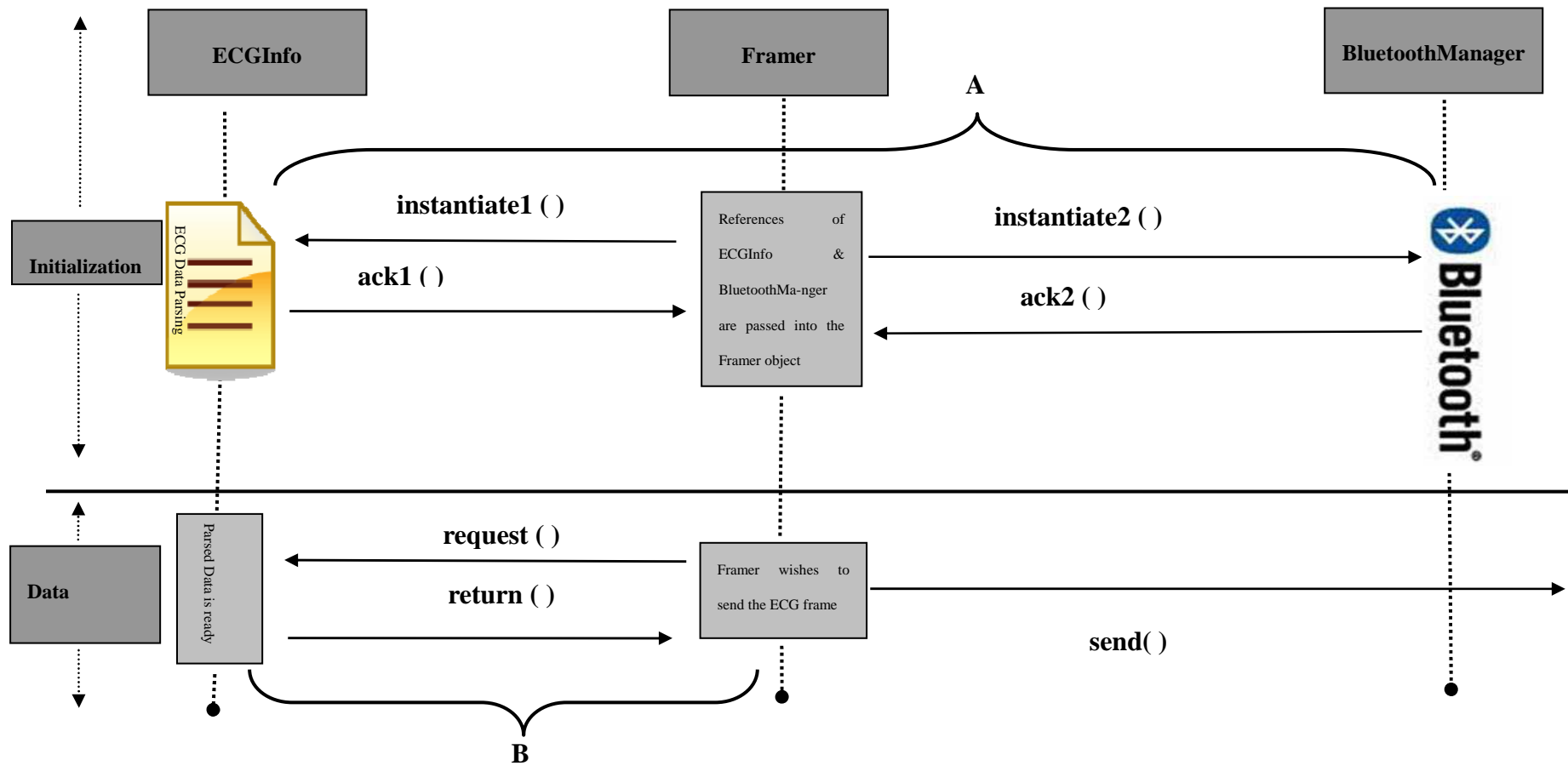


Figure 31: Simulator work process

Step A is “Initialization”; whenever the `Framer` object is created, it will first instantiate an `ECGInfo` object which will parse the file’s data format into bytes and transfer the entire contents of the ECG file into frames (i.e 500 samples/frame) ready for the `Framer` object’s later request. Then an acknowledge is sent back to the `Framer` object. Therefore the *Framer* object has the reference for the `ECGInfo` object and also indicates that the ECG data is ready to be sent.

The `Framer` object will also instantiate the `BluetoothManager` object that will create the Bluetooth serial stream for data transportation. The object instantiation process is then complete.

When the instantiation process is finished, step B, the “Data transportation - sending” step begins. The *Framer* object will send a request to `ECGInfo` for the ECG frames, and wrap the data in its `send( )` method, which adds extra information into this frame. Because the `Framer` object instantiated the `BluetoothManager` object, it has the reference of the Bluetooth serial channel, so it will pass the ECG frame onto the output stream that has been set up by the `BluetoothManager` object for the PDA to receive.

### **6.1.2 PDA - receiving**

On the receiver side, the *BluetoothManager* and *DataReceiver* objects are waiting to deal with the incoming ECG data.

At the initialization stage, `BluetoothManager` not only initialises itself but also the `DataReceiver` object such that `DataReceiver` will copy the data that is stored in the temporary buffer of `BluetoothManager`. This buffer has size of 2013 bytes that means the incoming ECG data in “long integer” type, which can represent 2 channels of ECG data in “short integer” type. The ECG data takes 2000 bytes, the patient name takes 4 bytes, the sample rate 3 bytes, and the frame number 4 bytes. The size of this buffer is subject to the frame setup of the simulator side. If those parameters such as the patient name, the sample rate, the frame number, and the ECG data have changes for their own size on the simulator side, the receiving buffer has to be exactly the same size to match the simulator side. Whenever the setup is ready, if the size of the buffer does not match the incoming ECG frame, it means that the ECG frame is from another source or the ECG frame has been corrupted during the transmission. This situation rarely occurs. If it happens, the system needs to be shut down completely and a brand new transmission commenced. Repeated re-sending of the ECG data delays the beat detection so that the system could be confused for the heart beat intervals if any serious events happen.

`BluetoothManager` implements `SerialPortEventListener`, so whenever ECG data is available on the Bluetooth serial input stream and the size of its temporary data buffer matches the size of the incoming ECG frame, it will store them into that buffer, and `DataReceiver` will copy this buffer and store them into its own array, which is illustrated in Chapter 3, Figure 13.

DataReceiver has a responsibility to check whether this frame is from the right patient and all the ECG frames has been received. When the ECG frame arrives into the buffer of DataReceiver, it will parse this frame. It will extract the information about the patient ID, the sample rate and the frame number. If the patient ID or the sample is not the same as the one saved already inside the DataReceiver, it will terminate the whole process immediately, because the system is receiving the wrong information. If they are the same, DataReceiver will also store the extracted array of numbers into a file for unexpected data retrieval. The process is illustrated in Figure 32.

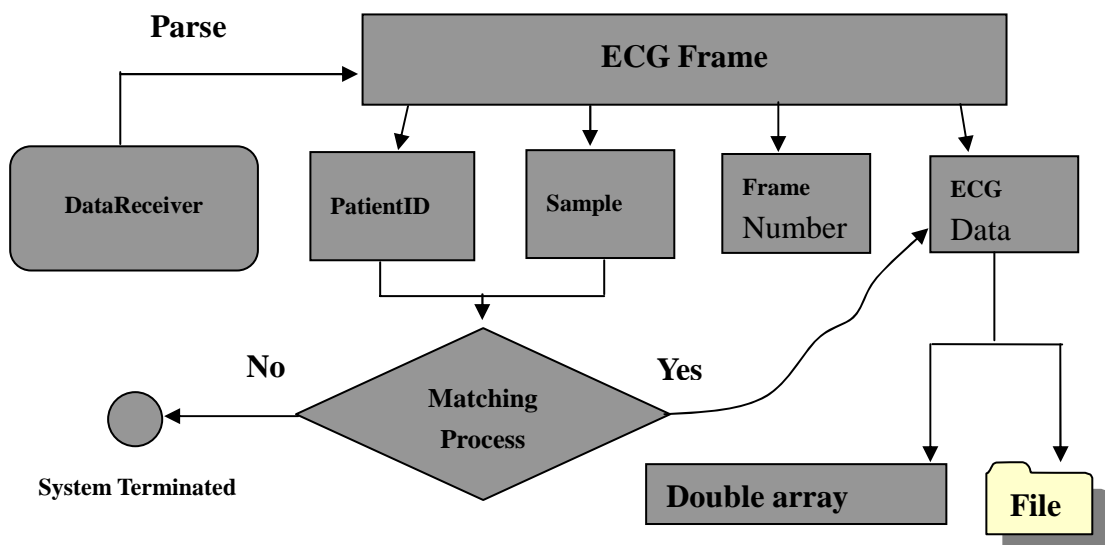


Figure 32: The work of DataReceiver

If the ECG data is from the right patient, the ECG data will be parsed into double format, then stored into a double array for ECG data processing by the ECG beat detection algorithm that is detailed in Chapter 5.

## **6.2 CONTEXT REVIEWED**

In Section 1.1, the author's definition of the primary context type for this project is given. This research does not seek to detect the context signals intelligently as in Healy's work [13] but to manually record this context information. Ideally this information would be stored as annotation tags on the ECG signal for better diagnosis afterwards but for simplicity, it is stored textually in a separate temporal log file.

## **6.3 CONTEXT COLLECTION – GENERAL STRUCTURE**

This research aims to investigate whether real-time context collection through user interaction is technically feasible in a Holter monitor system. Consequently this functionality is a core part of this system. As described earlier, this part is developed using the Event – Condition – Action approach. The whole system is based on this approach. Whenever the ECG data arrives in the PDA, the data communication part will store and parse them, and then pass them into the next stage – the Algorithm module. The Algorithm module will analyse the incoming blocks of ECG data to “locate” the heart beat and attach the corresponding time information to it. Now the context collection part is introduced.

The first stage of this context collection is the Event that has a beat event listener. Once the beat event is generated, the beat event listener will notify the Event part to store this beat into an array of beat Events. Whenever there are two beat Events in the array, the time interval between neighbouring beats is calculated. The condition part

also has a listener that listens to the calculated result from the Event part. A healthy adult should have a time interval between heart beats of around 800 milliseconds (75beats/min) [109]. So when the time interval has become much smaller or larger than 800 milliseconds, this condition part will place the calculated time interval into its personalised rule engine for evaluation to see if it is too low or too high. When evaluation is finished, the condition is alerted that the time interval is too low or too high, as a result it will call the corresponding user interface to ask the user what they are doing at that moment to gather the context information.

This context query will be stored for further ECG analysis, and Figure 33 illustrates this process.

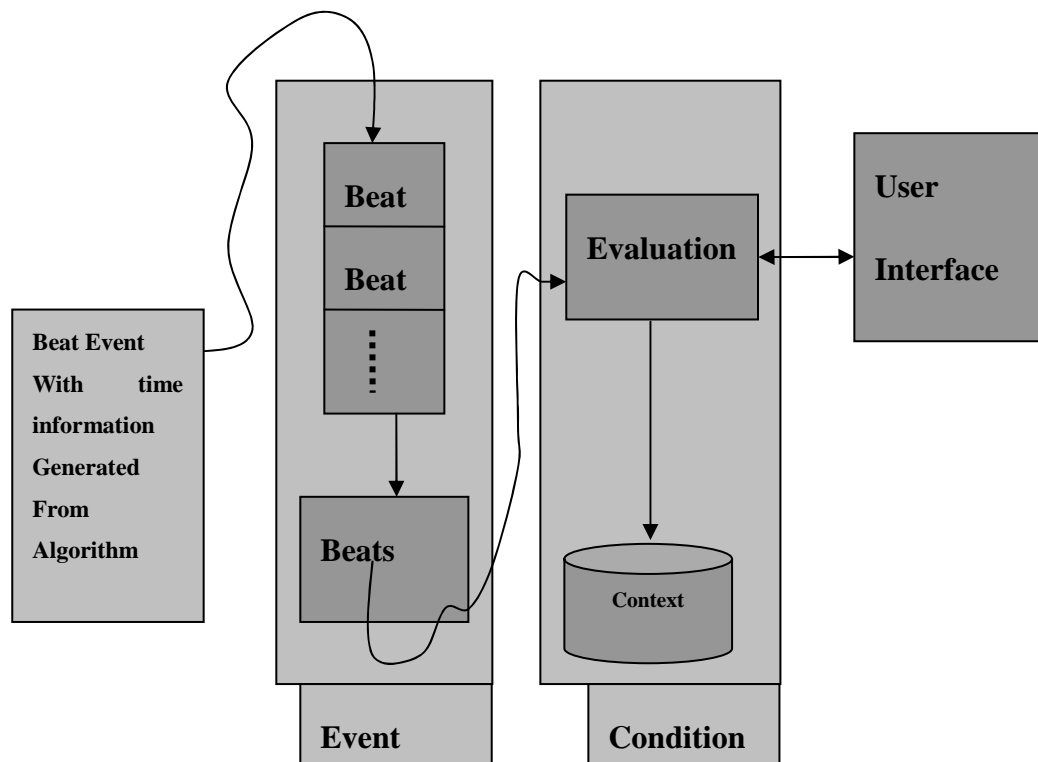


Figure 33: Context collection process



## 6.4 CONTEXT INFORMATION COLLECTION

Based on Figure 34, the context collection process can be pseudo coded in the following flowchart, and every part of the context collection will be detailed in this section.

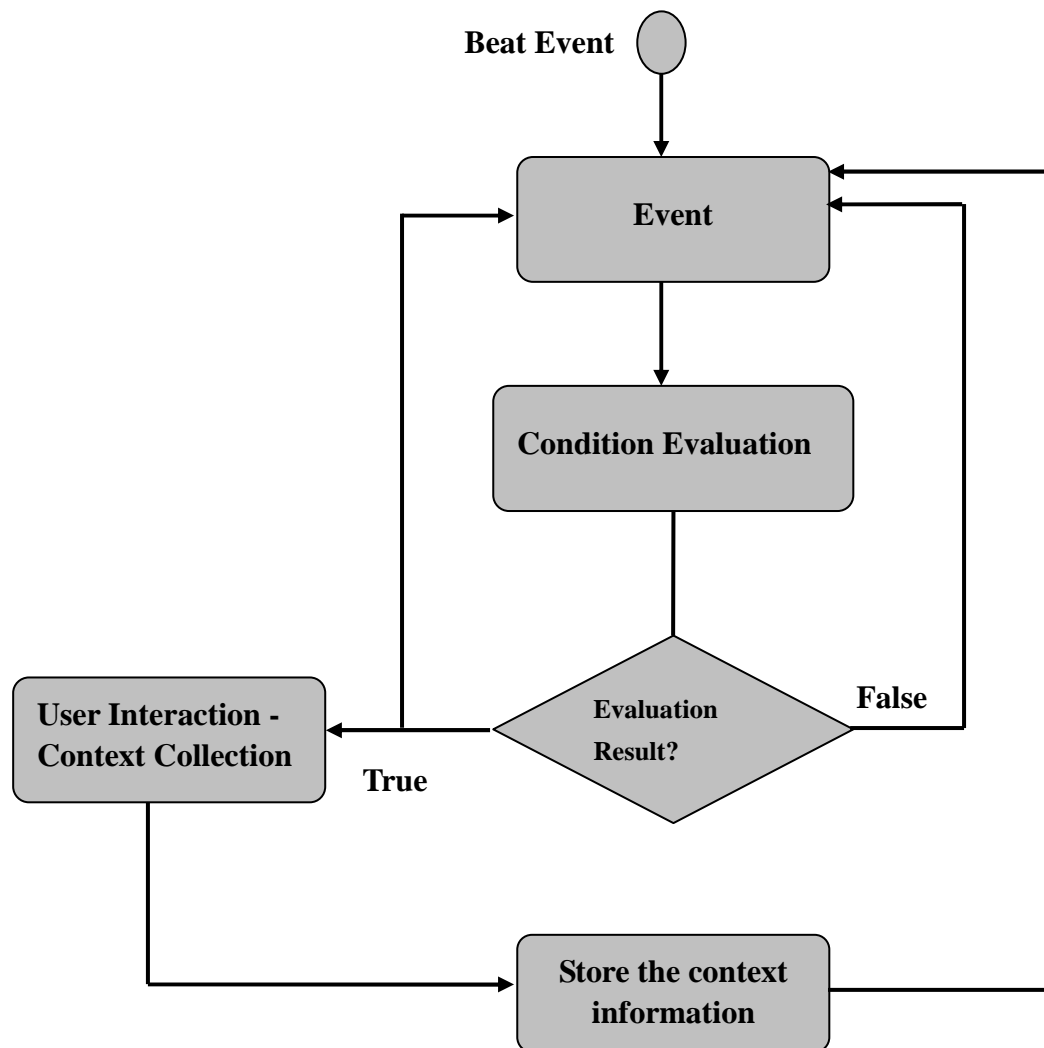


Figure 34: A flowchart of the context collection process

### 6.4.1 Event process

This Event part places the detected beat with its corresponding time information into an array. This array follows the rule “First In / First Out” – FIFO. There are various

calculations that can be carried out on this array as shown in Figure 35.

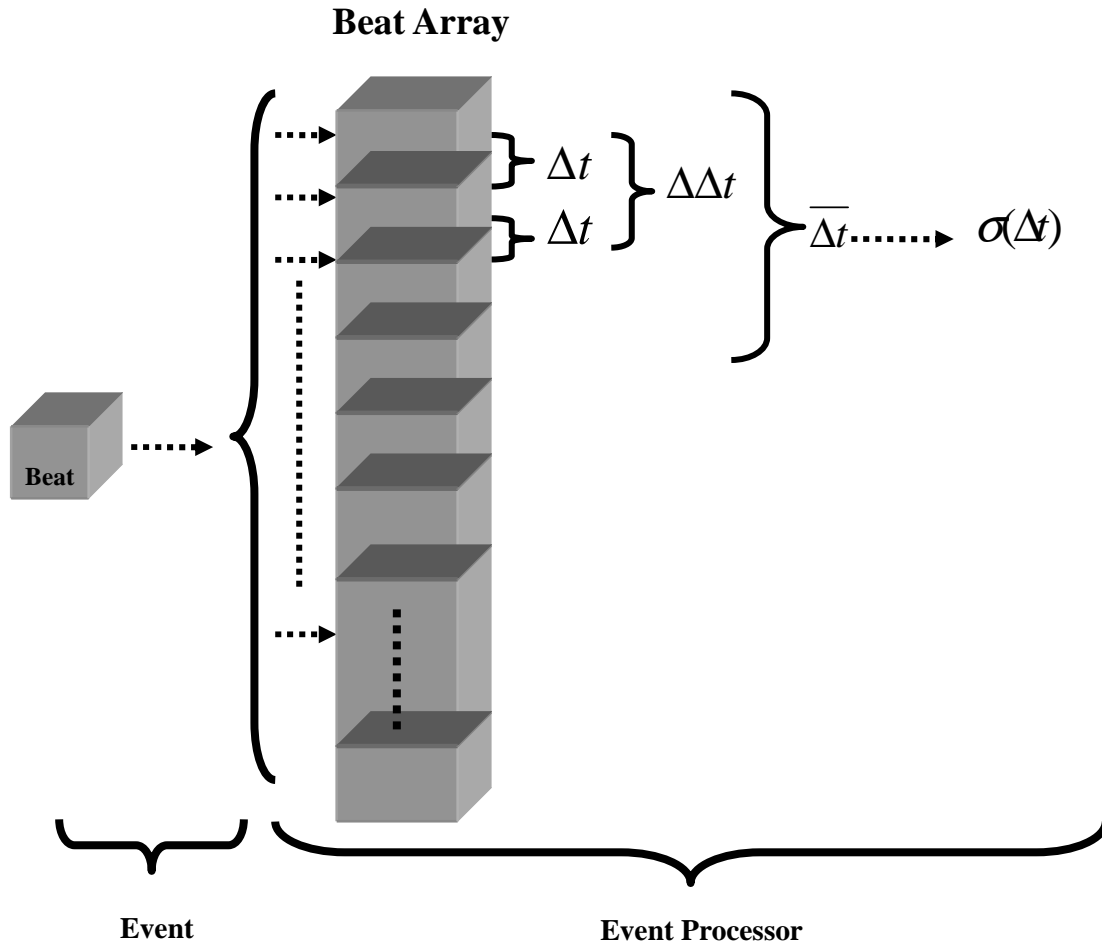


Figure 35: Event processing

For this research, the time interval between two consecutive beats is calculated, and the result will be passed on to the next stage – the Condition. Many other values can be calculated such as the mean, the standard deviation, maximum and minimum.

For the Event part, a Java object `BeatEvent` is developed to perform the beat event processing. Firstly, this `BeatEvent` extends `EventObject` that is the root class from

JDK from which all event state objects shall be derived. This `BeatEvent` object registers with a listener that has communication with the next step – (Condition) so that the processed result which is generated for the `BeatEvent` can be passed into Condition for further analysis when the corresponding flag turns true.

When the beat event is generated from the Algorithm, the `BeatEvent` object has a method that will store this beat into a Vector that implements an array of objects. Like an array, it contains components that can be accessed using an integer index. However, the size of a vector can grow or shrink as needed to accommodate adding and removing items after the vector has been created. The central feature of the generated beat event with time information is the input to this method. The returned vector will be checked by another method called `calculateBeatEvents()`, which calculates the time interval between consecutive beats following the rule of “FIFO”. Several flags ensure that there are no data consistency problems when the store and calculation method are running at the same time.

Now that the Event part is finished, its returned long array contains the entire calculated time intervals between all pairs of consecutive detected beats. Because this `BeatEvent` object registers a listener with the `ConditionManager` object, it is time for `ConditionManager` to come into play.

### 6.4.2 Condition evaluation

This `ConditionManager` also follows the rule of “FIFO” and assesses the time interval by its own evaluation rule sets.

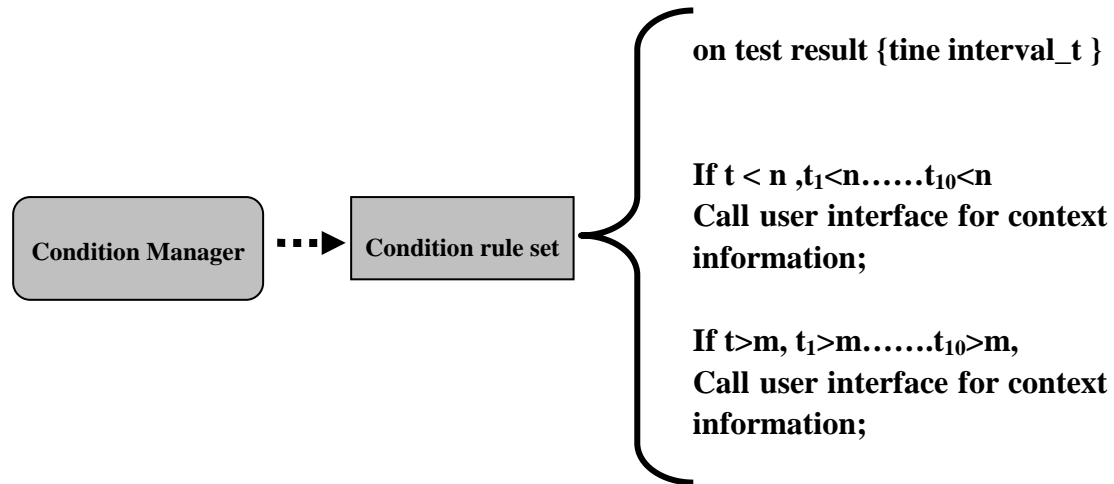


Figure 36: Condition Evaluation

This `ConditionManager` evaluates the incoming calculated time intervals and calls the user interface to collect the current activities of the patient against the current heart situations. Initially it has to implement a listener that “listens to” the Event part for the calculated result. The `ConditionManager` object initialises itself with a `BeatEvent` object and implements the method of the listener - `ProcessedEventsListener` so that the calculated time intervals will be passed to the `ConditionManager` for evaluation. For instance, four personalised rules referred to Chapter 2 Section 2.1.1 could be added for the subject as follows,

- a. *The heart rate is too high – over 153 beats/min*
- b. *The heart rate is high – between 90 and 153 beats/min*
- c. *The heart rate is normal – between 55 and 90 beats/min*

*d. The heart rate is too low – below 55 beats/min*

When the time interval comes into this rule set, this method – `evaluateCondition()` will compare the time interval with those rules individually and call the user interface to collect the context information.

### **6.4.3 User interface**

According to Chapter 4, there are two ways to design the User interface. The first one is the pop up window and the second one is the logger.

The pop up window user interface manages different user interfaces to collect the context information. When the user interface screen pops up, it will ask the user questions and those questions are pre-defined to suit the context collection. This user interface can also be implemented in other ways: a voice recorder that can record people's voice for context collection, or the user's hand writing input to describe the situation they have encountered when the interface comes up.

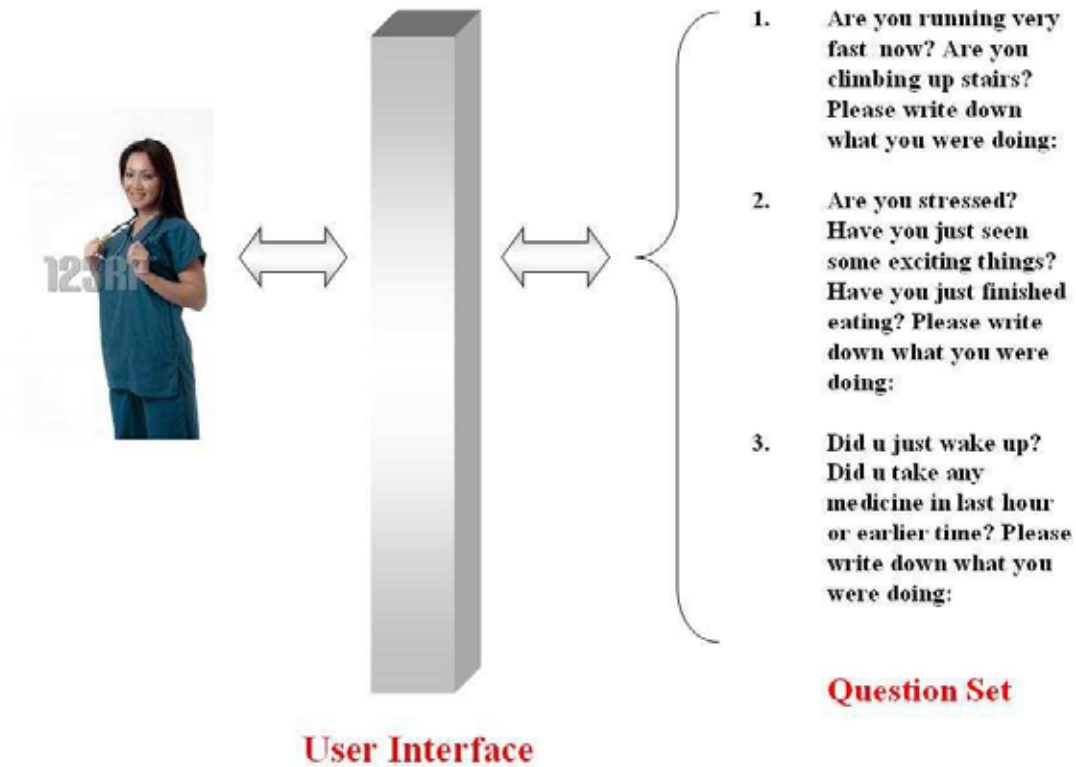


Figure 37: User interface

The user interface developed under Java's AWT [Appendix D] cannot run smoothly in the iPAQ 5550 PDA according to the test result in Chapter 7 because of the associated computational load and asynchronous transmission from the simulator. Therefore an alternative is required to accomplish this.

Instead of the AWT pop up screen, a logger file is used to record all the context information – ADLs. When a heart beat interval of unusual duration is detected, the application will ask the user via the logger file to input the current context information and also record the associated time information shown in Figure 37,

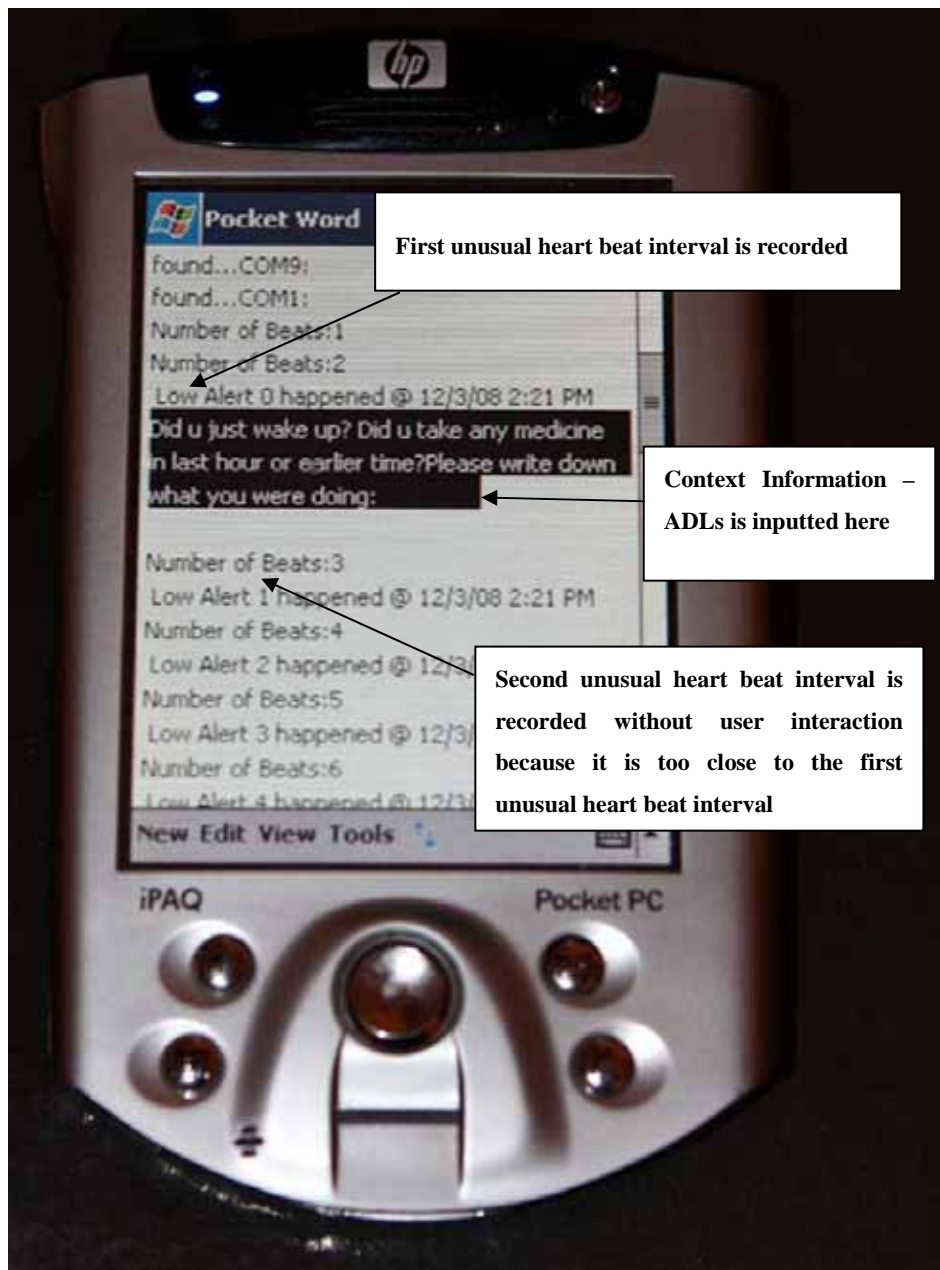


Figure 38: Information logger

As shown in the picture above, when the first low heart beat interval is detected; the information logger will ask the user to input the current activities and record the inputted information and also record the time information of this detection. When a person suffers from arrhythmia, the heart rate will not be restored to normal immediately, so the following number of beat intervals will still be irregular and it is

difficult to ask the user to perform the process again. Therefore the system will not ask the user to repeat the process within the next 100 beat intervals. This “100” is not a number from a standard clinical guideline and it is only used in the laboratory. However all the suspicious heart beat intervals will be recorded with its associated time information.

With all those implementations for the data transportations, algorithm and context collection are finished, the testing will be carried out on each of them and the overall system in Chapter 7.



## **CHAPTER 7**

### **SYSTEM TESTING**

This testing of the application is divided into the unit testing and integration testing. The unit testing includes the testing on the data transportation, algorithm and context collection. The integration testing is composed of two parts. The first part of integration testing tests only the data transportation and algorithm modules within the system; the second part tests the whole system, including the context collection module.

#### **7.1 SYSTEM SPECIFICATION**

Before considering the unit testing, the expected behaviour of the application is reviewed.

As it is intended for the system to operate as a body area network, both the sending device (ECG measurement device) and the receiver (the PDA) will be worn by the user. When the data arrives in the PDA, the algorithm should detect if a QRS complex exists in the incoming frame of the ECG data. If the algorithm detects the presence of a QRS complex, the corresponding beat event will be instantiated and will be passed into the context collection module.

The first task of the context collection module is to calculate the time interval between

the new beat event and the previous event. If a beat interval is abnormal as specified by the condition part of an E-C-A rule, i.e. too long or too short, the event processing module will issue an action for the user interface window to appear to ask the user questions to fill the context information about activities.

If the user believes that the user interface window did not ask him/her enough questions, the user can manually input the context information that will be combined with those that are automatically generated by the answered questions.

## **7.2 A BRIEF EXPLANATION OF UNIT TESTING**

In computer programming, unit testing is a procedure used to validate that individual units of source code are working properly. The system contains three modules, which are Data Transportation, Algorithm and Context Collection. Every module is tested separately first to investigate its performance.

## **7.3 UNIT TESTING OF DATA TRANSPORTATION MODULE**

### **7.3.1 Introduction**

Although the properties of Bluetooth communications have been well tested and documented [5], tests were carried out to determine how it would perform in the context of this work. For this reason, the Bluetooth data transportation channel was tested in various ways, which can be classified into three categories; distance, obstacles and baud rate.

### **7.3.2 Unit testing detail on data transportation module**

For the first experiment, the PDA was set up to receive ECG data from the simulator while the PDA was slowly moved away from the transmitting simulated measuring device in the presence of partial obstructions that obscured the line of sight. The separation distance was progressively increased from 1 metre to 30 metres. In a number of simple variations on this trial it was discovered that the PDA was able to receive the ECG data within 20 meters. In agreement with the literature, it was discovered that when the PDA is 30 metres away in line of sight from the simulator, it could no longer receive data from the Bluetooth channel. A distance of 20 to 25 metres in the presence of obstructions is certainly adequate for the purposes of this work, as it is expected that both PDA and ECG device would be worn by the subject except when the PDA is charging in its cradle in which case the subject would need too stay nearby in order to stay in range for the Bluetooth signal. The second experiment was based in a laboratory room and the setting for the experiment is shown in Figure 39.

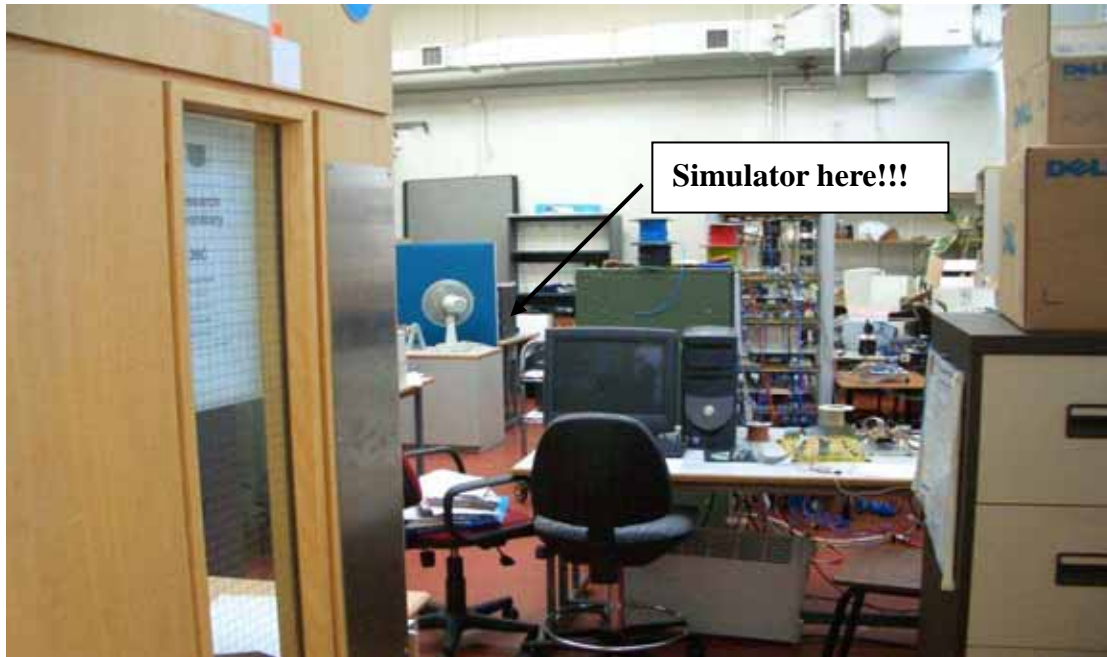


Figure 39: Laboratory room

As shown in Figure 39, the lab in which the work of this project was undertaken is shared by the electrical power quality control (UPQC) research group. This makes for an electrically and acoustically noisy environment when the UPQC machine is running and the room may also be crowded on occasion. Another Bluetooth communications test was carried out when many people with Bluetooth enabled mobile phones, portables and desktop machines, were in the lab and the UPQC machine was running. The author believed that there might possibly be more interference of the type encountered in daily life. Outside the glass window of this lab, the author ventured further from the simulator in the lab shown in Figure 40.

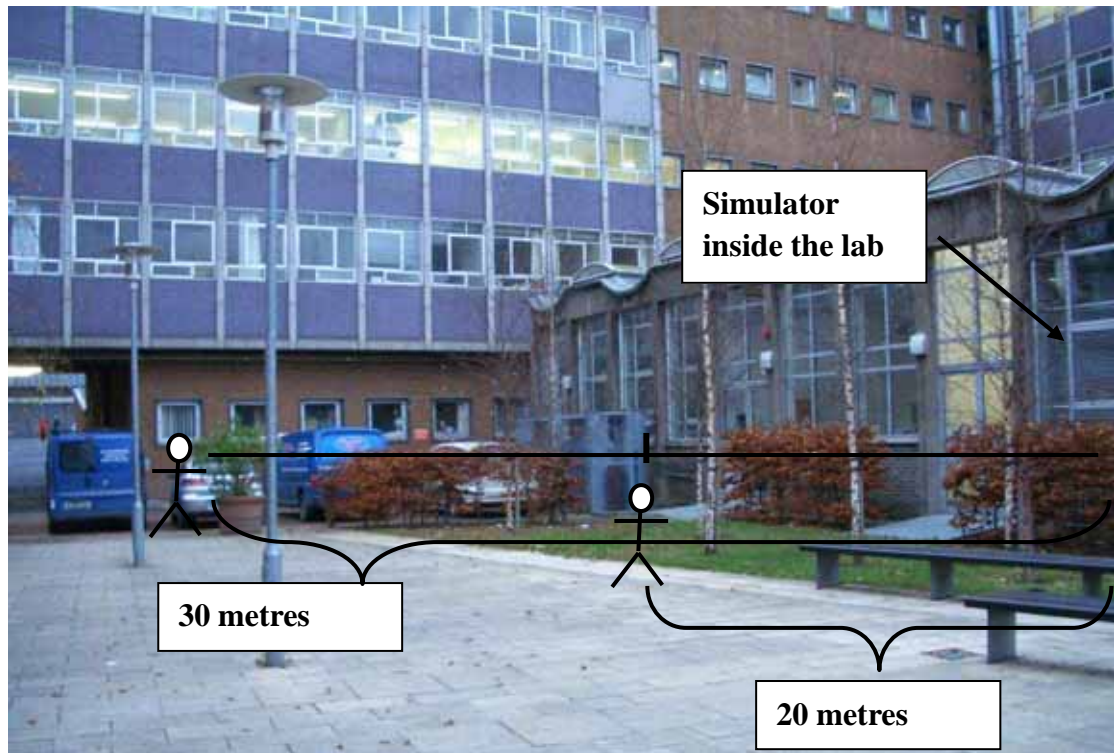


Figure 40: The campus

The author also carried the PDA along the hall adjacent to the research room (which is shown in Figure 41) with the simulator to test the Bluetooth transmission strength,

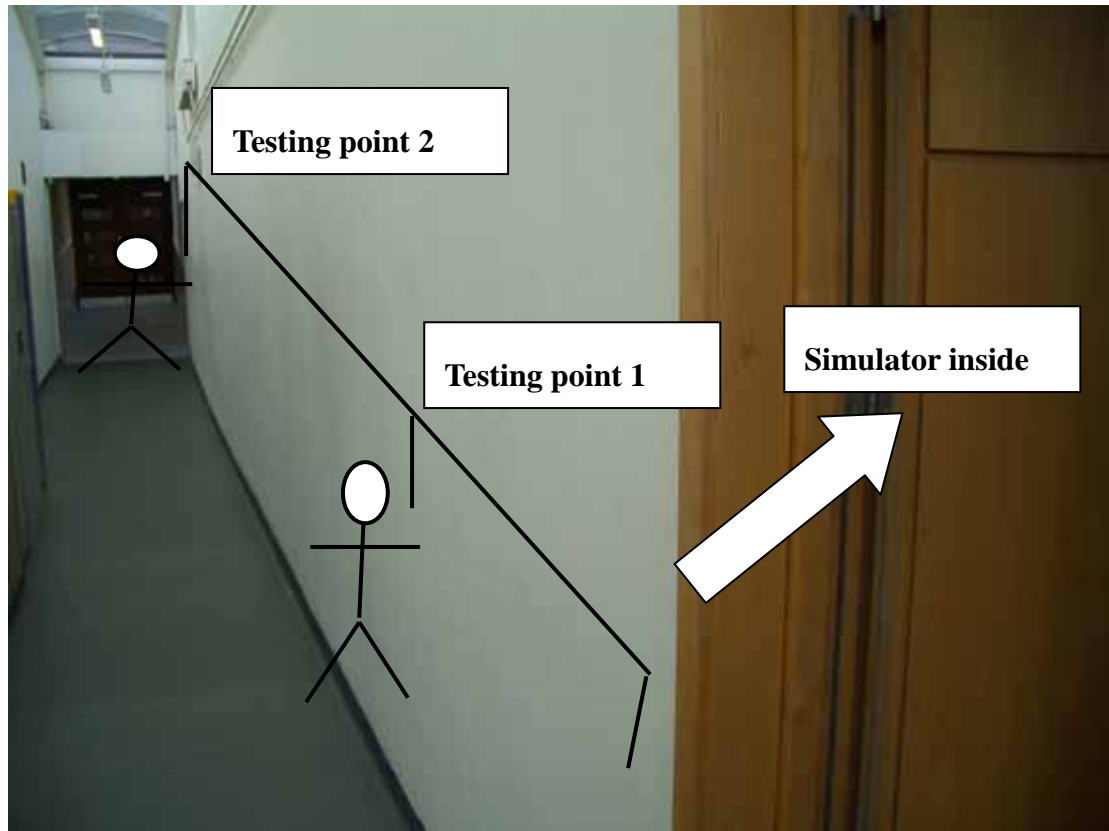


Figure 41: Along the hall inside the building

The simulator is supposed to take 500 samples a second from an ECG file with 2 bytes per sample. The minimum requirement for transmission assuming the default communication settings of 8-N-1 (8 data bits, no parity bit, 1 stop bit) would then be 8000 bps per channel. To establish whether the system behaved as expected, in a third communications experiment, different baud rates were applied in this communication, of 57600 bps, 38400 bps, 14400 bps, and 9600 bps. (Noting that 9,600 is only sufficient for one channel) A summary of the test results are shown in Table 5.

<b>Baud Rate</b>	<b>Receiving accuracy</b>
57600	100%
38400	100%
14400	100%
9600	100%

<b>Distance meter</b>	<b>Receiving accuracy</b>
1	100%
3	100%
10	100%
20	100%

<b>Surround Environment</b>	<b>Receiving accuracy</b>
Lab room	100%
Concrete wall 20cm depth	0%
Wooden door	100%
Glass window	100%

Table 5: Test result of the data communication

## 7.4 UNIT TESTING OF BEAT DETECTION ALGORITHM

The beat detection algorithm that was developed in Java, implements Kohler's zero-crossing algorithm [80] that has the preferred characteristics of low

computational load and high detection accuracy.

The simulator continuously sends frames of ECG data comprising 100 samples (or 200 bytes) down through the Bluetooth channel to the PDA, so the PDA only deals with 100 samples at a time. This permits testing of the algorithm's accuracy and memory usage. This test was somewhat hampered by some limitations of the iPAQ 5550. It was discovered upon continued testing that without any programs running in the PDA, the Bluetooth manager on the iPAQ 5550 disconnects the device from the PC every 3.5 minutes approximately. It requires the user to set up the Bluetooth connection again. This hardware problem cannot be fixed at the time of writing, which limits the functions of this system.

#### **7.4.1 Comparison between Java and MATLAB**

Before commencing the stage of testing the program in the PDA, it is necessary to compare the results of each stage in the algorithm from Java and MATLAB [110] to ensure that the more involved Java implementation is behaving similarly to the MATLAB implementation.

To compare the two versions of the algorithm, 500 samples of ECG data were tested in two different cases, and the output of each stage was compared as follows.

The first step was to filter the ECG samples through a FIR filter with a cut-off



frequency 18-35 Hz, and the outputs are listed below. The difference is due to the display of MATLAB. In MATLAB, it usually will not display many decimal places by default.

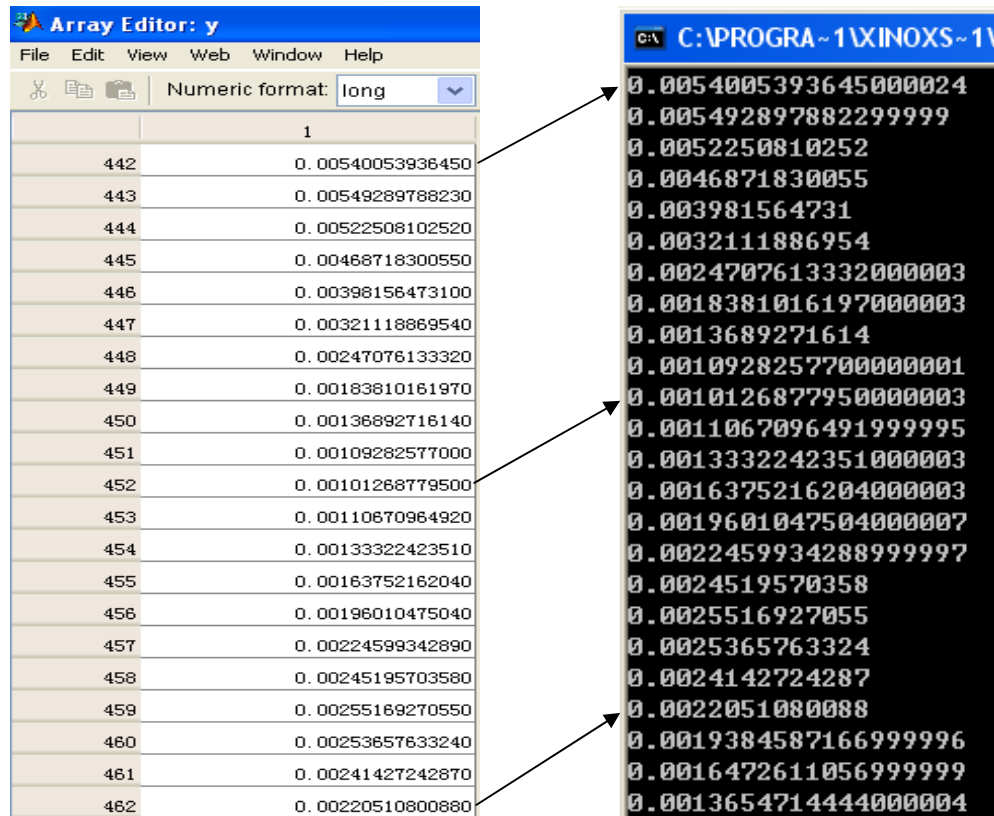


Figure 42: The output of the filtering process

The second step is the non-linear transformation of the filtered ECG data to make the QRS complex easier to detect in subsequent stages of the algorithm, and the output from both versions are almost the same.

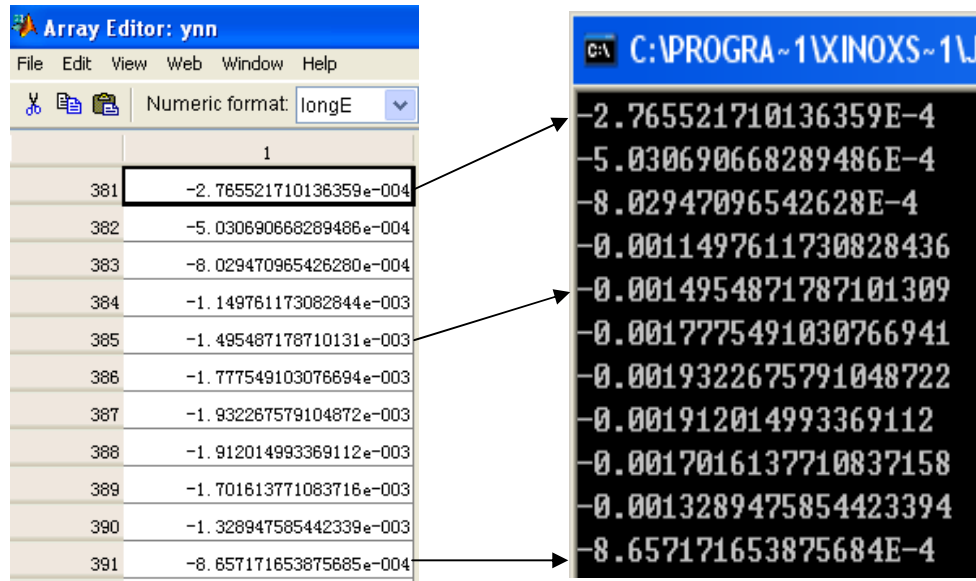


Figure 43: The output of the non-linear process

The third step is to add a high frequency sequence into the non-linear transformed ECG and get ready to detect zero-crossings.

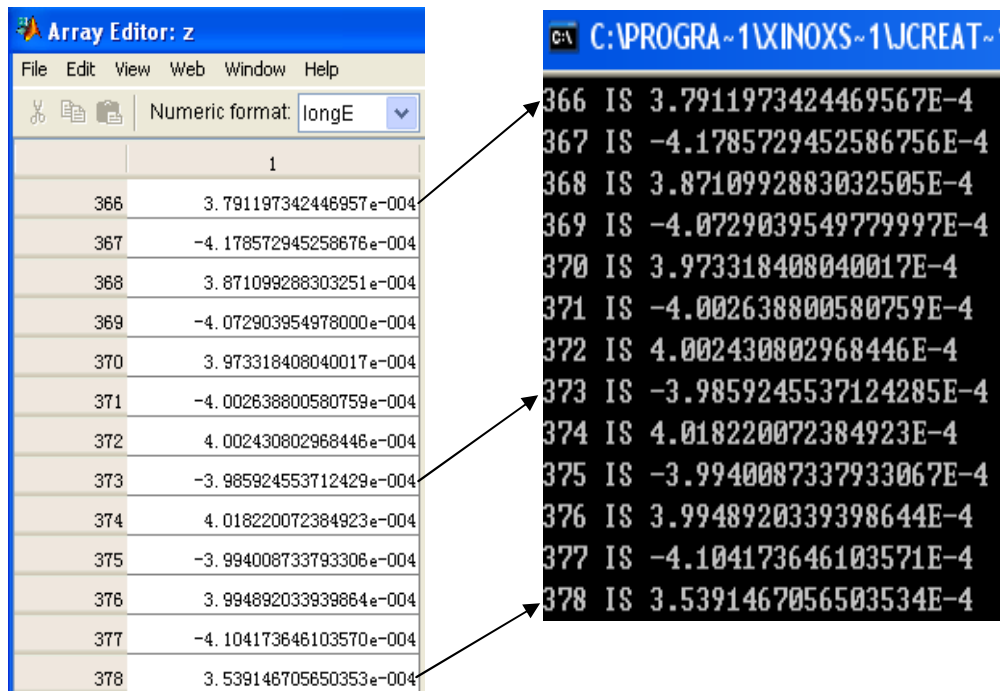


Figure 44: Comparison of the outputs of the high frequency sequence addition process

The fourth step – feature signal extraction, and the result from both of them are same.

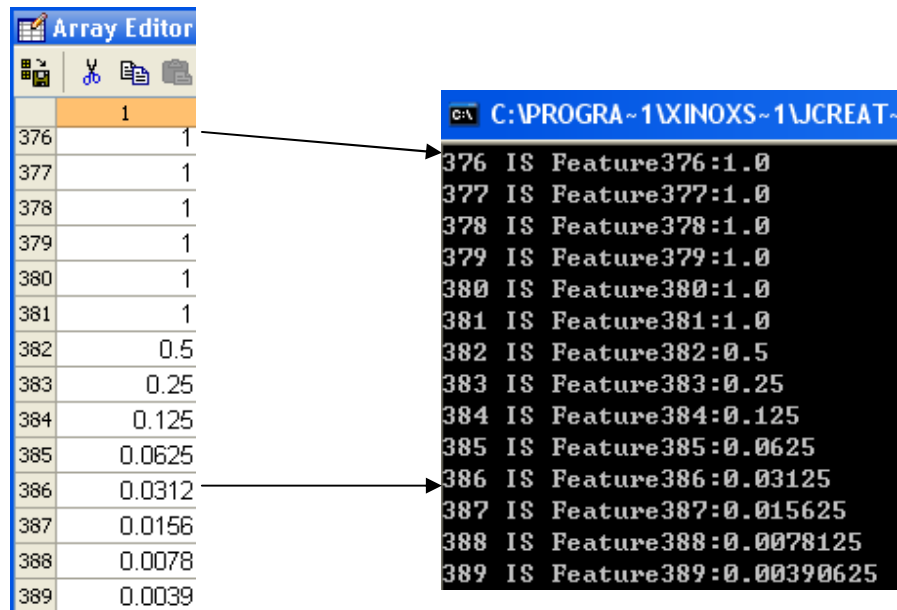


Figure 45: The output of the feature signal extraction process

Based on the feature signal, an adaptive threshold can be obtained as shown in Figure

46.

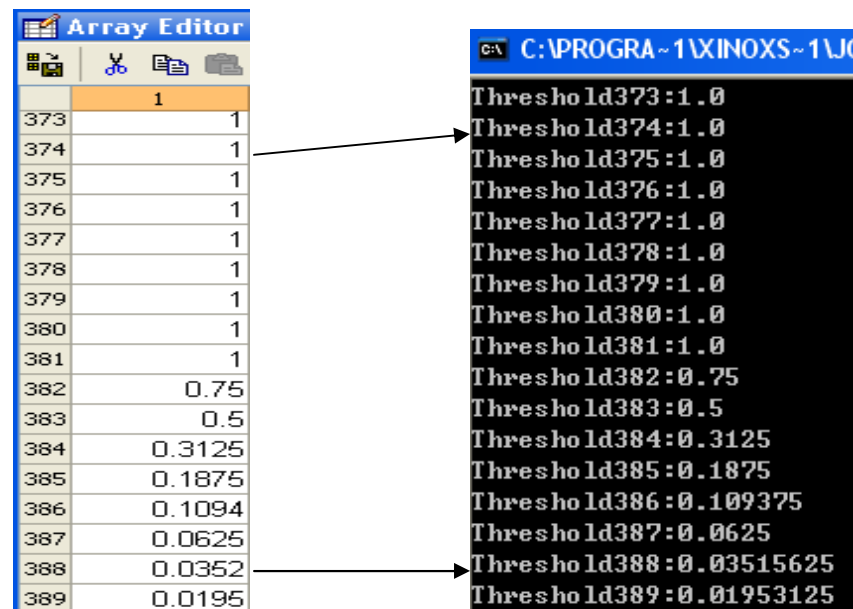


Figure 46: The output of the threshold determination process

The beat in this stream of ECG data that has 500 samples is detected in location 382

of these 500 samples,

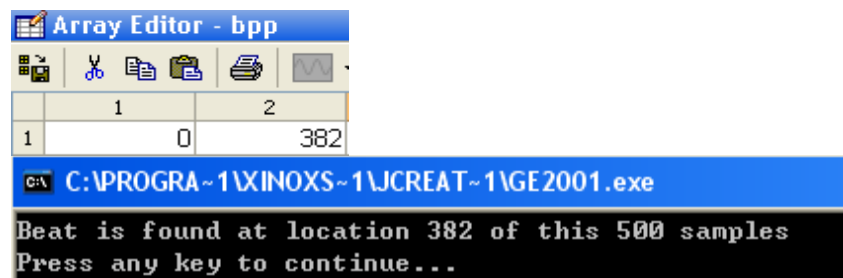


Figure 47: Beat is detected at the same location at 382

#### 7.4.2 Unit testing of beat detection algorithm in PDA

Following the successful comparison between MATLAB and Java implementations, it has been established that Java can be employed to implement this zero-crossing beat detection algorithm. The algorithm was tested for 30 minutes ECG data from MIT-BIH ECG database in the PC, and it detected 2271 beats from 2273 real heart beats.

When one arrhythmia interval (too short or too wide) is detected, the system will record this event anyway as part of the sampled ECG signal. But if this event keeps happening for the next 20 beats (10 arrhythmia intervals may take 9 seconds approximately), the system will call the user interface to ask the patient to input the current activities and the following irregular heart beat intervals will be recorded without the user interaction.

When the Jar file is tested under the NSICOM's Java environment emulator "PJSCP",

the memory usage while monitoring 100 seconds of ECG data is shown in Figure 48.

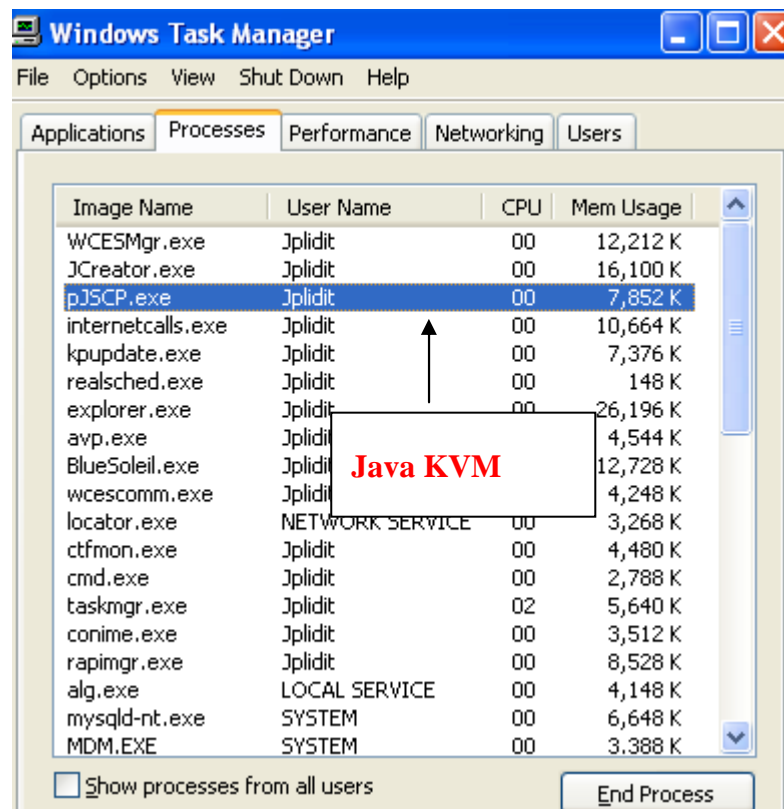
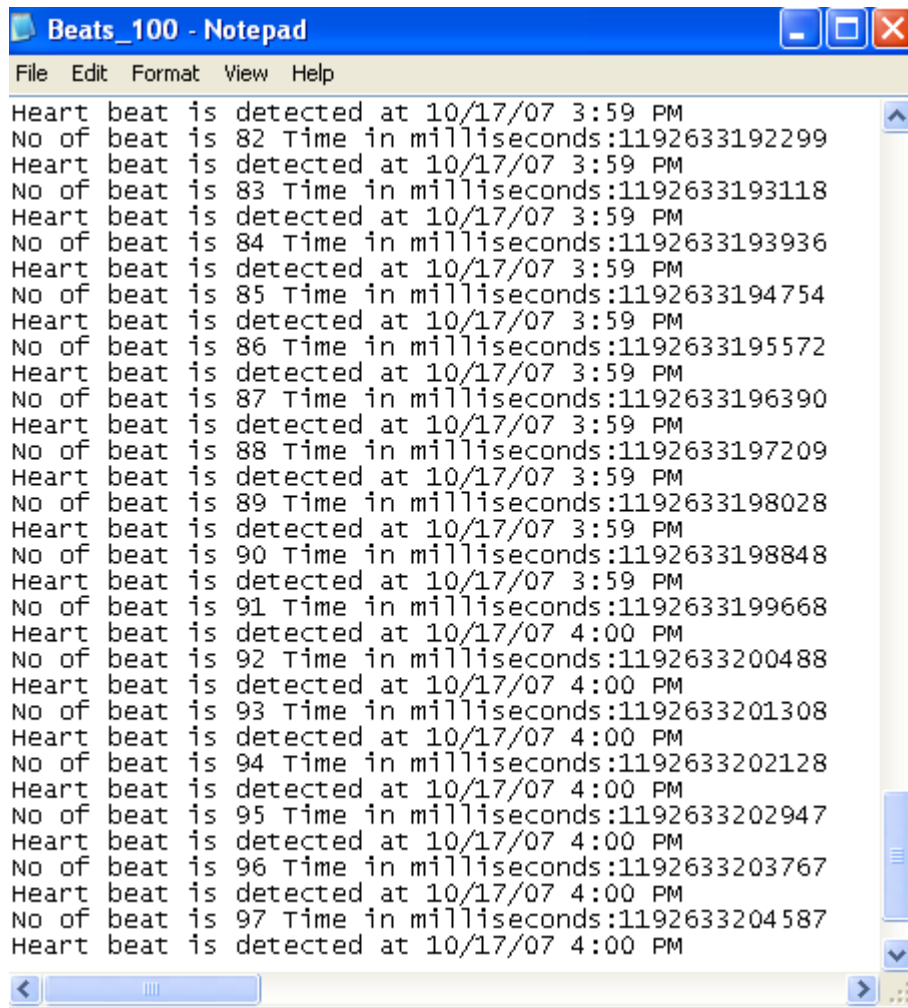


Figure 48: The memory usage while analysing 100 seconds ECG data

The Windows Task Manager is used to discover the memory usage of each application, which may not be the best solution but it gives some idea about how much memory each application will take.

The memory usage for detecting different periods of time only varies a little, so it is not necessary to consider the period of testing. The following log file shows when running 100 beats test, the beat details are required. The beat in milliseconds is very important because those numbers are used to calculate the heart beat intervals.



```
Beats_100 - Notepad
File Edit Format View Help
Heart beat is detected at 10/17/07 3:59 PM
No of beat is 82 Time in milliseconds:1192633192299
Heart beat is detected at 10/17/07 3:59 PM
No of beat is 83 Time in milliseconds:1192633193118
Heart beat is detected at 10/17/07 3:59 PM
No of beat is 84 Time in milliseconds:1192633193936
Heart beat is detected at 10/17/07 3:59 PM
No of beat is 85 Time in milliseconds:1192633194754
Heart beat is detected at 10/17/07 3:59 PM
No of beat is 86 Time in milliseconds:1192633195572
Heart beat is detected at 10/17/07 3:59 PM
No of beat is 87 Time in milliseconds:1192633196390
Heart beat is detected at 10/17/07 3:59 PM
No of beat is 88 Time in milliseconds:1192633197209
Heart beat is detected at 10/17/07 3:59 PM
No of beat is 89 Time in milliseconds:1192633198028
Heart beat is detected at 10/17/07 3:59 PM
No of beat is 90 Time in milliseconds:1192633198848
Heart beat is detected at 10/17/07 3:59 PM
No of beat is 91 Time in milliseconds:1192633199668
Heart beat is detected at 10/17/07 4:00 PM
No of beat is 92 Time in milliseconds:1192633200488
Heart beat is detected at 10/17/07 4:00 PM
No of beat is 93 Time in milliseconds:1192633201308
Heart beat is detected at 10/17/07 4:00 PM
No of beat is 94 Time in milliseconds:1192633202128
Heart beat is detected at 10/17/07 4:00 PM
No of beat is 95 Time in milliseconds:1192633202947
Heart beat is detected at 10/17/07 4:00 PM
No of beat is 96 Time in milliseconds:1192633203767
Heart beat is detected at 10/17/07 4:00 PM
No of beat is 97 Time in milliseconds:1192633204587
Heart beat is detected at 10/17/07 4:00 PM
```

Figure 49: The detected beats for 100 seconds of ECG data

## 7.5 UNIT TESTING OF EVENT BASED CONTEXT COLLECTION

Context collection is carried out by a short set of rules shown in Chapter 6. In a final system these rules would be personalised for each wearer of the system and compared against the incoming beat intervals to the thresholds set in the rule conditions and, when appropriate, the user interface is activated by the action part of the rule to request the user's assistance in recording the context information to facilitate the ECG

monitoring. The event monitoring module is tested under a combined testing environment that includes the emulator of NSICOM – PJSCP and the PDA. With the assistance of Rene [87] an engineer in NSICOM, the emulator “PJSCP” can emulate the Java environment in PDA.

### 7.5.1 Outfit of the interface window

A particular user interface in the PDA is shown in Figure 50.



Figure 50: A user interface – pop up window approach

When the user clicks the “Yes” button, the system will assume that the user is taking a heavy loaded exercise so that the heart rate has gone very high. This particular context information will be stored for future analysis.

There are 3 buttons and 2 text fields in this window. The text field on the top asks the user questions that can have multiple answers or just a single one, which depends on the user’s input. If the user keeps clicking “No”, the system will keep asking the user questions until the user clicks “Yes” and then the system will store this context

information. When the button “Exit” is clicked, the window will be closed.

### **7.5.2 Memory usage of the interface window**

The memory requirements of the user interface is an important issue. The next experiment tested this user interface to investigate how much memory a typical graphical user interface would require compared to the memory available to NSICOM’s Java virtual machine. (In the iPAQ 5550 PDA, this is 27,776 kb.) When a patient experiences arrhythmia, their heart rate may change rapidly within a few minutes. For instance, if a patient has a heart rate at 100 beats/min over 3 minutes, the system will trigger a personalised rule. If the beat interval becomes 667 milliseconds, the system will display only one user interface window upon re-occurrence of the same event instead of presenting many identical windows to get the context information.

First one arrhythmia interval is inputted into the system with Cre-Me’s emulator “pJSCP” for testing the user interface. This scenario consumed a peak of 11,104 kb of memory at the point where the user interface is displayed, which is 40% of the valid memory that NSICOM Cre-Me Java KVM can supply.



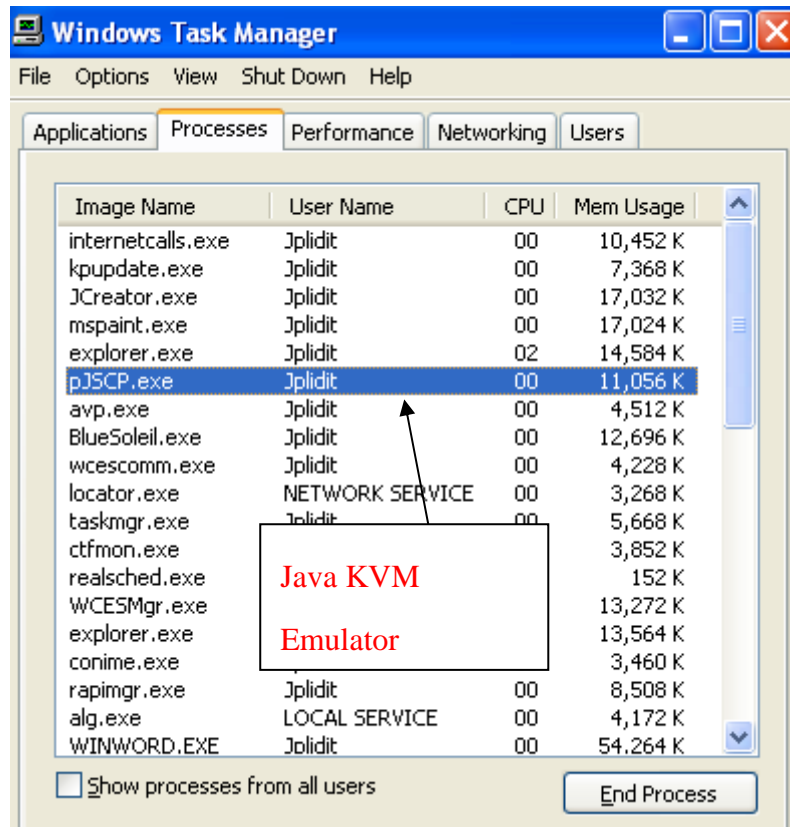


Figure 51: Interface window for 1 suspicious beat interval

A log file “jscpout” is generated that records the context information collection events.

This file is shown in Figure 52 when one suspicious beat interval is questioned.

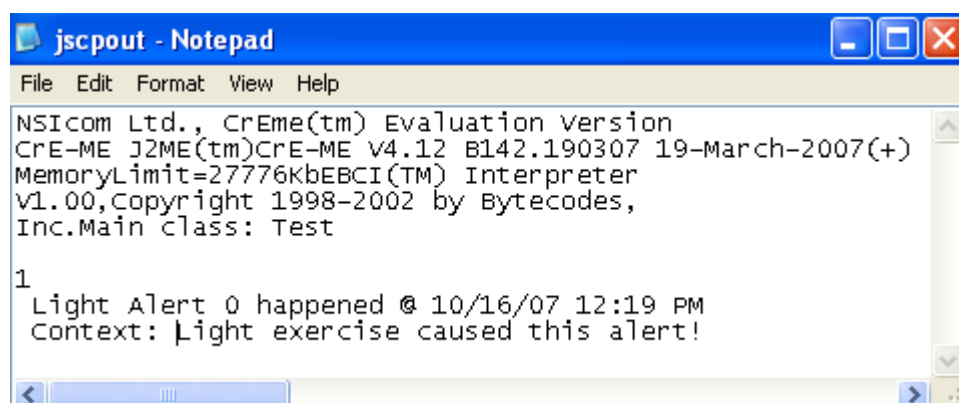


Figure 52: A log file for 1 unusual beat interval duration

This log file can be manually accessed if the user wants to add more information into it such that the system can provide more integrated context information to facilitate the doctor to make a better decision for the user, and it is shown in Figure 53.

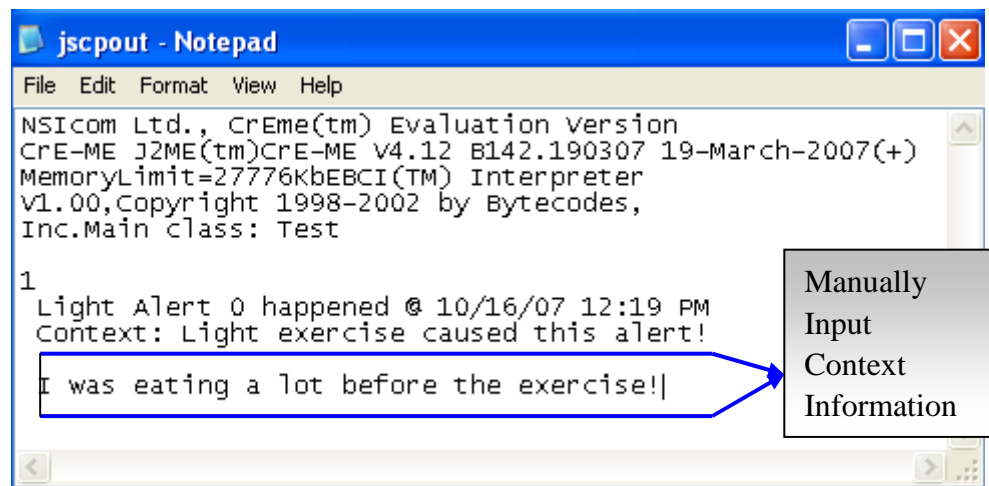


Figure 53: Manually input extra information into the log file

## 7.6 INTEGRATION TESTING

As mentioned in section 3.1, the three parts of this system are *data transportation*, *algorithm* and *context collection*. First of all, the data transportation and algorithm can work together without difficulty if the PDA can receive the ECG data. The ECG data is saved into a file called "ecgFile" that can grow continuously when streams of data arrive in the PDA, and this file's location can be either automatically generated in the root directory or self-specified. It is possible to store the ECG data file into an SD card when the PDA needs extra memory to store the incoming data.

Secondly, when the context collection was activated, problems arose in the system.

The pop up windowing system takes too much processing time so that accuracy of the algorithm's detection has gone down to 30%, which made the context collection not possible. The result is shown below in Table 6.

<b>Distance (metre)</b>	<b>Receiving Accuracy</b>	<b>Detection Accuracy Without Context</b>	<b>Detection Accuracy With Context</b>	<b>Context Awareness Accuracy</b>
<b>1</b>	100%	100%	30%	Un-stable
<b>3</b>	100%	100%	30%	Un-stable
<b>10</b>	100%	100%	30%	Un-stable
<b>20</b>	100%	100%	N/A	Un-stable

Table 6: Integration testing result - failed

Therefore, as described in previous chapters a “Log file” is introduced for the integration testing. Because of its light resource consumption, the algorithm's detection accuracy is back to normal and the whole system starts to work smoothly.

The test result is shown in Table 7.

<b>Distance (metre)</b>	<b>Receiving Accuracy</b>	<b>Detection Accuracy Without Context</b>	<b>Detection Accuracy With Context</b>	<b>Context Awareness Accuracy</b>
<b>1</b>	100%	100%	100%	stable
<b>3</b>	100%	100%	100%	stable
<b>10</b>	100%	100%	100%	stable
<b>20</b>	100%	100%	100%	stable

Table 7: Integration testing result – successful

As shown in Table 7, the system works with a logger file. Therefore it is necessary to discover how much time is taken for the beat detection algorithm and the context collection.

At the system starts, the object initialization takes typically around 400 milliseconds; subsequently, the average time for processing one sample is about 0.2 milliseconds if no beat is detected and 0.75 milliseconds if a beat is detected and the E-C-A and storage features become active. Given a two channel configuration which would be typical of Holter recording, the worst case situation would occur when a beat is simultaneously detected on two channels. In this case it would take 1.5 milliseconds to execute the entire Bluetooth reception, beat detection, E-C-A, signal recording process for a single sample. This is less than the 2 millisecond interval available to process two channels when the sampling frequency is 500 Hz. Therefore it is concluded that the algorithm can (just about) execute on this platform. As mentioned

above in this section, this assumes that the simple file-based “logger” user interface is being employed. If the full windowed user interface is used, then the PDA cannot cope with real-time operation.

## CHAPTER 8

### RESEARCH EVALUATION

The aim of this work was to develop context information collection to support mobile ECG recording. In doing so the system analysis compared the novelty of this work to other published work in the area and this is followed by the design and implementation of the system that has been described in Chapters 3 to 6. The combination of cardiac bio signal analysis, event-based context collection that is triggered by the detection of features in bio-signals, and wireless technology, comprise the core of the system that was developed during the course of the project. This chapter presents some of the author's views on the conduct and outcomes of the work. The author will also suggest some directions for further evolution from the end point of the project.

Based on the test results from Chapter 7 some of the interesting issues associated with developing a system of this type will be discussed by attempting to address the questions that were listed in Chapter 1.

#### 8.1 PDA PERFORMANCE

**Q 1:** *Can a resource limited device such as a PDA receive, analyse and react to ECG signals while simultaneously requesting context information from the user in an acceptable manner?*

The term *resource limited* in the above research question can refer to a number of different parameters such as power capacity, memory capacity, CPU speed, storage capacity and transmission bandwidth. According to the data communication testing results in Chapter 7, the Bluetooth receiver on the iPAQ 5550 can receive all ECG data as far as 20 meters away. This gives a positive chance for the ECG measuring device to work if the user leaves it on a worktop or table in a home or work environment. This saves the user from having to carry the PDA on their belt while at home or work and also allows the PDA to be charged. The Bluetooth transmitter and receiver consume substantial power if they were to operate continuously for the 24 hours that is required by a Holter monitor system. The iPAQ has a 3.7V 800mAh lithium polymer battery, which can supply 6 hours of power with consumed current that is less than 120mA [26]. This means that while the user can move within a room where the iPAQ is charging, they can only be completely mobile for 6 hours at a time. This in the author's view is not unreasonable over a 24 or 48 hour recording period. As mentioned in section 7.4 there is also the issue of maintaining the Bluetooth connection however. This would need to be solved if the system was ever to become a practical possibility.

The beat detection algorithm and Bluetooth receiver modules can run smoothly together with Bluetooth data communication in the 9,600 – 57,600 Baud range. It is worth noting however that while a baud rate of 9,600 Baud is sufficient for single channel execution, two channel transmission requires a baud rate of at least 19,200

baud. The window management system required to support the first version of the context collection part occupied over 10MB of memory and consumed considerable processor time, which crashed the whole detection system. The second version replaced a number of functions from Java.AWT package with a more efficient file-based logger and this revised version was able to smoothly record the manually inputted context information and the arrhythmia detections and worked without problems.

So in answer to the question, a PDA running a Java KVM can just about manage real-time Bluetooth reception, beat detection and a rudimentary form of context collection. It must be noted however, that the PDA used in this study was purchased in 2002 and has been surpassed by more recent models.

## **8.2 ECG DATA RECORDING**

**Q2:** *How much ECG data is it reasonable to expect a PDA to record for future examination?*

The 30 minutes of ECG data tested in this project required 11MB of storage space so that a 24-hour recording would take 528MB approximately. At time of writing in late 2008, SD card capacities range from 8MB to 32GB and can have Read/Write rate up to 10MB/Second [48]. It is expected that an extension of the SD standard will allow capacities of up to 2TB but these capacities have yet to be realised. One second of this series of ECG data is approximately 6KB which is well within the capabilities of



the SD technology. So from the above information, storage of one second of ECG data into the SD card would take the PDA about 0.6 milliseconds. In answer to the research question, modern SD cards at time of writing can store up to 32GB, more than is required for a Holter recording. In any case, it is possible to store sufficient information to support a three-lead ambulatory Holter recording over a typical 48 hour period.

### **8.3 ECG ALGORITHM AND CONTEXT COLLECTION**

**Q 3:** *Can real-time ECG analysis algorithms be implemented successfully on a memory limited platform while simultaneously requesting context information from the user?*

The IEEE [215] defines “Reliability” as the ability of a system or component to perform its required functions under stated conditions for a specified period of time.

Kohler also evaluated this method and concluded that it has a low computational load [69] and has a sensitivity of 99.70% (277 false negatives within 91006 real heart beats) and a positive “predictivity” of 99.57% (390 false positives within 91006 real heart beats) against the MIT-BIH arrhythmia database [32], which indicates that zero-crossing is very reliable for detecting heart beats in the real time scenario.

The Java implementation of the Kohler beat detection algorithm was tested for 30

minutes of the MIT-BIH ECG data, performing continuous ECG analysis, and it detected 2271 out of 2273 heart beats that satisfies the result from Kohler. It has an acceptably low memory usage; tests showed that NSICOM's KVM emulator takes 7.8 MB of the RAM during execution.

It is possible that the efficiency of the two modules – “Algorithm” and “Context Collection” could be further tuned to decrease the execution time and memory usage and so increase the reliability of the application. It is worth noting that the E-C-A module while it takes 15 milliseconds to run, only executes when a beat interval becomes available. This occurs, typically every 600 milliseconds so it doesn't contribute significantly to the average execution time of the system. The context collection module is activated even less frequently, when a series of anomalous or unusual heart beat intervals is detected. When an irregular beat interval is detected, the log will “ask” the user to input context information by simply writing a phrase to the logger file. This happens without significantly interrupting the ECG detection process.

In Chapter 7, the results of the tests demonstrated that the system would not operate satisfactorily with a full windowed user interface. However, it was demonstrated that when the second more efficient “logger”-based approach was used, the algorithm execution time and the available KVM memory of 26776 MB were sufficient to allow the full beat detection and context collection process to run smoothly on two channels

of ECG data at a sample frequency of 500 Hz. So in the author's opinion, the answer to question three is yes.

## **8.4 MINIMUM HARDWARE SPECIFICATIONS**

**Q 4:** *What are the minimum hardware specifications that will give acceptable performance?*

The iPAQ 5550 used in this work is 5 years old and has been used as the basis for two successive MPhil projects, so it cannot any longer be considered as a top-of-the-range PDA. From the discussion in section 7.6 and in section 8.3 it is apparent that this PDA corresponds to the minimum hardware specifications (**Appendix C**). Meanwhile top end phones and PDAs exceed the power of the iPAQ. It is worth noting however, that other platforms such as digital signal processors with more efficient compilers may also be sufficient for this task. So this question is really only being answered here for the case of a PDA which is programmed using Java.

## **8.5 CONCLUSION**

This work presented a mobile intelligent context collection system and an accompanying healthcare scenario that can be generalized to represent a context information collection to support other mobile self-wellness monitoring systems such as respiration and glucose levels. It can also in principle be applied to other time series signals in power signature analysis, and weather reporting.

Although the resources consumed by the first version of the context collection - “Pop up window” in this application degrades the systems performance to the extent that it ceased to function, this “Pop up window” approach is worth further investigation given that there are now more powerful PDAs and smartphones on the market.

The author’s experiences during the project indicate that the current mobile devices and wireless communications still suffer from certain limitations which may restrict the ability to store, process and transmit large volumes of multimedia clinical data in real time. However, while mobile devices still have limited memory and processing power, and are especially restricted by of battery life, it is the author’s firm view that the future is bright for mobile healthcare.

With the rapid advances in computational power and communications infrastructure PDA-like devices with low memory usage and integrated context information collection which are managed by the patient have the potential to be an important and powerful tool for the doctor to make a better decision for the user’s health condition afterwards or on remote monitoring.

## **8.6 FUTURE WORK**

While carrying out the work of the project, the author identified further work, which would enhance the system described here. The monitoring and context collection

system is composed of three parts: which are the data communication, algorithm and context collection (based on E-C-A). Each part has associated issues that could be resolved.

### **8.6.1 DATA COMMUNICATION ENHANCEMENTS**

The battery capacity of the PDA limits the duration of recording period so that a power-saving approach is one innovation that would certainly improve the user experience. As described in Chapter 1, Zigbee consumes less power than Bluetooth when transmitting wireless signals. But as far as the author is aware, it is not yet supported by PDAs. Nevertheless it will be a “must try” solution for the signal transportation on the ECG instrument and PDA when Zigbee becomes available on PDAs.

The signal transportation time could be reduced by a burst transmission. The baud rate could be set to 38,400 Baud to send a stored ECG signal such that the signal transportation time is reduced, thus possibly saving battery power at least on the transmitter side. The difficulty with this approach is that there would then be a need for a sizeable buffer on the ECG measuring side of the wireless link.

If the signal transmission element of the system was reduced or eliminated, the battery would last longer. Taking this approach to the extreme while still allowing satisfactory user interface capabilities requires that the PDA has an embedded biosignal acquisition module. However this approach represents a move back towards more

wires on the patient, which is against the flow of the work presented here.

Nevertheless it remains a valid topic for further investigation.

### **8.6.2 ALGORITHM**

Considering the literature survey and MATLAB implementation of the compared real time ECG detection algorithms, selection of these algorithms could be implemented into a library of algorithms. Furthermore, this set of algorithms could be extended with additional real-time algorithms, which detected the presence of other features of ECG i.e. the “shape” of the ECG. The author envisages that this real time ECG detection library would allow the user to choose an algorithm and configure it to suit the detection requirement or the patient. Therefore the detection becomes more integrated.

### **8.6.3 CONTEXT COLLECTION (E-C-A)**

This context collection system utilises a hard-coded E-C-A module. Following previous work at DIT in the TUDOR project, this could be replaced with a configurable rule engine and a matching E-C-A rule editor to suit the patient needs. The rule engine manages many categorized rules. For instance, the administrator may choose an E-C-A rule set that is designed for a particular clinical scenario such as cardiac rehabilitation therapy, or post operative monitoring. A rule set could also be personalised to suit a particular patient.

The obvious next step would be the creation of an interactive mobile guideline system.

The patients would not only collect the context information when unusual features of the ECG are detected but also following clinical best practice guidelines automated health alerts could be sent to the local GP and appropriate clinical suggestion could be available for the patient.

## REFERENCE

1. American Heart Association, “*ECG Basics*”, 2006
2. J. R. Hampton, “*The ECG MADE EASY*”, Fifth Edition, Churchill Livingstone, An imprint of Elsevier Science Limited, 1997
3. D. Dubin, “*Rapid Interpretation of EKG’s*”, 6<sup>th</sup> Edition, Library of Congress Catalog Card Number 88-072108, Printed in Hong Kong, 2000, ISBN 0-912912-06-5
4. IEEE Standard 802.15.1 “*Wireless MAC and PHY Specifications for Wireless Personal Area Networks*”, [www.ieee802.org/15/pub/TG1.html](http://www.ieee802.org/15/pub/TG1.html) [Last accessed on 01/01/2009]
5. Bluetooth, [www.bluetooth.com/bluetooth](http://www.bluetooth.com/bluetooth) [Last accessed on 01/01/2009]
6. A. Cronin, “*The Investigation of a Wireless Medical Homecare Data Logger*”, Dublin Institute of Technology, Department of Control Systems and Electrical Engineering, 2005.
7. WordNet, [wordnet.princeton.edu/](http://wordnet.princeton.edu/) [Last accessed on 01/01/2009]
8. Veteran’s Review Board Australia, “*Activities of Daily Living*”, available on the website: [www.vrb.gov.au/pubs/garp-chapter16.pdf](http://www.vrb.gov.au/pubs/garp-chapter16.pdf) [Last accessed on 01/01/2009]
9. V.M Jones, and H Mei, and Broens, T.H.F. and Widya, I.A. and Peuscher, J, “*Context Aware Body Area Networks for Telemedicine*”, 8th Pacific Rim Conference on Multimedia, Hong Kong, China, Springer Verlag, December 11-14, 2007, pp590-599, ISBN 978-3-540-77254-5
10. M. Wegdam, “*AWARENESS: A project on Context AWARE mobile NETworks and*



- ServiceS*”, Lucent Technologies, Bell Labs Europe, 2004-2008
11. H. J. Kim, “A *Context-Aware Traveler Healthcare Service (THS) System*”, Pervasive Health Conference and Workshops, November 29-December 1, 2006, pp1-3
  12. J. H. Hong, N. J. Kim, E. J. Cha, T. S. Lee, “*Classification Technique of Human Motion Context based on Wireless Sensor Network*”, Proceedings of the 2005 IEEE, Engineering in Medicine and Biology 27th Annual Conference, Shanghai, China, September 1-4, 2005, pp5201-5202
  13. J. Healey and B. Logan, “*Wearable Wellness Monitoring Using ECG and Accelerometer Data*”, Hewlett-Packard Cambridge Research Laboratory, Wearable Computers, 2005. Proceedings. Ninth IEEE International Symposium, October 18-21, 2005, pp220-221, ISBN: 0-7695-2419-2
  14. Y.S.Y. Comez, M.T.S.R Gomes, “*Smart Sensors and MEMS*”, Springer, January 2005 ISBN-10: 1402029276
  15. S. E. Lyshevski, “*MEMS and NEMS: Systems, Devices and Structures*”, CRC Press Inc, January 18, 2002, ISBN-13: 978-0849312625
  16. Jackson Health System, [www.jhsmiami.org](http://www.jhsmiami.org) [Last accessed on 01/01/2009]
  17. High Frequency Electronics, “*What is a “Personal Area Network” (PAN)?*”, Summit Technical Media, LLC, March, 2004
  18. G. Chen and D. Kotz, “*A Survey of Context-Aware Mobile Computing Research*”, Computer Science Technical Report TR2000-381, Department of Computer Science, Dartmouth College, Dartmouth

19. American Heart Associations, *“Management of Patients With Ventricular Arrhythmias and the Prevention of Sudden Cardiac Death”*, ACC/AHA/ESC Pocket Guideline Based on the ACC/AHA/ESC 2006 Guidelines, September, 2006
20. *“Holter Devices Comparison”*, available on the website: [www.medcompare.com/matrix/132/Holter-Monitor.html](http://www.medcompare.com/matrix/132/Holter-Monitor.html) [Last accessed on 01/01/2009]
21. Ambulatory ECG, [www.americanheart.org/presenter.jhtml?identifier=4425](http://www.americanheart.org/presenter.jhtml?identifier=4425) [Last accessed on 01/01/2009]
22. P. R. Kowey and D. Z. Kocovic *“Ambulatory Electrocardiographic Recording”*, American Heart Association, ISSN: 0009-7322, Online 72514
23. K. HL, P. J. Podrid. *“Role of Holter monitoring and exercise testing for arrhythmia assessment and management”*, Cardiac Arrhythmia Philadelphia, Pa: Lippincott Williams & Wilkins, 2001, pp165-193
24. AH Kadish, AE Buxton, HL Kennedy, *“ACC/AHA clinical competence statement on electrocardiography and ambulatory electrocardiography: a report of the ACC/AHA/ACP-ASIM task force on clinical competence”*, Circulation. 2001, 104: pp3169-3178
25. M. H. Crawford, SJ Bernstein, PC Deedwania, JP DiMarco, KJ Ferrick, A Jr Garson, *“ACC/AHA guidelines for ambulatory electrocardiography”*, A report of the American College of Cardiology/American Heart Association Task Force on Practice Guidelines (Committee to Revise the Guidelines for Ambulatory

- Electrocardiography), Journal of the American College of Cardiology, 1999, Volume 34, Issue 4, October, pp1262-1347
26. PJ Zimetbaum, KY Kim, ME Josephson, AL Goldberger, DJ Cohen, “*Diagnostic yield and optimal duration of continuous-loop event monitoring for the diagnosis of palpitations, cost-effectiveness analysis*”, Ann Intern Med 1998, 128(11): pp890-895
  27. R. Fensli, E. Gunnarson, O. Hejlesen, “*A Wireless ECG System for Continuous Event Recording and Communication to a Clinical Alarm Station*”, Engineering in Medicine and Biology Society, IEMBS; 26th Annual International Conference of the IEEE, September 1-5, 2004, Volume 1, Issue , pp2208-2211
  28. U. K. H Wiegand, H. bonnemeier, F. Bode, F. Eberhardt, J. K. R Chun, H. A Katus, W. Peters, “*Continuous Holter Telemetry of Atrial Electrograms and Marker Annotations Using a Common Holter Recording System: Impact on Holter Electrocardiogram Interpretation in Patients with Dual Chamber Pacemakers*”, Pacing and Clinical Electrophysiology, December, 2002, Volume 25, Issue 12, p1724
  29. N. Buscemi “*Melatonin for Treatment of Sleep Disorders*”, Prepared for: Agency for Healthcare Research and Quality U.S. Department of Health and Human Services     [www.ahrq.gov/downloads/pub/evidence/pdf/melatonin/melatonin.pdf](http://www.ahrq.gov/downloads/pub/evidence/pdf/melatonin/melatonin.pdf)  
[Last accessed on 01/01/2009]
  30. M. Sung, C. Marci, and A. Pentland, “*Wearable Feedback Systems for Rehabilitation*”, The media laboratory, MIT, Cambridge, MA; Massachusetts

- General Hospital, Department of Psychiatry, Boston, MA, Journal of NeuroEngineering and Rehabilitation, published June 29, 2005
31. R. Sutton, M. Brignole, "*The Implantable Loop Recorder*", Royal Brompton Hospital, London; Arrhythmologic Centre, Osped-ali Tigullio, Lavagna
  32. J. Rodríguez, A. Goni and A. Illarramendi, "*Real-Time Classification of ECGs on a PDA*", March, 2005, IEEE Transactions on Volume 9, Issue 1, pp23-34
  33. PAN, [www.ieee802.org/15/](http://www.ieee802.org/15/) [Last accessed on 01/01/2009]
  34. Y. Farmine, M. D. Silva, "*W-CDMA Uplink Soft Handover Gain Measurements*", Central Radio Design, Dublin, Ireland and Estel Consulting Ltd, Slough, England
  35. T. Broens, D. Quartel, and M. V. Sinderen, "*Towards a Context Binding Transparency*", presented at 13th EUNICE Open European Summer School, Enschede, the Netherlands, 2007, Lecture Notes in Computer Science(LNCS), Springer Berlin / Heidelberg, ISBN 978-3-540-73529-8
  36. Zigbee, [www.zigbee.org/en/index.asp](http://www.zigbee.org/en/index.asp) [Last accessed on 01/01/2009]
  37. Wi – Fi, [www.wi-fi.org](http://www.wi-fi.org) [Last accessed on 01/01/2009]
  38. S. Co Ergen, "*ZigBee/IEEE 802.15.4 Summary*", September 10, 2004
  39. F. L. Lewis, "*Wireless Sensor Networks*", Associate Director for Research Head, Advanced Controls, Sensors, and MEMS Group Automation and Robotics Research Institute The University of Texas at Arlington, and this research was supported by ARO Research Grant DAAD 19-02-1-0366
  40. M. Kohvakka, "*Performance Analysis of IEEE 802.15.4 and ZigBee for Large-Scale Wireless Sensor Network Applications*", Proceedings of the 3rd

- ACM international workshop on Performance evaluation of wireless ad hoc, sensor and ubiquitous networks, Terromolinos, Spain, SESSION: Energy Efficient algorithms, 2006, pp48-57, ISBN:1-59593-487-1
41. M. Berggren, “*Wireless communication in telemedicine using Bluetooth and IEEE 802.11b*”, Technical report 2001-028, Department of Information Technology, Uppsala University, Box 337, SE-751 05 Uppsala, Sweden, ISSN 1404-3203
  42. E Callaway, P Gorday, L Hester, J.A Gutierrez, M Naeve, B Heile, and V Bahl, “*Home Networking with IEEE 802.15.4: A developing standard for low-rate wireless personal area networks*”, Communications Magazine, IEEE , August, 2002, Volume: 40 , Issue: 8, pp70-77
  43. IrDA, [standards.ieee.org/wireless/](http://standards.ieee.org/wireless/) [Last accessed on 01/01/2009]
  44. American Heart Association, “*Guidelines for Ambulatory Electrocardiography, Part II*”, September, 1999, Volume 34, No. 3, pp912-48, [www.americanheart.org](http://www.americanheart.org) [Last accessed on 01/01/2009]
  45. American Heart Association, [www.americanheart.org/](http://www.americanheart.org/) [Last accessed on 01/01/2009]
  46. L. Angelini, G. Lattanzi, R. Maestri, D. Marinazzo, G. Nardulli, L. Nitti, M. Pellicoro, G. D. Pinna, and S. Stramaglia, “*Phase shifts of synchronized oscillators and the systolic-diastolic blood pressure relation*”, Center of Innovative Technologies for Signal Detection and Processing, Università di Bari, Bari, Italy, Phys. Rev. E 69, 061923 (2004), Issue 6, June, 2004

47. J. Enderle, S. Blanchard, J. Bronzino, *“Introduction to Biomedical Engineering”*, Second Edition, Elsevier Academic Press, 2005
48. PJ Zimetbaum, ME. Josephson *“The evolving role of ambulatory arrhythmia monitoring in general clinical practice”*, Ann Intern Med, 1999, pp848-856
49. John G. Webster, *“Medical Instrumentation: Application and Design Third Edition”*, Published by Wiley, 1998, ISBN 0471153680
50. Medtronic Reveal® Insertable Loop Recorder. *“The World's First Implantable Diagnostic Device”*, available on the website: [www.medtronic.com](http://www.medtronic.com) [Last accessed on 01/01/2009]
51. QRS Diagnostic, *“PDAs-Mobile Diagnostic Workstations. QRS Diagnostic”*, Sep 82004, available on the website: [www.qrsdiagnostic.com](http://www.qrsdiagnostic.com) [Last accessed on 01/01/2009]
52. V. Thulasi Bai, and Srivatsa S. K, *“Design and Simulation of Portable Telemedicine System for High Risk Cardiac Patients”*, International Journal of Biomedical Sciences, 2006, Volume 1, No. 1, pp1306-1216
53. Cardio Control, [www.cardiocontrol.com/cardio.htm](http://www.cardiocontrol.com/cardio.htm) [Last accessed on 01/01/2009]
54. Pulse Medical Limited, *“ECG System, MeditSense 100 H”*, Sep 8, 2004, available on the website: [www.pulsemmedical.co.uk](http://www.pulsemmedical.co.uk) [Last accessed on 01/01/2009]
55. D. Konstantas, V. Jones, R. Bults, *“MobiHealth –innovative 2.5 / 3G mobile services and applications for healthcare”*, University of Twente Center for Telematics and Information Technology – APS 7500 AE Enschede

56. 3G, [en.wikipedia.org/wiki/3G](http://en.wikipedia.org/wiki/3G) [Last accessed on 01/01/2009]
57. “*Global System for Mobile Communication*”, The International Engineering Consortium, [www.iec.org](http://www.iec.org) [Last accessed on 01/01/2009]
58. I. Sachpazidis, “*@Home: a modular telemedicine system*” in Proc. 2<sup>nd</sup> workshop on Mobile Computing in Medicine, Heidelberg, Germany, April, 2002
59. C. Kunze, U. Grossmann, W. Stork and K.D. Müller-Glaser, “*Application of ubiquitous computing in personal health monitoring systems*” in Proc. 36th Ann. meeting of the German Society for Biomed. Eng, 2002
60. [www.es.lth.se/home\\_tde/index\\_sven.html](http://www.es.lth.se/home_tde/index_sven.html) [Last accessed on 01/01/2009]
61. [www.winlab.rutgers.edu/focus/Focus1999/Haartsen.html](http://www.winlab.rutgers.edu/focus/Focus1999/Haartsen.html) [Last accessed on 01/01/2009]
62. S. Rahn, “*Let’s Talk About Bluetooth!*”, Kennesaw State University, Educational Technology Training Center
63. C. Andren, “*A Comparison of Frequency Hopping and Direct Sequence Spread Spectrum Modulation for IEEE 802.11 Applications at 2.4 GHz*”, Harris Semiconductor Palm Bay, Florida
64. Java Community Process, [jcp.org/en/home/index](http://jcp.org/en/home/index) [Last accessed on 01/01/2009]
65. Rococo Java Bluetooth, [www.rococosoft.com/](http://www.rococosoft.com/) [Last accessed on 01/01/2009]
66. ETSI, [www.etsi.org/WebSite/homepage.aspx](http://www.etsi.org/WebSite/homepage.aspx) [Last accessed on 01/01/2009]
67. Oracle, [www.oracle.com](http://www.oracle.com) [Last accessed on 01/01/2009]
68. B. Wu and K. Dube, “*Applying Event-Condition-Action Mechanism in Healthcare: a Computerised Clinical Test-Ordering Protocol System (TOPS)*”,

- Cooperative Database Systems for Advanced Applications, CODAS, IEEE  
INSPEC Accession Number: 7068993, 2001, ISBN: 0-7695-1128-7
69. J. Bailey, “*An Event-Condition-Action Language for XML*”, International World Wide Web Conference, Proceedings of the 11th international conference on World Wide Web, Honolulu, Hawaii, USA, 2002, pp486-495, ISBN:1-58113-449-5
70. S. Gatzia, Unbundling Active Functionality, ACM SIGMOD Record, 1998, Volume 27, Issue 1, pp35-40, ISSN:0163-5808
71. M. Zoumboulakis, “*Active Rules for Sensor Databases*”, ACM International Conference Proceeding Series, Volume 72, Proceedings of the 1st international workshop on Data management for sensor networks: in conjunction with VLDB, Toronto, Canada, Session: Programming languages and architectures, 2004, pp 98-103.
72. KVM, [java.sun.com/products/cldc/wp/index.html#spec](http://java.sun.com/products/cldc/wp/index.html#spec) [Last accessed on 01/01/2009]
73. K. Raymond, “*Reference Model of Open Distributed Processing (RM-ODP): Introduction, CRC for Distributed Systems Technology*”, Centre for Information Technology Research, University of Queensland, Brisbane 4072 Australia
74. R. O. Sinnott and K. J. Turner, “*Applying formal methods to standard development: The Open Distributed Processing experience*”, Computer Standards and Interfaces, October, 1995, volume 17: pp615-630
75. UML, [www.uml.org/](http://www.uml.org/) [Last accessed on 01/01/2009]



76. OMG, [www.omg.org/](http://www.omg.org/) [Last accessed on 01/01/2009]
77. D. P. Wood, K. C. Kang, “*A Classification and Bibliography of Software Prototyping*”, Software Engineering Institute. This work is sponsored by the U.S. Department of Defense. Copyright © by Carnegie Mellon University, 1992, CMU/SEI-92-TR-013
78. E. Callaway, “*Low Power Consumption Features of the IEEE 802.15.4/ZigBee LR-WPAN Standard*”, Florida Communication Research Lab, Motorola Labs
79. V, U. and V, R. “*Emerging wireless and mobile networks*”, Communications of the Association for Computing Machinery (ACM), June, 2000, Volume 43, No. 6, pp73-81
80. K. B. U, H, C, and R Orglmeister (2003) “*QRS Detection Using Zero Crossing Counts*”, Progress in Biomedical Research, 8 (3), pp138-145
81. Sun, [java.sun.com/javame/index.jsp](http://java.sun.com/javame/index.jsp) [Last accessed on 01/01/2009]
82. “*Java development on PDAs: Building applications for PocketPc and Palm devices*”, 1<sup>st</sup> printing, Addison Wesley, May, 2003, ISBN 0-201-71954-1.
83. Jeode, [www.linuxdevices.com/products/PD5339958658.html](http://www.linuxdevices.com/products/PD5339958658.html) [Last accessed on 01/01/2009]
84. Esmertec, [www.esmertec.com/](http://www.esmertec.com/) [Last accessed on 01/01/2009]
85. Avelink, [www.avelink.com/](http://www.avelink.com/) [Last accessed on 01/01/2009]
86. Avetana, [www.avetana-gmbh.de/avetana-gmbh/produkte/jsr82.eng.xml](http://www.avetana-gmbh.de/avetana-gmbh/produkte/jsr82.eng.xml) [Last accessed on 01/01/2009]
87. Personal communication of Rene, Engineer in NSICOM at [www.nsicom.com/](http://www.nsicom.com/)

[Last accessed on 01/01/2009]

88. [www.bluesoleil.com/](http://www.bluesoleil.com/) [Last accessed on 01/01/2009]
89. [www.rxtx.org/](http://www.rxtx.org/) [Last accessed on 01/01/2009]
90. [www.mathworks.com/](http://www.mathworks.com/) [Last accessed on 01/01/2009]
91. M. Grand, “*Patterns in Java*”, Wiley Computer Publishing, John Wiley & Sons, 1998, Volume 1, , ISBN 0-471-25839-3
92. H. Zhou, “*A Real-Time Continuous Cardiac Arrhythmias, Detection System: RECAD*”, School of Electronic Information, WUHAN University, China, Engineering in Medicine and Biology Society, IEEE-EMBS 2005, 27th Annual International Conference of the IEEE, 2005, pp875-881
93. S Led, J. Fernández and L. Serrano, “*Design of a Wearable Device for ECG Continuous Monitoring Using Wireless Technology*”, Department of Electrical and Electronic Engineering, Universidad Pública de Navarra, Engineering in Medicine and Biology Society, IEMBS apos, 26th Annual International Conference of the IEEE, September 1-5, 2004, Volume 2, pp3318-3321
94. J. P. Pan and W. J. Tompkins, “*A Real Time QRS Detection Algorithm*”, IEEE Transactions on Biomedical Engineering, March, 1985, Volume BME-32, No.3
95. P.S. Hamilton and W.J. Tompkins, “*Quantitative investigation of QRS detection rules using the MIT/BIH arrhythmic database*”, IEEE Transactions on Biomedical Engineering, 1986, Volume 33, pp1157-1165
96. V.X. Afonso, W.J Tompkins, T.Q Nguyen and S Lou, “*ECG Beat Detection Using Filter Banks*”, IEEE Transactions on Biomedical Engineering, 1999, pp

97. E. Jovanov, A. Milenkovic, C. Otto, and P. C de Groen, “*A wireless body area network of intelligent motion sensors for computer assisted physical rehabilitation*”, Journal of NeuroEngineering and Rehabilitation 2005, DOI:10.1186/1743-0003-2-6
98. C. Otto, A. Milenković, C. Sanders, E. Jovanov, “*System architecture of a wireless body area sensor network for ubiquitous health monitoring*”, university of alabama in Huntsville, M.S thesis received October 1, 2005
99. U.S Department of Health and Human Services, “*What you should know about preventing and controlling high blood pressure*”, [hp2010.nhlbi.nih.net/mission/partner/should\\_know.pdf](http2010.nhlbi.nih.net/mission/partner/should_know.pdf) [Last accessed on 01/01/2009]
- 100.G. Mancia, G. Parati, “*Ambulatory Blood Pressure Monitoring and Organ Damage*”, AHA, Hypertension, 2000
- 101.“*Your Guide To Physical Activity and Your Heart*”, U.S. Department of Health and Human Services, National Institutes of Health, National Heart, Lung, and Blood Institute, NIH Publication NO.06-5714, June, 2006
- 102.[www.wimaxforum.org/home/](http://www.wimaxforum.org/home/) [Last accessed on 01/01/2009]
- 103.I. Howitt, “*IEEE 802.15.4 Low Rate –Wireless Personal Area Network Coexistence Issues*”, Wireless Communications and Networking, WCNC IEEE, March 16-20, 2003, Volume 3, pp1481-1486
- 104.Electromyography (EMG), [en.wikipedia.org/wiki/Electromyography](http://en.wikipedia.org/wiki/Electromyography) [Last

accessed on 01/01/2009]

- 105.R. Fensli, “A *Wearable ECG-recording System for Continuous Arrhythmia Monitoring in a Wireless Tele-Home-Care Situation*”, CBMS Proceedings of the 18th IEEE Symposium on Computer-Based Medical Systems, 2005, pp407-412, ISBN: 063-7125, ISSN: 0-7695-2355-2
- 106.MIT-BIH ECG Record 100,  
[www.physionet.org/physiobank/database/html/mitdbdir/records.htm#100](http://www.physionet.org/physiobank/database/html/mitdbdir/records.htm#100) [Last accessed on 01/01/2009]
- 107.K. Bert-Uwe, C. Hennig, and R.Orglmeister, “*The Principles of Software QRS Detection*”, Engineering in Medicine and Biology Magazine, IEEE, January/February, 2002, Volume 21, Issue 1, pp42-57, ISSN: 0739-5175
- 108.V. X. Afonso, “*ECG QRS Detection*”, Endocardial Solutions, 1350 Energy Lane, 55108, Saint Paul, MN, USA
109. A. F. Rickards, “*An Implantable Intracardiac Accelerometer for Monitoring Myocardial Contractility*”, Pacing and Clinical Electrophysiology, December, 1996, Volume 19, Issue 12, pp2066-2071
- 110.MATLAB, [www.mathworks.com/](http://www.mathworks.com/) [Last accessed on 01/01/2009]
- 111.C. L. Shen, T. Kao, C. T. Huang, J. H. Lee, “*Wearable Band Using a Fabric-Based Sensor for Exercise ECG Monitoring*”, Wearable Computers, 2006, 10th IEEE International Symposium pp143-144
- 112.“*SD Memory Card Specifications*”, Matsushita Electric Industrial Co. Ltd, SanDisk Corporation, Toshiba Corporation.

- 113.IEEE, [www.ieee.org](http://www.ieee.org) [Last accessed on 01/01/2009]
- 114.C. Papadelis, C. Kourtidou-Papadeli, P. D. Bamidis, I. Chouvarda, D. Koufogiannis, E. Bekiaris and N. Maglaveras, “*Indicators of Sleepiness in an ambulatory EEG study of night driving*”, Proceedings of the 28th IEEE EMBS Annual International Conference, New York City, USA, August 30-September 3, 2006
- 115.P. P. Vaidyanathan, “*Multirate Systems and Filter Banks*”, Englewood Cliffs, NJ: Prentice-Hall.
- 116.K.U.Leuven, “*Digital Signal Processing II, Filter Banks – Preliminaries*”, Marc Moonen, Dept. E.E./ESAT, [www.homes.esat.kuleuven.be/~moonen](http://www.homes.esat.kuleuven.be/~moonen) [Last accessed on 01/01/2009]
- 117.H.G.Goovaerts, H.H.Ros, T.J.vaden Akker, and H.Schneider, “*a digital QRS detector based on the principle of contour limiting*”, IEEE Transactions on Biomedical Engineering, 1976, Volume BME-23, p154
- 118.N.V.Thakor, J.G.Webster, and W.J.Tompkins, “*Optimal QRS detector*”, Med.Biol.Eng, 1983, Volume 21, pp343-350
- 119.M. L Ahlstrom,. and W. J. Tompkins, “*Digital filters for real-time ECG signal processing using microprocessors*”, IEEE Transactions on Biomedical Engineering, 1985, Volume BME-32, pp708–713
- 120.J. Fraden and M.R. Neumann, “*QRS wave detection*”, Med. Biol. Eng. Comput, 1980, Volume 18, pp125-132
- 121.W.P. Holsinger, K.M. Kempner, and M.H.Miller, “*A QRS preprocessor based on*

- digital differentiation*”, IEEE Transactions on Biomedical Engineering, May, 1971, Volume 18, pp121-217
- 122.L. Keselbrener, M. Keselbrener, and S.Akselrod, “*Nonlinear high pass filter for R-wave detection in ECG signal*”, Medical Engineering and Physics, 1997, Volume 19, No. 5, pp481-484
- 123.F. Jager, G. B. Moody and R. G. Mark, “*Detection of Transient ST Segment Episodes During Ambulatory ECG Monitoring*”, Faculty of Computer and Information Science, University of Ljubljana, 1001 Ljubljana, Slovenia, Division of Health Sciences and Technology, Harvard–Massachusetts Institute of Technology, and Cambridge, Massachusetts 02139, Thesis was received on April 28, 1998
- 124.T.A. Gyaw and S.R. Ray, “*The wavelet transform as a tool for recognition of biosignals*”, Biomed. Sci. Instrum, 1994, Volume 30, pp63-68
- 125.S. Mallat and W.L. Hwang, “*Singularity detection and processing with wavelets*”, IEEE Trans. Inform. Theory, 1992, Volume 38, pp617-643
- 126.S. Mallat, “*Zero-crossings of a wavelet-transform*”, IEEE Trans. Inform. Theory, 1991, Volume 37, pp1019-1033
- 127.G. Vijaya, V. Kumar, and H.K. Verma, “*ANN-based QRS-complex analysis of ECG*”, J.Med. Eng. Technol, 1998, Volume 22, No. 4, pp160-167
- 128.R. Silipo and C. Marchesi, “*Artificial Neural Networks for Automatic ECG Analysis*”, IEEE, IEEE Transactions on Signal Processing, May 1998, Volume 46, No. 5

- 129.D.A. Coast, R.M. Stern, G.G. Cano and S.A.Briller, "*An approach to cardiac arrhythmia analysis using hidden Markov models*", IEEE Transactions on Biomedical Engineering, 1990, Volume 37, pp826-836
- 130.L.R. Rabiner, "*A tutorial on hidden Markov models and selected applications in speech recognition*", Proc. IEEE, February, 1989, Volume 77, pp257-286
- 131.C.-H.H. Chu and E.J. Delp, "*Impulsive noise suppression and background normalization of electrocardiogram signals using morphological operators*", IEEE Transactions on Biomedical Engineering, 1989, Volume 36, pp262-273
- 132.P.E. Trahanias, "*An approach to QRS complex detection using mathematical morphology*", IEEE Transactions on Biomedical Engineering, 1993, Volume 40, No. 2, pp201-205
- 133.A. Ruha, S. Sallinen, and S. Nissila, "*A real-time microprocessor QRS detector system with a 1-ms timing accuracy for the measurement of ambulatory HRV*", IEEE Transactions on Biomedical Engineering, 1997, Volume 44, pp159-167
- 134.R. Poli, S. Cagnoni, and G. Valli, "*Genetic design of optimum linear and nonlinear QRS detectors*", IEEE Transactions on Biomedical Engineering, 1995, Volume 42, pp1137-1141
- 135.S.-K. Zhou, J.-T. Wang and J.-R. Xu, "*The real-time detection of QRS-complex using the envelop of ECG*", in Proc. 10th Annu. Int. Conf, IEEE Engineering in Medicine and Biology Society, New Orleans, LA, 1988, p38
- 136.M.-E. Nygard and L. Sörnmo, "*Delineation of the QRS complex using the envelope of the ECG*", Med. Biol. Eng. Comput, 1983, Volume 21

- 137.F. Gritzali, G. Frangakis, and G. Papakonstantinou, “*A comparison of the length and energy transformations for the QRS detection*”, in Proc. 9th Annu. Conf, IEEE Engineering in Medicine and Biology Society, Boston, MA, 1987, pp549-550
- 138.E.J. Ciaccio, S.M. Dunn, and M. Akay, “*Biosignal pattern recognition and interpretation systems*”, IEEE Eng. Med. Biol. Mag, 1994, pp269-273
- 139.G. Belforte, R. De Mori, and F. Ferraris, “*A contribution to the automatic processing of electrocardiograms using syntactic methods*”, IEEE Transactions on Biomedical Engineering, March, 1979, Volume 26, pp125-136
- 140.P.O. Börjesson, O. Pahlm, L. Sörnmo, and M.-E. Nygards, “*Adaptive QRS detection based on maximum a posteriori estimation*”, IEEE Transactions on Biomedical Engineering, May, 1982, Volume 29, pp341-351
- 141.K. Raymond, “*Reference Model of Open Distributed Processing (RM-ODP): Introduction, CRC for Distributed Systems Technology*”, Centre for Information Technology Research, University of Queensland, Brisbane 4072 Australia
- 142.[www.java.sun.com](http://www.java.sun.com) [Last accessed on 01/01/2009]
- 143.D. Flanagan, “*Java in a Nutshell: A Desktop Quick Reference (Java Series) (3<sup>rd</sup> Edition)*”, Publisher: O’Reilly, November, 1999, ISBN-10: 1565924878, ISBN-13: 978-1565924871
- 144.[www.olimex.com](http://www.olimex.com) [Last accessed on 01/01/2009]
- 145.[www.securityfocus.com/infocus/1836](http://www.securityfocus.com/infocus/1836) [Last accessed on 09/01/2009]



## GLOSSARY

Ambulatory	A process whereby a patient wears a small recorder called a
Electrocardiography	Holter monitor as he or she goes about normal daily life. The machine makes a record of the heart's electrical currents.
Arrhythmia	It is any of a group of conditions in which the electrical activity of the heart is irregular or is faster or slower than normal.
AV Node	Atrioventricular node, which is an area of specialized tissue between the atria and the ventricles of the heart, which conducts the normal electrical impulse from the atria to the ventricles.
API	Application Programming Interface
ADC	Analogue Digital Converter, which is an electronic integrated circuit, which converts continuous signals to discrete digital numbers.
Abstract Factory	Java Design Pattern, which is a general repeatable solution to a commonly occurring problem in Java software design.
ARM	Advanced RISC Machine, which is a series of low-cost, power-efficient 32-bit RISC microprocessors for embedded control, computing, digital signal processing, games, consumer multimedia and portable applications.

AWT	Java API Package
Avelink	It is targeted to device manufacturers that want the ability to design and produce mobility products. Its BT solutions and SDK is a complete package of Bluetooth Protocol Stack.
Avetana	The avetanaBluetooth JSR-82 implementation provides Bluetooth functionalities to Java software in a standardized way
Bluetooth SIG	The Bluetooth Special Interest Group
Bluetooth	An industrial specification for wireless personal area networks (PANs) and it is based on IEEE 802.15.1.
Bradycardia	It applied to adult medicine, is defined as a resting heart rate of under 60 beats per minute, though it is seldom symptomatic until the rate drops below 50 beat/min.
Burst transmission	Any relatively high-bandwidth transmission over a short period of time. For example, a download might use 2Mbps on average, whilst have "peaks" bursting up to, say, 2.4Mbps.
CDC	Connected Device Configuration, which is a framework for building Java ME applications on embedded devices ranging from pagers up to set-top boxes.
crème	NSICOM's KVM for the iPAQ 5550
CORBA	Common Object Request Broker Architecture, which is a standard defined by the Object Management Group (OMG)

that enables software components written in multiple computer languages and running on multiple computers to work together.

DSSS	Direct-Sequence Spread Spectrum
EMG	Electromyography
ECA	Event Condition Action, which is a short-cut for referring to the structure of active rules in event driven architecture and database systems.
ECG, EKG	Electrocardiogram, which is a graphic produced by an electrocardiograph, which records the electrical activity of the heart over time.
ETSI	European Telecommunications Standards Institute, which is an independent, non-for-profit, standardization organization of the telecommunications industry (equipment makers and network operators) in Europe, with worldwide projection. ETSI has been successful in standardizing the GSM cell phone system and the TETRA professional mobile radio system.
FIR Filter	Finite Impulse Response Filter, which is a filter structure that can be used to implement almost any sort of frequency response digitally. An FIR filter is usually implemented by using a series of delays, multipliers, and adders to create the

filter's output.

GPRS	General Packet Radio Service, which is a Mobile Data Service available to users of Global System for Mobile Communications (GSM) and IS-136 mobile phones.
Holter Monitor	Also called an ambulatory electrocardiography device, named after its inventor, Dr. Norman J. Holter, is a portable device for continuously monitoring the electrical activity of the heart for 24 hours or more.
IEEE	Institute of Electrical and Electronics Engineers
Java	a family of computer software products and specifications from Sun Microsystems that together provide a system for developing application software and deploying it in a cross-platform environment
JAR	Java Archive (File format)
J2ME	Java Platform, Micro Edition – the version of Java that is used for limited resource devices
JCP	Java Community Process
JSR82	The Java API for Bluetooth
Jeode	Insignia's KVM
Jbed	Esmertec's KVM
JVM	Java virtual machine, a process that is resident on a host machine or device which allows the Java run time

	environment to be relatively independent of platform.
JAR	Java Archive (File format)
KVM	Kilobyte virtual machine an efficient JVM that is used for limited memory devices and some embedded systems
L2CAP	Logical Link Control and Adaptation Protocol
MIPS	Microprocessor without Interlocked Pipeline Stages
MEMS	Micro-Electro-Mechanical-Systems, which is the technology of the very small, and merges at the nano-scale into nanoelectromechanical systems (NEMS) and Nanotechnology.
MCOT	Mobile cardiac outpatient telemetry (MCOT) systems
OMG	The Object Management Group the organization responsible for UML and CORBA.
ODP	Open Distributed Processing, which is a method of computer processing in which different parts of a program are run simultaneously on two or more computers that are communicating with each other over a network.
PJSCP	The emulator of CrEme
Predictivity	$\frac{TruePositive}{TruePositive + FalsePositive}$
QRS Complex	The periodic 'high amplitude' section between Q wave and S wave in an ECG
RFCOMM	The Serial Cable Emulation Protocol in JSR82

RS232	A standard for serial binary data signals connecting between DTE (Data terminal equipment) and DCEe (Data Circuit-terminating Equipment) enabled devices
RRI	R wave to R wave Interval
SAECA	Signal Algorithm Event Condition Action
Sensitivity	$\frac{TruePositive}{TruePositive + FalseNegative}$
SQL	Structured Query Language
strongARM	is a faster version of the Advanced RISC Machines ARM design
SuperH(SH, SH-3)	It is fundamentally a 32-bit load/store RISC architecture found in a large number of embedded systems.
Swing	Java API Package
Tachyarrhythmia	It is an abnormally fast heart rate (usually 100 to 400 beats) in either the upper heart chambers or lower heart chambers.
UML	The Unified Modeling Language
WPAN	Wireless Personal Area Network
Wi-Fi	It is intended to improve the interoperability of wireless local area network products based on the IEEE 802.11 standards
WiMAX	Worldwide Interoperability for Microwave Access
X86	It refers to the instruction set of the most commercially

successful CPU architecture

XScale

It is a microprocessor core, is Marvell's (formerly Intel's) implementation of the fifth generation of the ARM architecture, and consists of several distinct families: IXP, IXC, IOP, PXA and CE

## APPENDIX A DETECTION ALGORITHMS

### A.1 Alfonso Filter Banks

Mr Valtino X. Afonso developed a multi-rate digital signal processing algorithm to detect heartbeats in ECG, which incorporates a filter bank (FB) that decomposes the ECG into sub-bands with uniform frequency bandwidths [96].

A digital filter bank is a collection of digital filters with a common input and a common output [115] that is illustrated in Figure A.1.1,

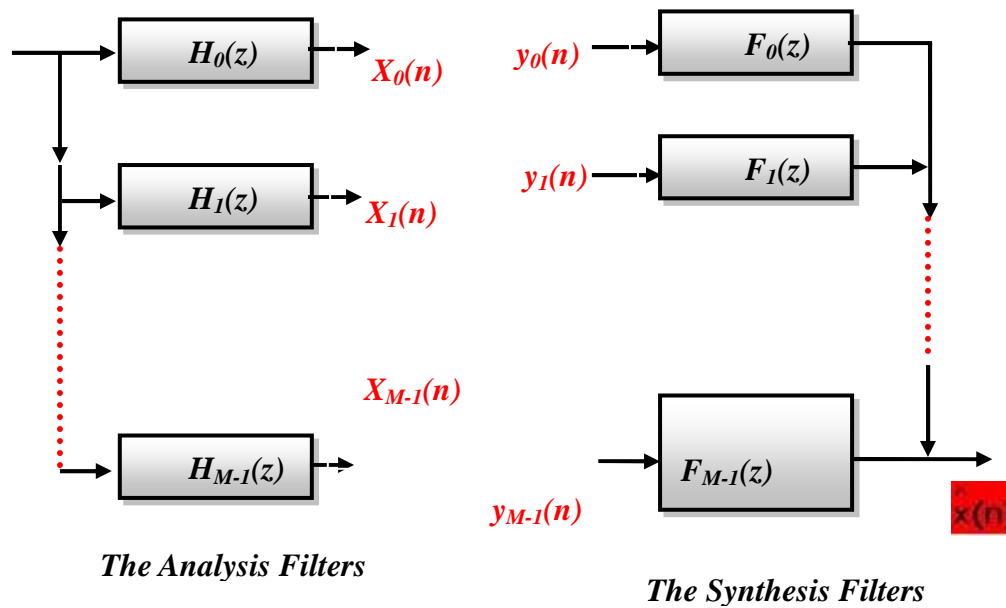


Figure A.1.1 Digital filter banks [115]

Inside Figure A.1.1, the analysis filters are the left side and the synthesis filters are on



the right side. The analysis filter bank split the signal  $x(n)$  into  $M$  signals  $x_k(n)$  typically called sub-band signals. The synthesis filter bank can combine the  $M$  sub-band signals into a single signal  $\hat{x}(n)$  that is very similar to the original signal  $x(n)$ . Because down-sampling introduces aliasing, the synthesis filters are designed to remove aliasing effects [51]. The filters are FIR and result in a FB which no aliasing, or magnitude and phase distortions.

The filter response for the analysis filters is shown below, which is marginally overlapping and provides a smooth transition from one block to the next. This is important if the frequency domain processing varies with time.

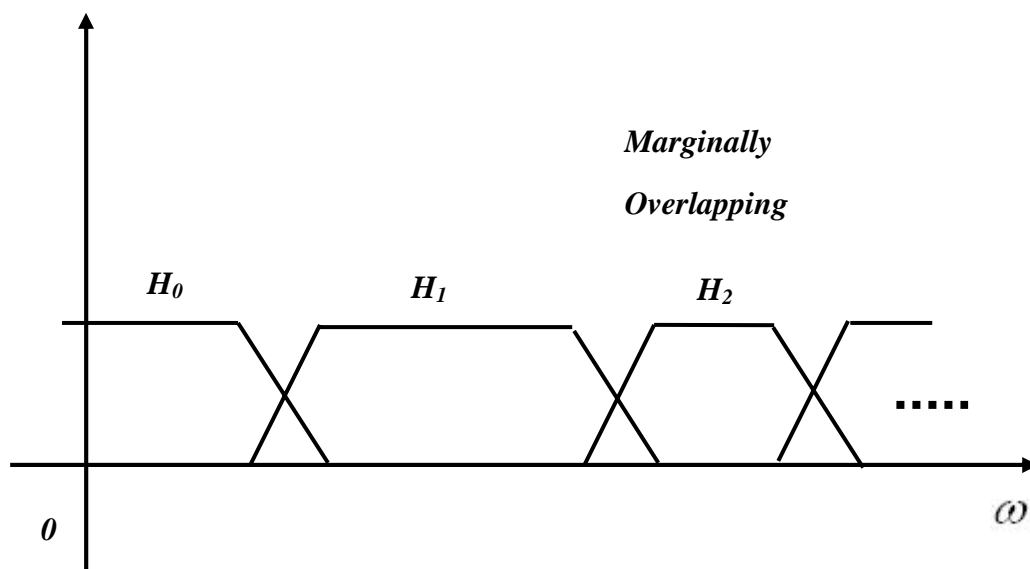


Figure A.1.2 Filter response [77]

An incoming signal can be decomposed into specific frequency bands or sub-bands by the analysis filters, and this algorithm involves decomposing a signal into frequency sub-bands, processing these sub-bands according to the application at hand, and then sometimes reconstructing the processed sub-bands.

Four bandpass FIR filters are developed with the uniform frequency bandwidth 5.6 Hz, and the filter transfer function is as follows.

$$H(x) = \frac{\cos(\frac{2\pi[f_{P2}-0.2]}{f_s}) - \cos(\frac{2\pi f_{P1}}{f_s})}{\pi} \quad \text{Equation A.1.1}$$

where  $f_s$  is sampling frequency,  $f_{P1}$  and  $f_{P2}$  are cut-off frequency.

Therefore the frequency responses of the four bandpass filters are plotted in the following figure,

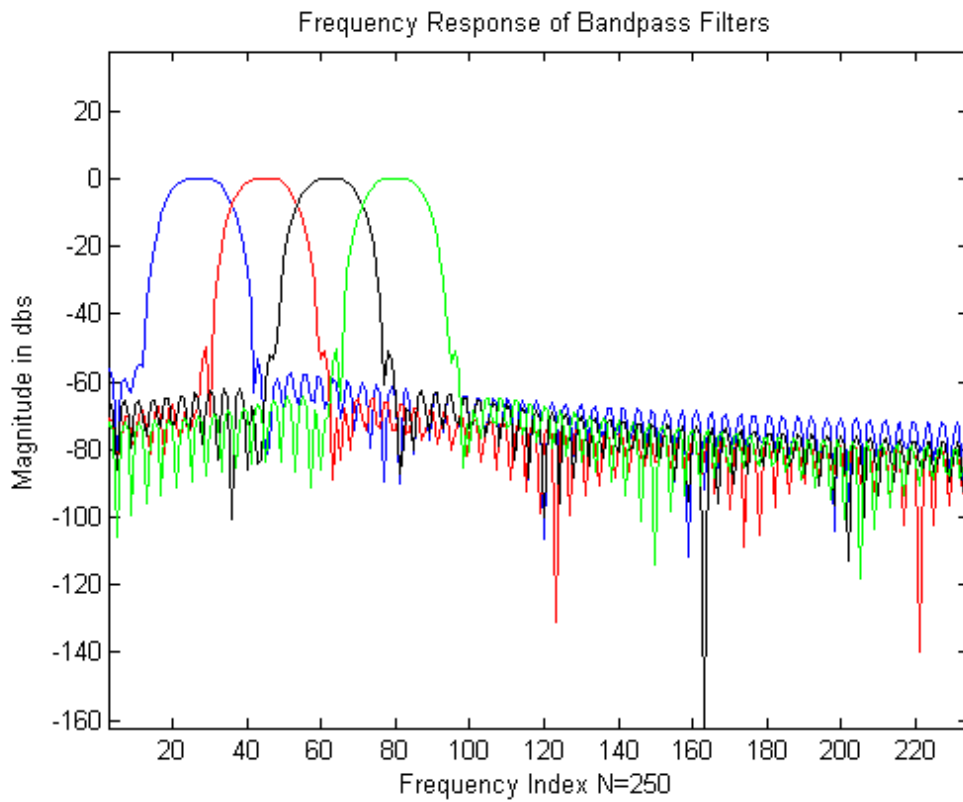


Figure A.1.3 Frequency response of the bandpass filters

(Frequency is normalized to 1)

The ECG signal (7.2 seconds ECG signals that has 3600 samples with 500 sampling frequency) is inputted into the four bandpass filters separately. Therefore the four sub-band signal waveforms are generated in Figure A.1.4. Signal decomposition for further ECG analysis such as Beat Detection, Beat Classification, ECG Enhancement and Noise Alert is based on these waveforms.

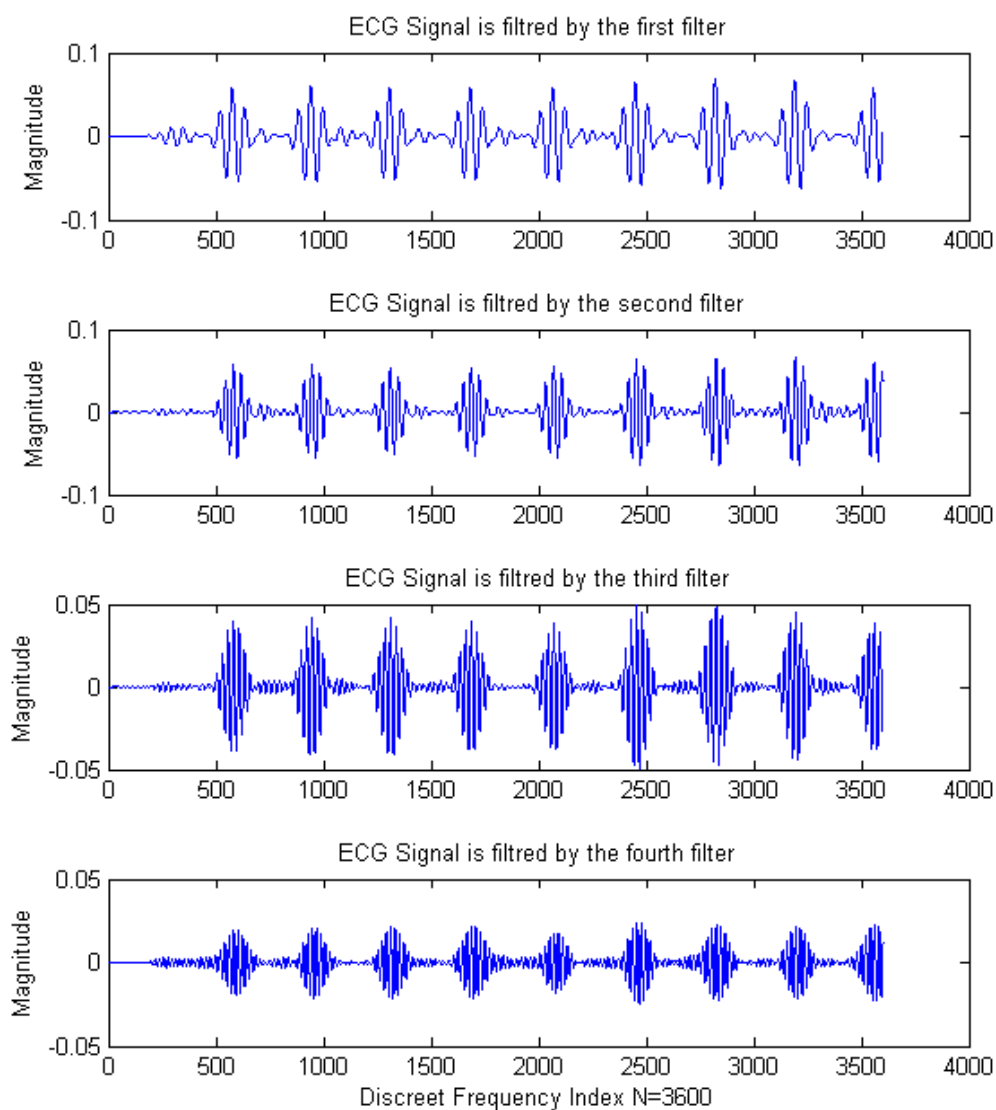


Figure A.1.4 ECG signals are filtered by filters

In Figure A.1.1, a Filter Bank contains  $M$  analysis filters and  $M$  synthesis filters, each

of length  $L$ . The analysis filters

$$H_l(z) = 0, 1, \dots, M-1 \quad \text{Equation A.1.2}$$

bandpass the input signal  $X(z)$  to produce the subband signals  $U_l(z)$  [96],

$$U_l(z) = H_l(z)X(z) \quad \text{Equation A.1.3}$$

that is illustrated in Figure A.1.4.

The downsampling process keeps one sample out of a number of samples. The downsampled signal is the following,

$$W_l(z) = \frac{1}{M} \sum_{k=0}^{M-1} U_l(z^{1/M} W^k) \quad l = 0, 1, \dots, M-1 \quad \text{Equation A.1.3}$$

Where  $W = e^{-j(2\pi/M)}$ , the subbands  $U_l(z)$  and  $W_l(z)$  are bandpassed versions of the input and also  $W_l(z)$  has a lower rate than  $U_l(z)$ . The filtering process can be efficiently applied at  $1/M$  the input rate by taking advantage of the downsampling. This process is referred to as the polyphase implementation and contributes to the computational efficiency of the FB based algorithm [115].

The four downsampled ECG  $W_l(z)$  is plotted in Figure A.1.5

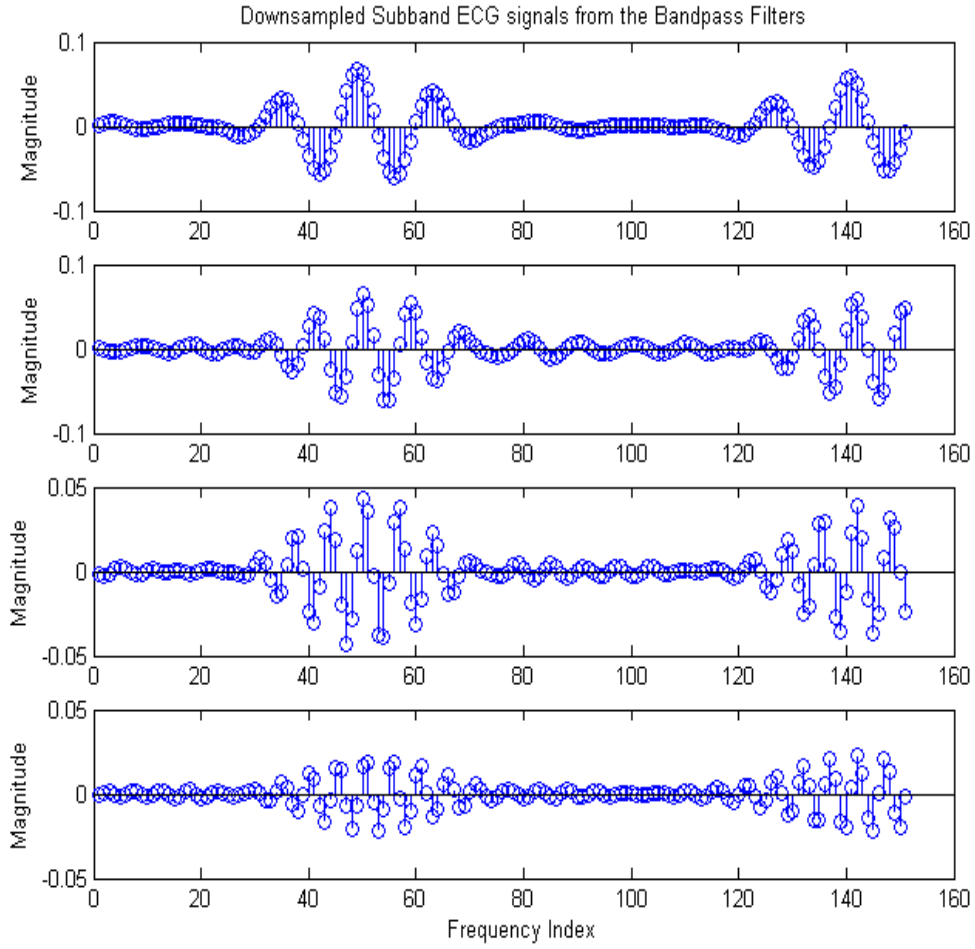


Figure A.1.5 Downsampled Subband ECG Signals  $W_l(z)$

A variety of features which are indicative of the QRS complex can be designed by combining subbands of interest from  $l = 0, 1, \dots, M - 1$ . For example a sum-of-absolute values feature can be computed using subbands 1, 2, and 3,

$$P_l = \sum_{l=1}^3 |W_l(z)| \quad \text{Equation A.1.4}$$

Where  $P_1$  has a value which corresponds to the energy in the frequency band [5.6, 22.5] Hz. Similarly,  $P_2$  and  $P_3$  can be computed using subbands  $\{1, 2, 3, 4\}$ , and

$\{2, 3, 4\}$ , respectively, and these values are proportional to the energy in their respective subbands.

These features have values which are proportional to the energy of the QRS complex, so the beat detection process occurs at the subband rate instead of the input ECG rate [76]. The Figure A.1.6 illustrates one of the feature signals  $P_1$ . In order to locate the heart beats, a moving window integration needs to be applied to the feature signal  $P_1$ .

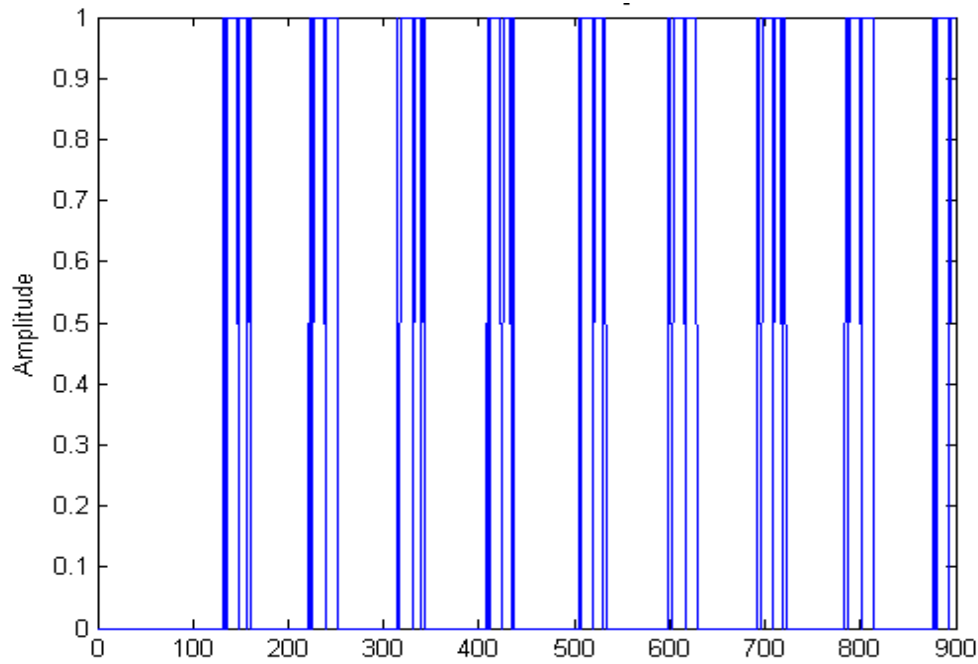


Figure A.1.6 Feature signal  $P_1$

According to Bert-Uwe Köhler [107], the computational load of this algorithm is medium. That is not the best choice for a low-memory device such as PDA in this project. Therefore the author decides not to go further for the implementations of this algorithm.

## **A.2 Differentiation technique**

Differentiation forms the basis of many QRS detection algorithms. Since it is basically a high-pass filter, the derivative amplifies the higher frequencies characteristic of the QRS complex while attenuating the lower frequencies of the P and T waves [61].

An algorithm based on first and second derivatives originally developed by Balda et al [62], was modified for use in high-speed analysis of recorded ECGs by Ahlstrom and Tompkins [70]. Friesen et al [64] subsequently implemented the algorithm as part of a study to compare noise sensitivity among certain types of QRS detection algorithms.

This differentiation algorithm calculates the first and second derivative from the ECG signal and then squares the sum of the two derivatives to generate a pulse output to indicate the QRS Complex. Figure A.2.1 illustrates the whole procedure of this algorithm.

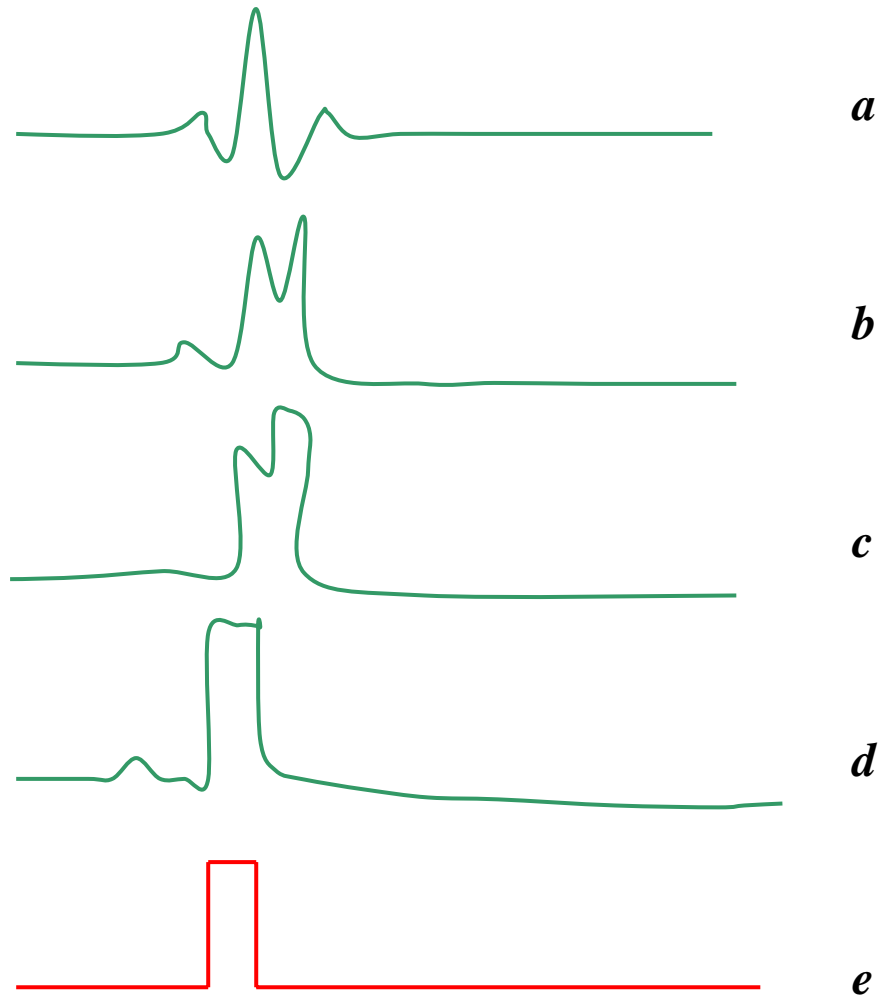


Figure A.2.1 Different signal stages in the QRS detection based on differentiation.

Panel a, Original ECG; Panel b, Smoothed and rectified first derivative; Panel c, Smoothed and rectified second derivative; Panel d, Smoothed sum of Panel b and Panel c; Panel e, Square pulse output for each QRS complex.

All the parts in Figure A.2.1 will be discussed by MATLAB program as the following.

The first step is to filter the original ECG signal in order to attenuate the mean, the P and T wave, and the high frequency noise. Figure A.2.2 illustrates Panel a in Figure



A.2.1.

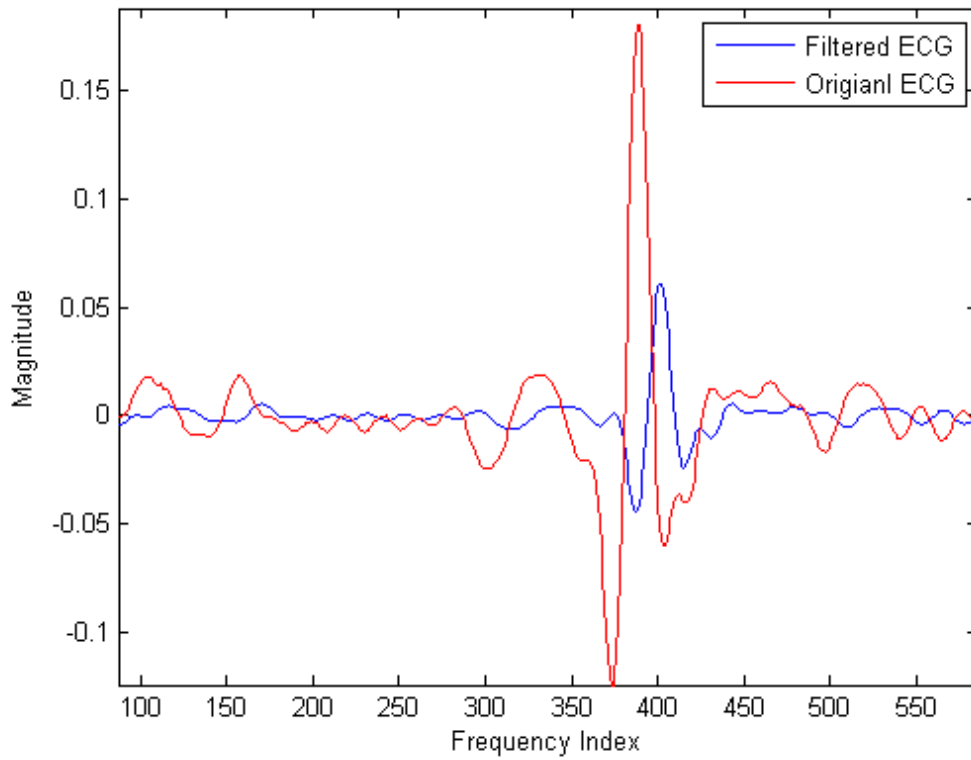


Figure A.2.2 Original ECG and Filtered ECG

After the pre-processing of the ECG signal, the first and second derivative are plotted against Panel b and c in Figure A.2.1. First derivative calculation equation is

$$y_o(nT) = |x(nT) - x(nT - 2T)| \quad \text{Equation A.2.1 [61]}$$

Second derivative calculation equation is,

$$y_l(nT) = |x(nT) - 2x(nT - 2T) + x(nT - 4T)| \quad \text{Equation A.2.2 [61]}$$

And the plot is below.

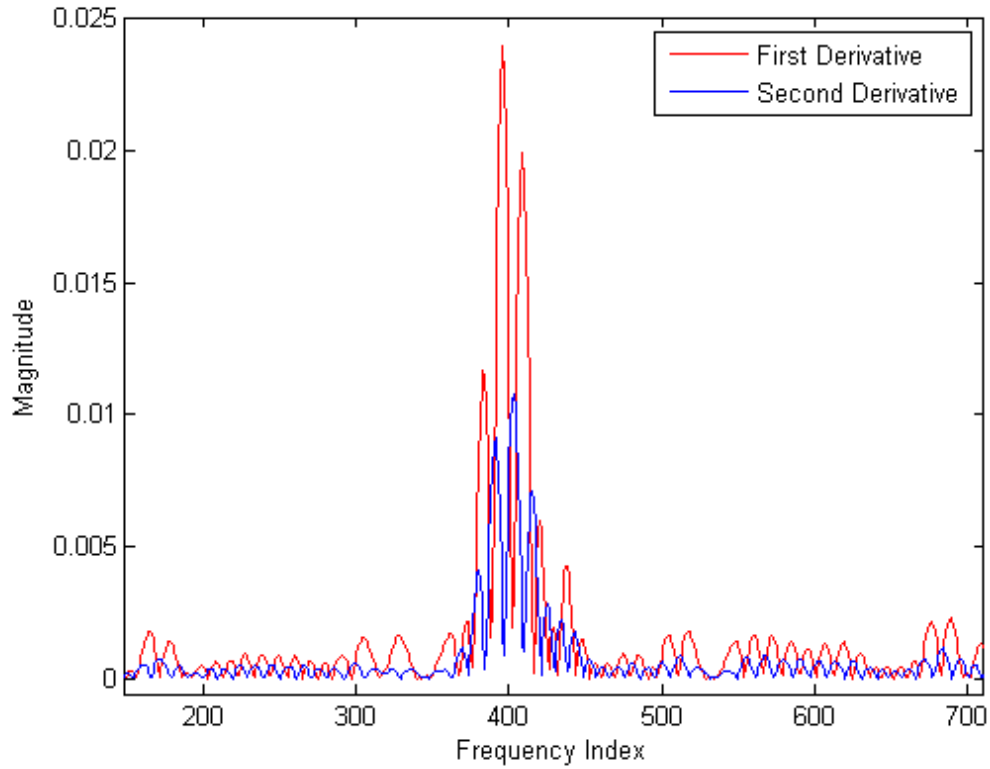


Figure A.2.3 First and second derivative

In order to generate the pulse output, the summation of the first and second derivative is calculated as following,

$$y_2(nT) = 1.3y_0(nT) + 1.1y_1(nT) \quad \text{Equation A.2.2 [61]}$$

where 1.3 and 1.1 are defined by Alfonso [61].

Now the summation of the first and second derivative is plotted in the following figure,

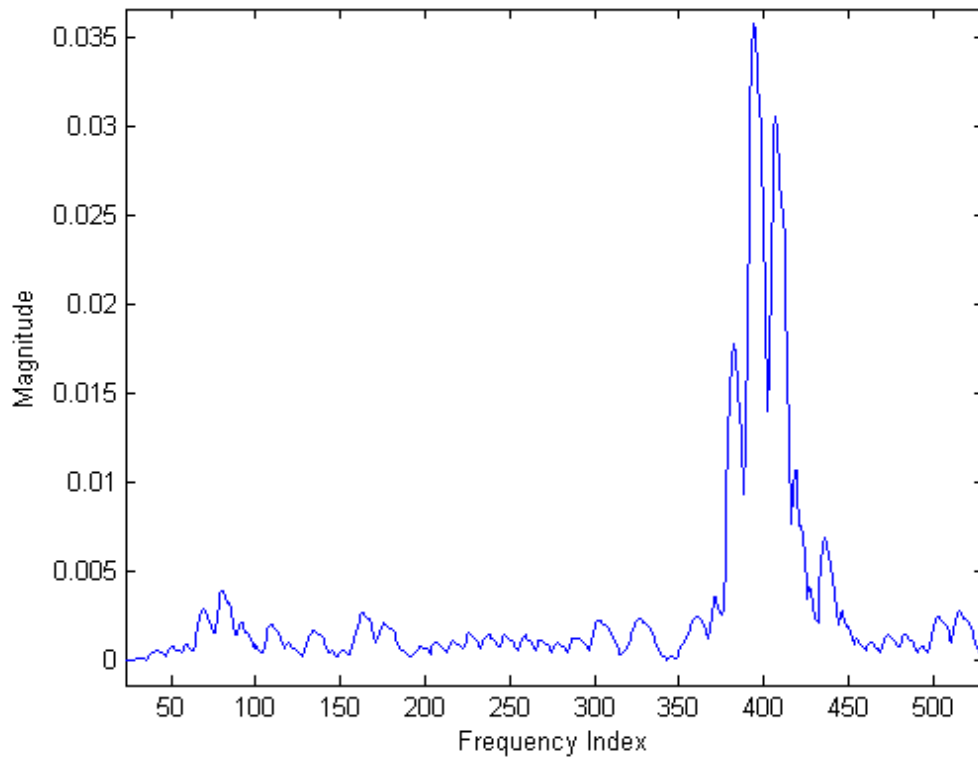


Figure A.2.4 Summation of 1<sup>st</sup> and 2<sup>nd</sup> derivative

In order to smooth the response, a decision logic and a threshold are applied to the square pulse output, therefore the following plot Figure A.2.5 is achieved as illustrated in Panel e of Figure A.2.1.

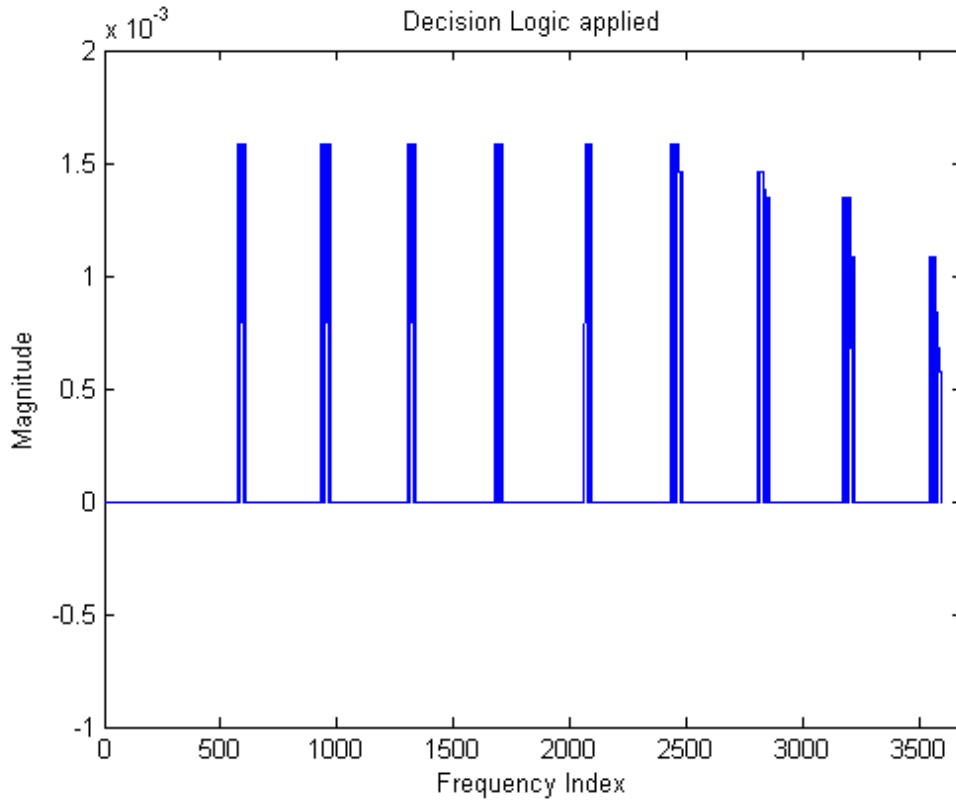


Figure A.2.5 Square pulse output

### A.3 Pan and Tompkins

A real-time QRS detection algorithm developed by Pan and Tompkins (1985) [65] was further described by Hamilton and Tompkins (1986) [66]. It recognizes QRS complexes based on analyses of the slope, amplitude, and width.

FigureA.3.1 illustrates the whole process of the ECG signal, which involves noise attenuation by bandpass filter and then differentiation, squaring and time averaging of the signal is carried out subsequently.  $Z(n)$  is the time-averaged signal.  $y(n)$  is the bandpassed ECG, and  $x(n)$  is the differentiated ECG.

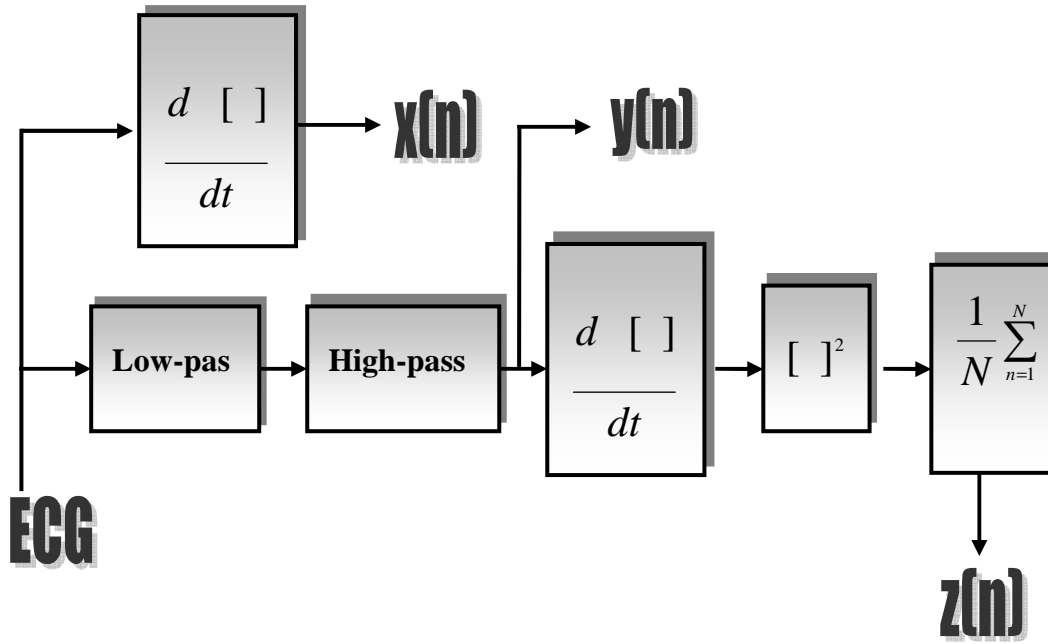


Figure A.3.1 Filter stages of the QRS detector

In Figure 4.5.1, the ECG signal is first bandpassed by a lowpass and a highpass filter to attenuate the mean, the P and T wave, and the high frequency noise. The desirable passband to maximize the QRS energy is approximately 5-15 Hz [117,118]. Then the ECG signal is differentiated to provide information about the slope of the QRS complex. Now the squaring process come into play to give the signal nonlinear operation, after this, the signal passes through a moving –window integrator. Adaptive thresholds then discriminate the locations of the QRS complexes [94].

The bandpass filter is composed by a lowpass and a highpass filter, which maximize the QRS energy is approximately 5-15 Hz [117,118] and improves the signal to noise ratio (SNR) and make the lower threshold more feasible for beat detection than unfiltered ECG [65], so the overall detection sensitivity is increased. Therefore the

transfer function and the difference equation of the lowpass is shown below in Equation A.3.1 and Equation A.3.2 respectively:

$$H(z) = \frac{(1 - z^{-6})^2}{(1 - z^{-1})^2} \quad \text{Equation A.3.1}$$

$$y(nT) = 2y(nT - T) - y(nT - 2T) - 2x(nT - 6T) + x(9nT - 12T) \quad \text{Equation A.3.2}$$

The highpass filter is based on subtracting the output of a first order lowpass filter from an all-pass filter. The transfer function and the difference equation of the highpass filter is also shown in Equation A.3.3 and Equation A.3.4 respectively:

$$H(z) = z^{-16} - \frac{H_{Lowpass}(z)}{32} \quad \text{Equation A.3.3 [108]}$$

$$y2(nT) = x(nT - 16T) - \frac{1}{32}[y(nT - T) + x(nT) - x(nT - 32T)]$$

Equation A.3.4 [108]

The overall performance of this bandpass filter has a pass-band from 5-11 Hz approximately, and the gain of the lowpass filter is 36 and 32 for the highpass filter, the filtered signal is delayed by 16 samples. The following plot Figure A.3.2 and Figure A.3.3 illustrate the difference between the filtered and the original ECG signal.

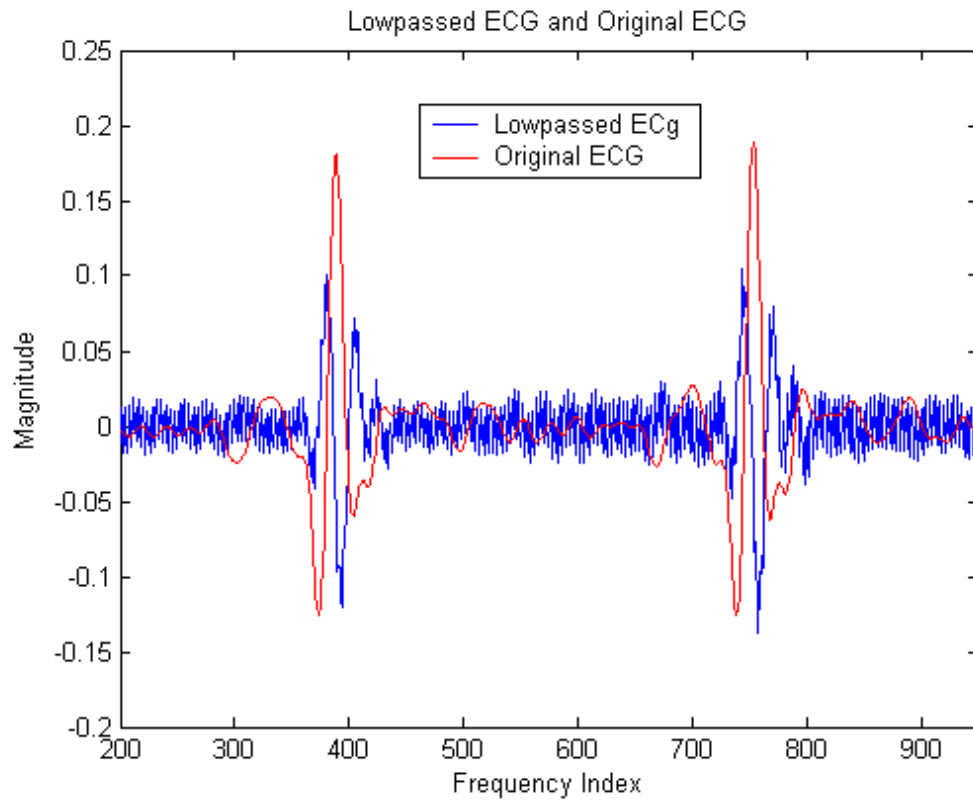


Figure A.3.2 Lowpass Filtered ECG and Original ECG

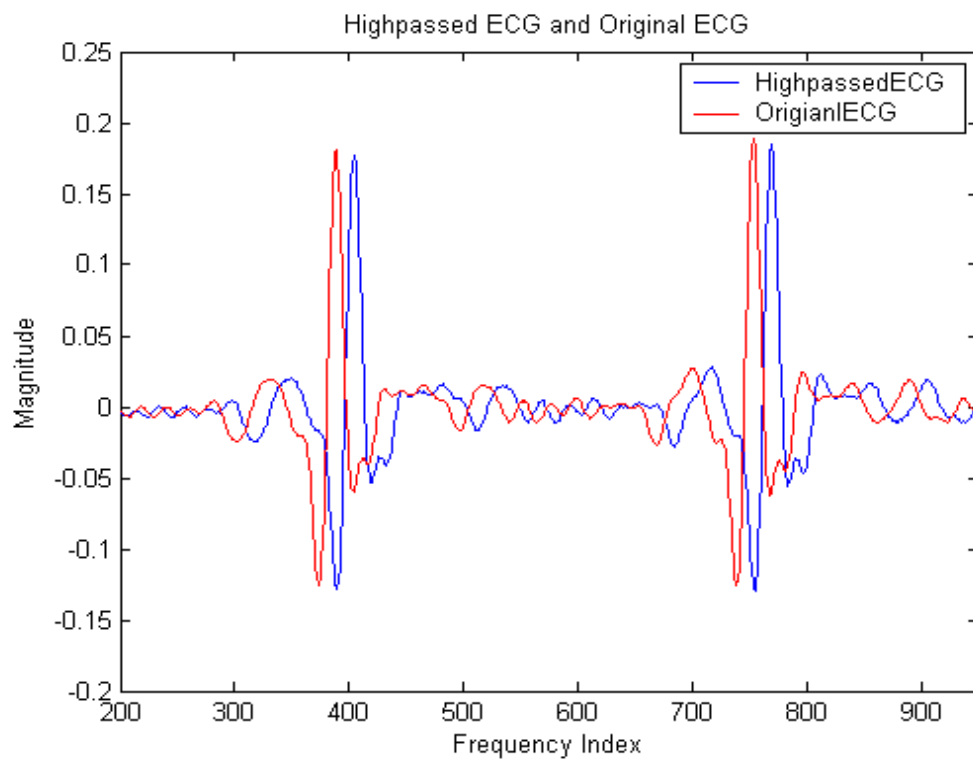


Figure A.3.3 Highpass Filtered ECG that has been lowpass filtered previously

In order to provide more information about the slope of the QRS complex, the bandpassed signal has to be differentiated. The transfer function is given by [108]:

$$H(z) = 0.1(2 + z^{-1} + z^{-3} - 2z^{-4}) \quad \text{Equation A.3.5 [108]}$$

The difference equation is shown below:

$$y(nT) = \frac{2x(nT) + x(nT - T) - x(nT - 3T) - 2x(nT - 4T)}{8} \quad \text{Equation A.3.6 [108]}$$

The fraction 1/8 is an approximation of the actual gain of 0.1. In this algorithm, the parameters with power-of-two values are approximated to facilitate the real-time operation. The differentiated ECG is plotted in Figure A.3.4 with the original ECG signal.



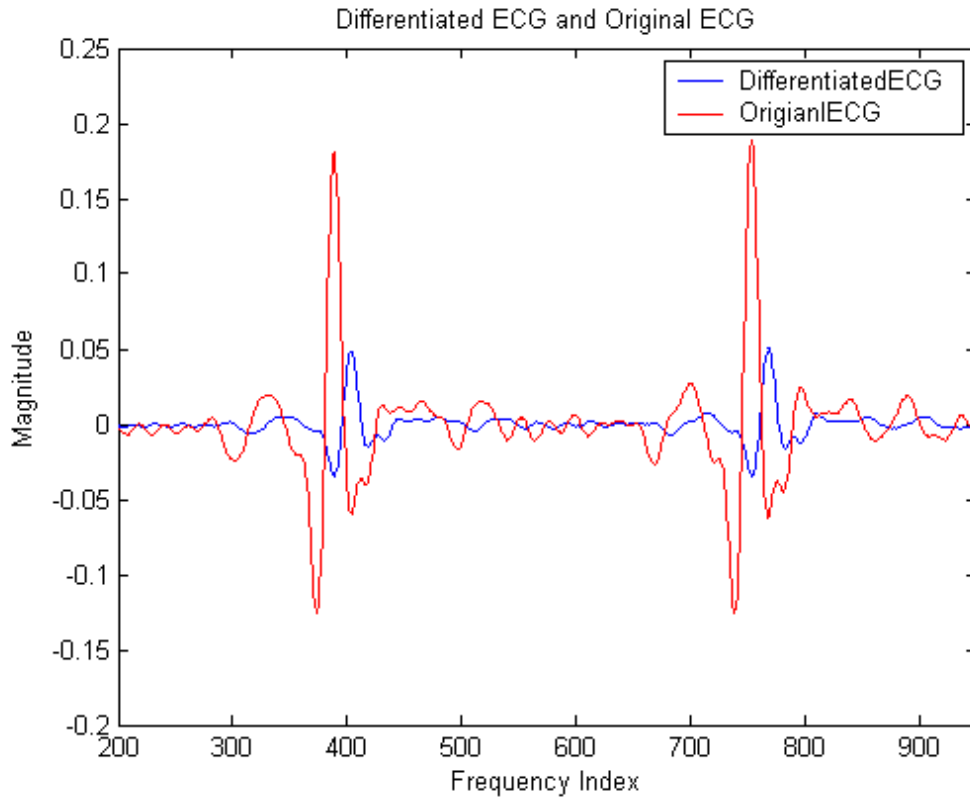


Figure A.3.4 The differentiated ECG and the original ECG

After the differentiation, the peak-to-peak signal corresponding to the QRS complex is further enhanced. The squaring function that the signal now passes through is a nonlinear operation. The equation that implements this operation is below:

$$y(nT) = [x(nT)]^2 \quad \text{Equation A.3.7 [108]}$$

This operation makes all data points in the processed signal positive, and it amplifies the output of the derivative process nonlinearly. It emphasizes the higher frequencies in the signal, which are mainly due to the QRS complex.

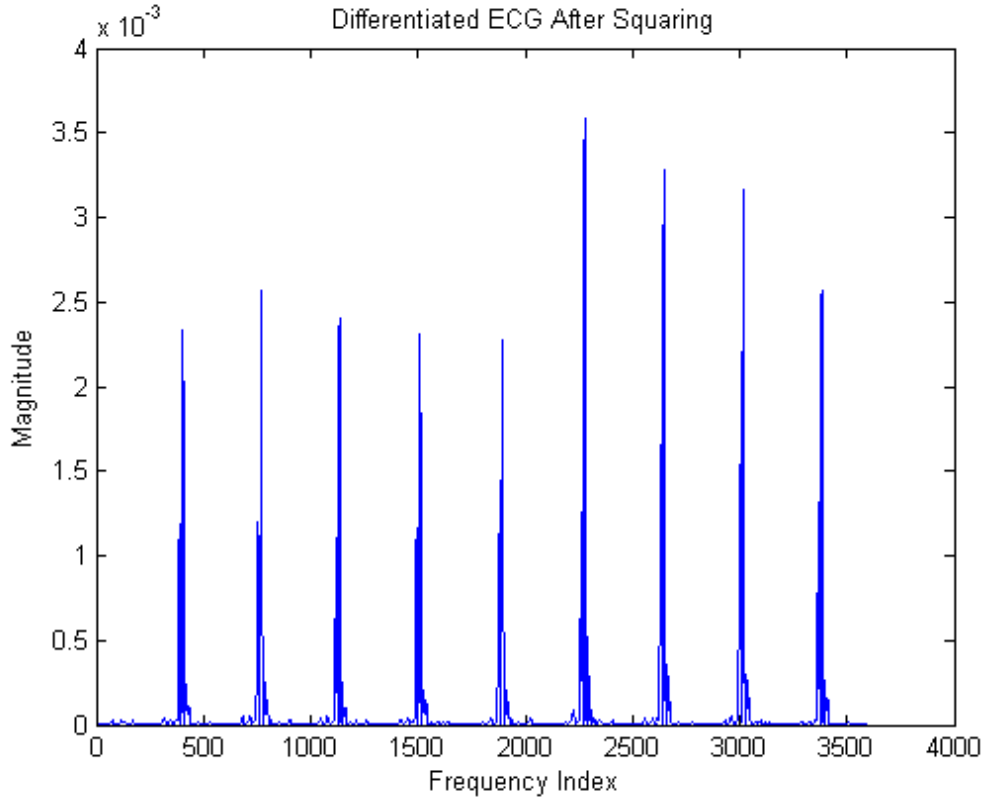


Figure A.3.5 Squared ECG after Differentiation

The slope of the R wave alone is not a guaranteed way to detect a QRS event. Many abnormal QRS complexes that have large amplitudes and long durations (not very steep slopes) might not be detected using information about slope of the R wave only. Thus, we need to extract more information from the signal to detect a QRS event [94]. Moving window integration extracts features in addition to the slope of the R wave. It is implemented with the following difference equation:

$$y(nT) = \frac{[x(nT - (N - 1)T) + x(nT - (N - 2)T) + \dots + x(nT)]}{N} \text{Equation A.3.8 [61]}$$

where  $N$  is the number of samples in the width of the moving window. Therefore the output of this process is shown in Figure A.3.5,

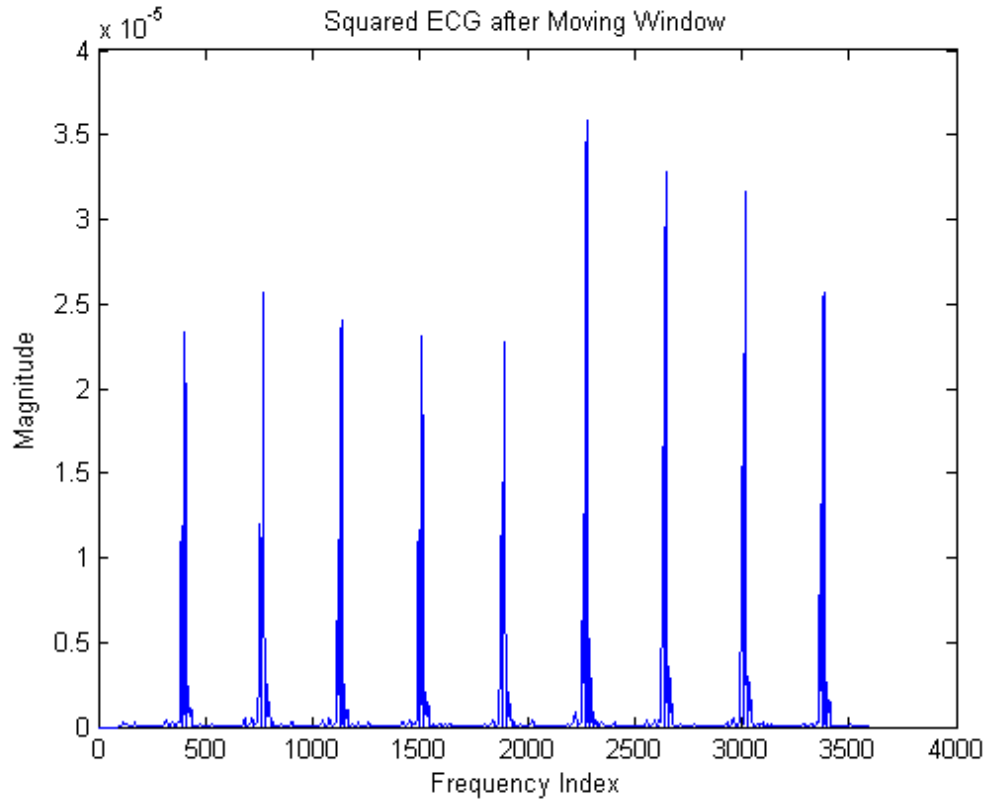


Figure A.3.5 Squared ECG after Moving Window

The width of the window should be approximately the same as the widest possible QRS complex. If the size of the window is too large, the integration waveform will merge the QRS and T complexes together. On the other hand, if the size of the window is too small, a QRS complex could produce several peaks at the output of the stage. Therefore, 100 samples for the width of the moving window integral is chosen for this particular test.

The set of thresholds that Pan and Tompkins (1985) [94] used for this stage of the QRS detection algorithm were set such that signal peaks (i.e., valid QRS complexes) were detected. Signal peaks are defined as those of the QRS complex, while noise

peaks are those of the T waves, muscle noise, etc. After the ECG signal has passed through the bandpass filter stages, its signal-to-noise ratio increases. This permits the use of thresholds that are just above the noise peak levels. Thus, the overall sensitivity of the detector improves [65]. The following equations explained the calculation of these parameters.

$$SPKI(n) = 0.125PEAKI + 0.875SPKI(n-1) \quad \text{Equation A.3.9 [94]}$$

$$NPKI(n) = 0.125PEAKI + 0.875NPKI(n-1) \quad \text{Equation A.3.10}$$

$$THRESHOLD \ I1 = NPKI + 0.25(SPKI - NPKI) \quad \text{Equation A.3.11}$$

$$THRESHOLD \ I2 = 0.5THRESHOLD \ I1 \quad \text{Equation A.3.12}$$

All the variables in these equations refer to the signal of the integration waveform and are described below [65]:

PEAKI is the overall peak.

SPKI is the running estimate of the signal peak.

NPKI is the running estimate of the noise peak.

THRESHOLD I1 is the first threshold applied.

THRESHOLD I2 is the second threshold applied

Pan and Tompkins (1985) [65] explain that a peak is determined when the signal changes direction within a certain time interval. Thus, *SPKI* is the peak that the algorithm has learned to be that of the QRS complex, while *NPKI* peak is any peak that is not related to the signal of interest. As can be seen from the equations, new values of thresholds are calculated from previous ones, and thus the algorithm adapts to changes in the ECG signal from a particular person. Whenever a new peak is detected, it must be categorized as a noise peak or a signal peak. If the peak level exceeds THRESHOLD I1 during the first analysis of the signal, then it is a QRS peak. If search back technique (explained in the next section) is used, then the signal peak should exceed THRESHOLD I2 to be classified as a QRS peak.

Based on these equations above, the parameters are plotted in Figure A.3.6,

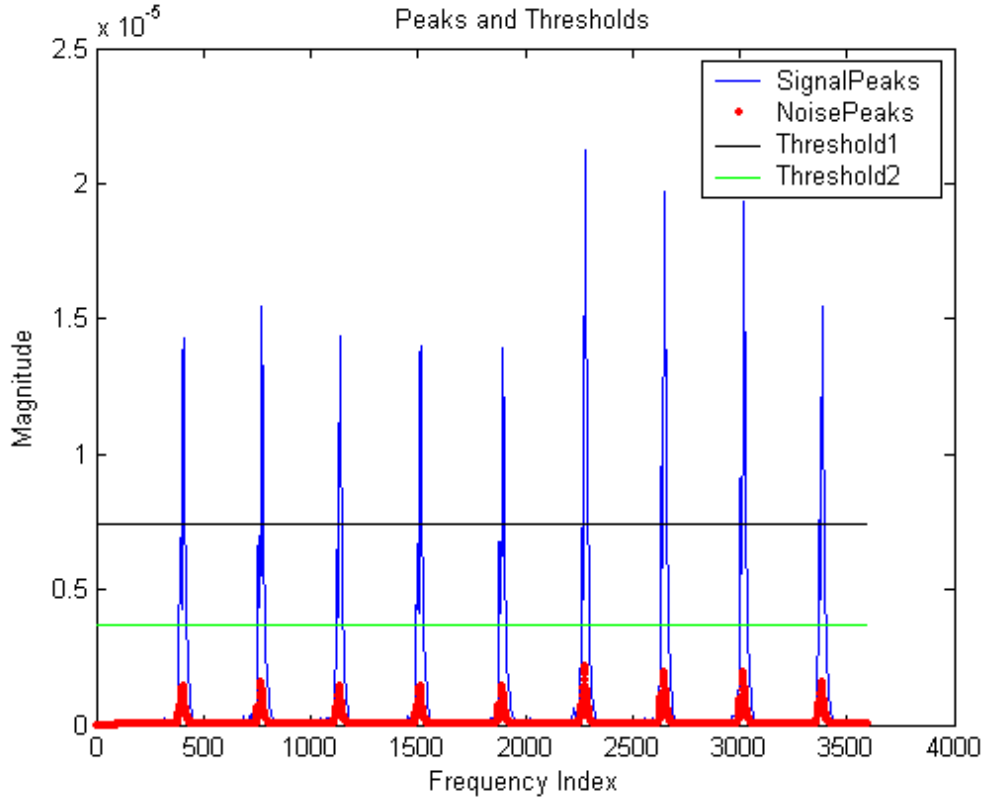


Figure A.3.6 Peaks and Thresholds

Sometimes ECG signals maybe under Threshold1 but over Threshold2, and those readings may be part of the QRS complex. Whenever Threshold2 comes into play, the search-back engine will be activated. The search-back engine is designed for detecting the missed beats. To implement the search back technique, this algorithm maintains two RR-interval averages. One average, RR AVERAGE1, is that of the eight most recent heartbeats.

$$RR \text{ AVERAGE1} = 0.125(RR_{n-7} + RR_{n-6} + \dots + RR_n) \quad \text{Equation A.3.13 [65]}$$

where  $RR_n$  is the most recent RR interval.

The other average, RR AVERAGE2, is based on the selected beats.

$$RR \text{ AVERAGE2} = 0.125(RR'_{n-7} + RR'_{n-6} + \dots RR'_n) \quad \text{Equation A.3.14 [94]}$$

The  $RR'_n$  values are the RR intervals that fell within the following limits:

$$RR \text{ Low LIMIT} = 92\% RR \text{ AVERAGE2} \quad \text{Equation A.3.15 [94]}$$

$$RR \text{ HIGH LIMIT} = 116\% RR \text{ AVERAGE2} \quad \text{Equation A.3.16 [94]}$$

Whenever the QRS waveform is not detected for a certain interval, RR MISSED LIMIT, then the QRS is the peak between the established thresholds mentioned in the previous section that are applied during search-back.

$$RR \text{ MISSED LIMIT} = 166\% RR \text{ AVERAGE2} \quad \text{Equation A.3.17 [94]}$$

The heart rate is said to be normal if each of the eight most recent RR intervals are between the limits established by RR LOW LIMIT and RR HIGH LIMIT.

## APPENDIX B JAVA CODING ALGORITHM

For the pre-processing of the incoming ECG data, the impulse response of the fir filter has 27 coefficients as follows, and those coefficients are generated from the software that is called FIR Filter Design shown below, for the pre-processing of the incoming ECG data, the impulse response of the fir filter has 27 coefficients as follows, and those coefficients are generated from the software that is called FIR Filter Design shown below,

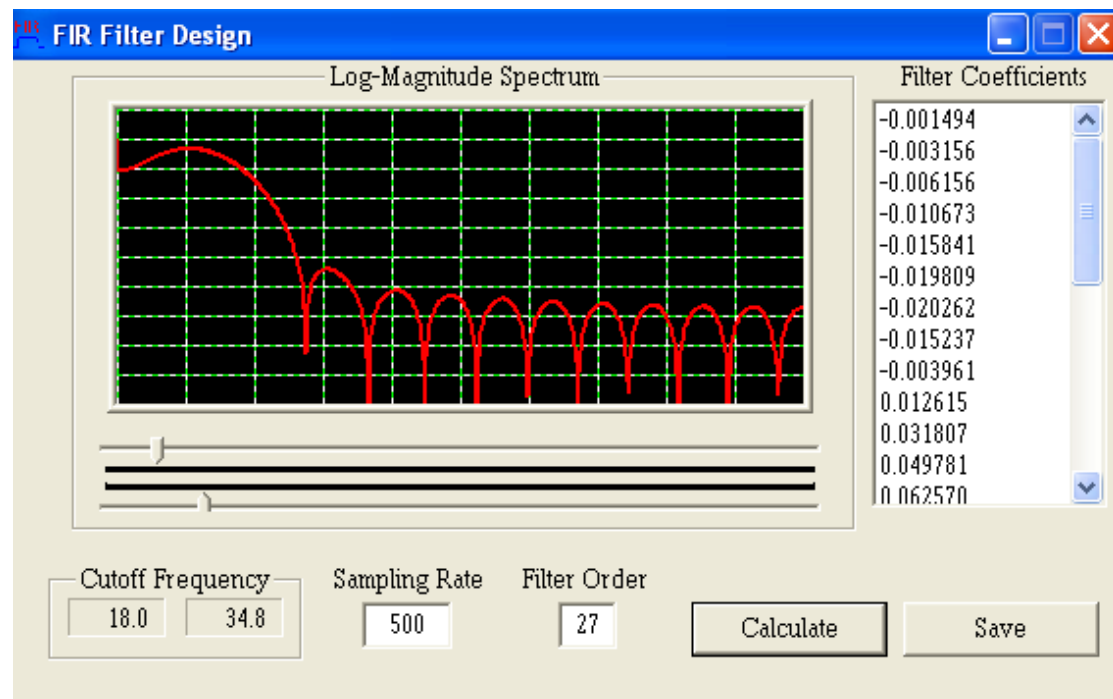


Figure 3.2.2.2.3.2 FIR Filter Design Software

*impulseResponse* = [-0.001494, -0.003156, -0.006156, -0.010673, -0.015841, -0.019809, -0.020262, -0.015237, -0.003961, 0.012615, 0.031807, 0.049781, 0.062570, 0.06720, 0.062570, 0.049781, 0.031807, 0.012615, -0.003961, -0.015237, -0.020262, -0.019809, -0.015841, -0.010673, -0.006156, -0.003156, -0.001494]



The data after the **impulseResponse** of the fir filter compared with the results from MATLAB, and the values are calculated to be same.

For non-linear transformation of the filtered ECG data, a function

***public double[ ] nonlinearTransform(double[ ] filteredECG)*** is implemented as the

following,

```
public double[ ] nonlinearTransform(double[ ] filteredECG){
    double[ ] nonlinear=new double[filteredECG.length];
    double[ ] differentResult=new double[filteredECG.length];

    for(int i=0;i<filteredECG.length;i++){
        if(filteredECG[i]>0){
            differentResult[i]=1;
        }else if(filteredECG[i]==0){
            differentResult[i]=0;
        }else{differentResult[i]=-1;    }
    }
    for(int j=0;j<filteredECG.length;j++){
        nonlinear[j]=differentResult[j]*(Math.pow(filteredECG[j],2));
    }
    return nonlinear;
}
```

Adding a high frequency sequence to the non-linear transformed ECG data is achieved by implementing the following function,

***public double[ ] hfsAddition( double[ ] ampArray, double[ ] nonlinearSignal )***

```

public double[] hfsAddition(double[] ampArray, double[] nonlinearSignal){

    double[] amplitude=new double[nonlinearSignal.length];
    double[] hfs=new double[nonlinearSignal.length];
    double[] addition=new double[nonlinearSignal.length];
    for (int i=1;i<nonlinearSignal.length;i++){

        amplitude[i]=0.5*amplitude[i-1]+((0.5*Math.abs(ampArray[i]))*4/10000);
        hfs[i]=(Math.pow(-1,i))*amplitude[i];
        addition[i]=hfs[i]+nonlinearSignal[i];
    }
    return addition;
}

```

Köhler [80] suggests using the amplitude of the nonlinear transformed signal to determine the amplitude of the high frequency sequence, because its amplitude is not uniform, it can not absolutely ensure many more zero crossings outside QRS portion. Therefore by introducing a fixed appropriate amplitude that is about 1/1000 of the max value in the stream of data, many more zero crossings are ensured outside the QRS complex compared to inside the QRS complex after adding the high frequency sequence to the nonlinear transformed signal.

The function *public double[ ] extractFeatureSignal( double[ ] additionResult )* is written to extract the feature signal that is absolute critical in the beat detection process,

```

public double[] extractFeatureSignal(double[] additionResult){
    double[] dTerm=new double[additionResult.length];
    double[] feature=new double[additionResult.length];
    double[] z=new double[additionResult.length];
    for(int i=0;i<additionResult.length;i++){
        if(additionResult[i]>0){
            z[i]=1;
        }else if(additionResult[i]==0.0){
            z[i]=0;
        }else{z[i]=-1;    }
    }
    int j;
    for( j=1;j<additionResult.length;j++){
        dTerm[j]=Math.abs((z[j]-z[j-1])/2);
        feature[j]=0.5*feature[j-1]+0.5*dTerm[j];
        System.out.println(j+": "+feature[j]);
    }
    return feature;
}

```

The change of the values for the feature signals are indicating the QRS complex, because the value of the feature signals is closer to 1 before QRS complex and it starts to vary under 1 inside the QRS complex.

The function **public double[] getThreshold(double[] featureSignal)** is to get an adaptive threshold that will be compared to the feature signal,

```

public double[] getThreshold(double[] featureSignal){
    double[] threshold=new double[featureSignal.length];
    for(int i=1;i<featureSignal.length;i++)
        threshold[i]=0.5*threshold[i-1]+0.5*featureSignal[i];
    return threshold;
}

```

When the feature signal begins to become smaller than the corresponding threshold,

the algorithm considers the QRS complex has been detected and a heart beat is found.

In order to decrease the workload for the next step, every detected beat will be assigned with the corresponding time information,

```
public long getBeat(double[] features,double[] thresholds){
    for(int i=0;i<features.length;i++){
        if (features[i]<thresholds[i]){
            int frameNumber=_dr.getFrameNumber();
            timeR=System.currentTimeMillis();
            System.out.println("FrameNumber:"+frameNumber+";"+"Beat
            Location in this Frame:"+i);
            this.setBeatFlag();
            Date date=new Date(timeR);
            String s=date.toGMTString();
            System.out.println(s);
            i=i+100;
        }
    }
    return timeR;
}
```

## **APPENDIX C TECHNICAL DESCRIPTION OF THE HARDWARE AND SOFTWARE PLATFORM HP IPAQ 5550 AND CREME KVM**

### **C.1 THE IPAQ 5550 PERSONAL DIGITAL ASSISTANT**

The PDA that is employed for this project is HP iPAQ H5550 became available on market on 30<sup>th</sup> June, 2003. A picture of the main screen of this PDA is shown below,



Figure C.1 The HP iPAQ 5550 personal digital assistant

The following table displays the technical specifications of this PDA,

MPN	FA146A#8ZP
<b>Key Features</b>	
Operating System	Microsoft Mobile Pocket PC 2003
Processor	400 MHz Intel XScale
Installed Memory	128 MB
Display	16-bit (64k colors) Transflective Color TFT
<b>Technical Features</b>	
Processor Type	Intel XScale
Processor Speed	400 MHz
Input Method	Microphone • Touch Screen • Keyboard (optional)
Interface Type	USB • Bluetooth Wireless Technology
Security Features	Biometric Fingerprint Reader • WEP Security
Expansion Slot	SDIO Slot • Secure Digital (SD) Card Slot
<b>Display</b>	
Display Tech	Transflective Color TFT
Screen Size	3.8 inch (Diagnol)
Resolution	240 x 320
Color Depth	16-bit (64k colors)
<b>Multimedia</b>	
Voice Recorder	With Voice Recorder
Audio Output	Headphone Jack • Built in Speaker
<b>Communication</b>	
Wireless	WLAN 802.11b • Bluetooth
<b>Battery</b>	
Type	Proprietary Lithium
<b>Dimensions</b>	
Width	3.3 in.
Depth	0.63 in
Height	5.43 in
Weight	7.29 oz.
<b>Warranty</b>	
Period	1 Year
<b>Miscellaneous</b>	
Cradle type	USB / Serial Cradle
Accessories	Charger Cradle • Charger Adapter • Backup Battery • AC adapter - Included
Release Date	30 June, 2003
Product ID	20815410

Table C.1 Technical Specification

As shown in the table above, 128 MB of installed memory is available for long term storage of programs and files as seen in figure C2 below. In order to make the system work faster, it is better to give the program more memory.

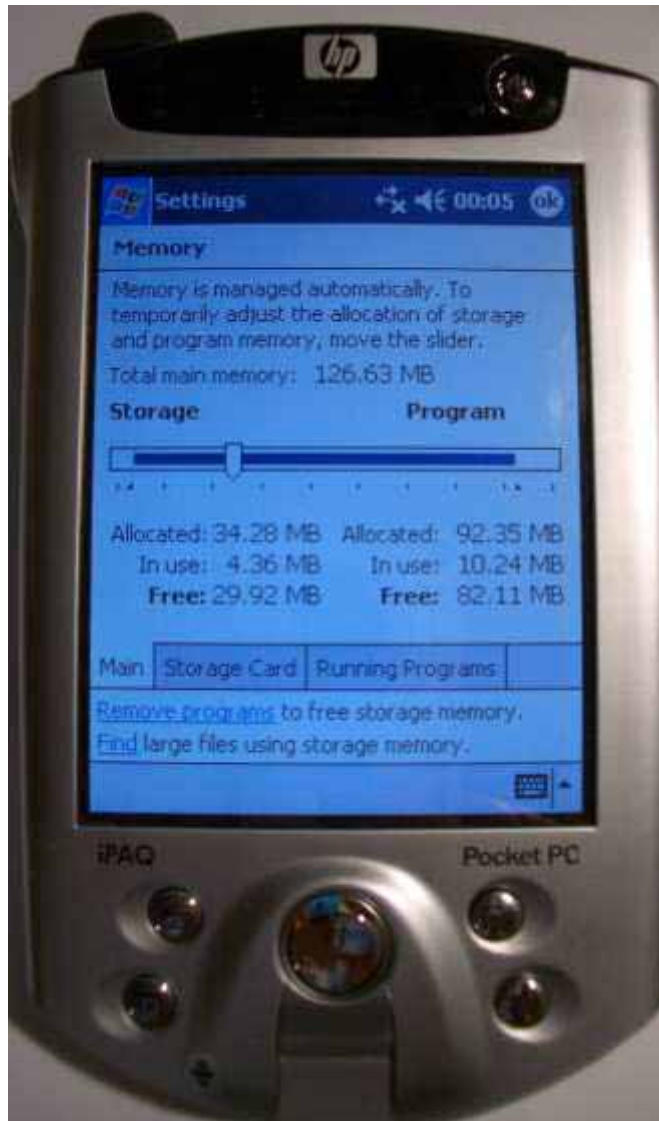


Figure C.2 Memory allocation on the iPAQ 5550

## C.2 CREME KVM

CrE-ME KVM is a small footprint Java Virtual Machine for memory limited devices from Negev Software Industries Ltd. (NSI). NSI is a software solutions provider, was established in 1992 by Digital Equipment Corporation.

Following a comparison of KVMs based on price and functionality, the CrEme Java Virtual Machine was selected for the project. One advantage for future development is

that Creme is readily available for most Windows-CE platforms, and evaluation versions are freely available for current platforms. CrEme is easy to install, and it is convenient to include in Windows CE images. The current CrEme version, Version 4.12, is compliant with J2ME/CDC PersonalProfile specification, which is based on JDK 1.3.1. The following picture shows the CrEme software suite installed on the iPAQ 5550.

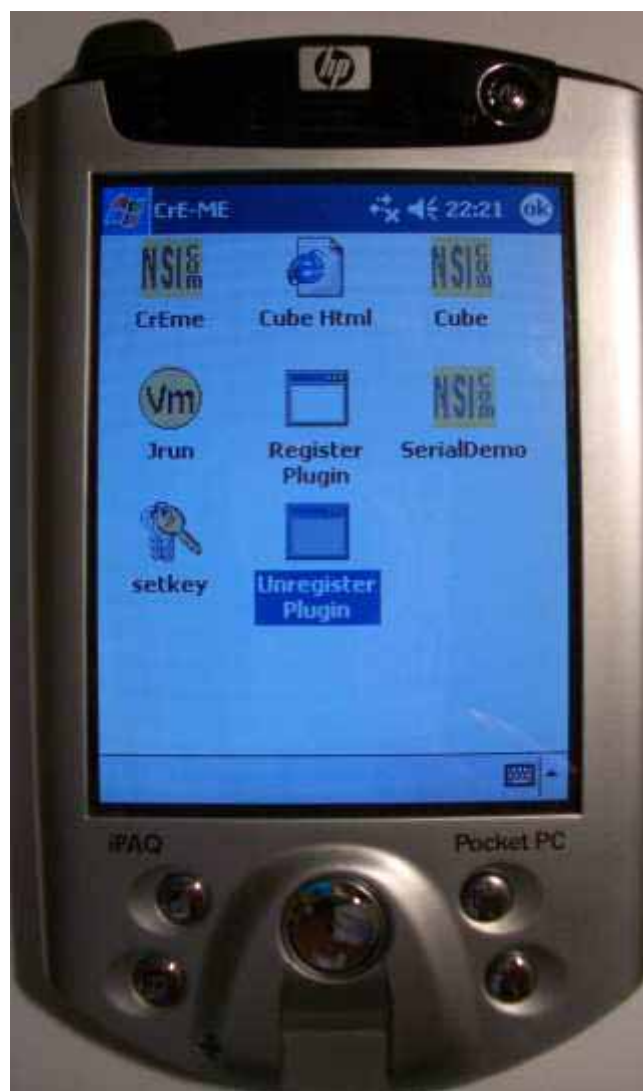


Figure C.3 Creme application menu on the iPAQ 5550

In order to test the application in the PC, CrE-ME supplies an emulator – “pjscp”, but



it can not run any application that has Javax.comm API. However other applications can run smoothly.

## APPENDIX D JAVA API

Considering the size of the whole Java API, this appendix only shows the APIs that were employed in this project.

<b><u>Package java.io.InputStream</u></b>	
<b>InputStream</b>	This abstract class is the superclass of all classes representing an input stream of bytes.
<b>Method Summary</b>	
int <b>available</b> ( )	Returns the number of bytes that can be read (or skipped over) from this input stream without blocking by the next caller of a method for this input stream.
void <b>Close</b> ( )	Closes this input stream and releases any system resources associated with the stream.
void <b>mark</b> (int readLimit)	Marks the current position in this input stream.
boolean <b>markSupported</b> ( )	Tests if this input stream supports the mark and reset methods.
abstract int <b>read</b> (byte[ ] b )	Reads some number of bytes from the input stream and stores them into the buffer array b.
int <b>read</b> (byte[ ] b, int off, int len )	Reads up to len bytes of data from the input stream into an array of bytes.
void <b>reset</b> ( )	Repositions this stream to the position at the time the mark method was last called on this input stream.
long <b>skip</b> ( long n)	Skips over and discards n bytes of data from this input stream.

<b><u>Package java.io.OutputStream</u></b>	
<b><u>OutputStream</u></b>	This abstract class is the superclass of all classes representing an output stream of bytes.
<b>Method Summary</b>	
void <b>Close</b> ( )	Closes this output stream and releases any system resources associated with this stream.
void <b>flush</b> ( )	Flushes this output stream and forces any buffered output bytes to be written out.
void <b>write</b> (byte[ ] b)	Writes b.length bytes from the specified byte array to this output stream.
void <b>write</b> (byte[ ] b, int off, int len )	Writes len bytes from the specified byte array starting at offset off to this output stream.
abstract void <b>write</b> (int b)	Writes the specified byte to this output stream.

<b><u>Package java.awt.BorderLayout</u></b>	
<b>BorderLayout ( )</b>	Constructs a new border layout with no gaps between components.
<b>Field Summary</b>	
static String <b>SOUTH</b>	The south layout constraint (bottom of container).
static String <b>WEST</b>	The west layout constraint (left side of container).
static String <b>CENTER</b>	The center layout constraint (middle of container).
static String <b>EAST</b>	The east layout constraint (right side of container).
static String <b>NORTH</b>	The north layout constraint (top of container).

<b><u>Package java.awt.Button</u></b>	
<b>Button (String label )</b>	Constructs a Button with the specified label.
<b>Method Summary</b>	
void <b>addActionListener</b> (ActionListener l)	Adds the specified action listener to receive action events from this button.
String <b>getActionCommand</b> ( )	Returns the command name of the action event fired by this button.
ActionListener[ ] <b>getActionListeners</b> ( )	Returns an array of all the action listeners registered on this button.

<b><u>Package java.awt.Frame</u></b>	
<b>Frame (String label )</b>	Constructs a new instance of Frame that is initially invisible.
<b>Method Summary</b>	
protected void <b>finalize</b> ( )	We have to remove the (hard) reference to weakThis in the Vector, otherwise the WeakReference instance will never get garbage collected.

<b><u>Package java.awt.TextField</u></b>	
<b>TextField (String text)</b>	Constructs a new text field initialized with the specified text.
<b>Method Summary</b>	
void <b>addActionListener</b> (ActionListener l)	Adds the specified action listener to receive action events from this text field.

<b><u>Package java.text.SimpleDateFormat</u></b>	
<b>SimpleDateFormat ( )</b>	Constructs a SimpleDateFormat using the default pattern and date format symbols for the default locale.
<b>Method Summary</b>	
StringBuffer <b>format</b> (Date date, StringBuffer toAppendTo, FieldPosition pos)	Formats the given Date into a date/time string and appends the result to the given StringBuffer.

<b><u>Package java.util.Arrays</u></b>	
<b>Method Summary</b>	
static void <b>sort</b> (double[] a)	Sorts the specified array of doubles into ascending numerical order.

<b><u>Package java.util.Date</u></b>	
<b>Date</b> ( )	Allocates a Date object and initializes it so that it represents the time at which it was allocated, measured to the nearest millisecond.
<b>Method Summary</b>	
long <b>getTime</b> ( )	Returns the number of milliseconds since January 1, 1970, 00:00:00 GMT represented by this Date object.

<b><u>Package java.util.EventObject</u></b>	
<b>EventObject</b> <b>Object</b> <b>source</b> )	Constructs a prototypical Event.
<b>Field Summary</b>	
<b>protected</b> <b>Object</b> <b>source</b>	The object on which the Event initially occurred.
<b>Method Summary</b>	
<b>Object</b> <b>getSource</b> ( )	The object on which the Event initially occurred.
<b>String</b> <b>toString</b> ( )	Returns a String representation of this EventObject.

<b><u>Package java.util.Vector</u></b>	
<b>EventObject</b> <b>Object</b> <b>source</b> )	Constructs a prototypical Event.
<b>Field Summary</b>	
<b>protected</b> <b>Object</b> <b>source</b>	The object on which the Event initially occurred.
<b>Method Summary</b>	
<b>Object</b> <b>getSource</b> ( )	The object on which the Event initially occurred.
<b>String</b> <b>toString</b> ( )	Returns a String representation of this EventObject.

<b><u>Package java.lang.Math</u></b>	
<b>Method Summary</b>	
static double <b>abs</b> (double a)	Returns the absolute value of a double value.
static long <b>abs</b> (double a)	Returns the closest long to the argument.
static double <b>pow</b> (double a, double b)	Returns of value of the first argument raised to the power of the second argument.

<b><u>Package java.lang.String</u></b>	
String ( )	Initializes a newly created String object so that it represents an empty character sequence.
String ( byte[ ] bytes)	Constructs a new String by decoding the specified array of bytes using the platform's default charset
<b>Method Summary</b>	
int <b>compareTo</b> (String anotherString)	Compares two strings lexicographically.
byte[ ] <b>getBytes</b> ( )	Encodes this String into a sequence of bytes using the platform's default charset, storing the result into a new byte array.
<b><u>Package java.lang.System</u></b>	
<b>Method Summary</b>	
static void <b>arraycopy</b> (Object src, int srcPos, Object dest, int destPos, int length)	Copies an array from the specified source array, beginning at the specified position, to the specified position of the destination array.
static long <b>currentTimeMillis</b> ( )	Returns the current time in milliseconds.
static void <b>exit</b> (int status)	Terminates the currently running Java Virtual Machine.
<b>Field Summary</b>	
static PrintStream <b>err</b>	The "standard" error output stream.
static InputStream <b>in</b>	The "standard" input stream.
static PrintStream <b>out</b>	The "standard" output stream.
<b>Exception Summary</b>	
InterruptedException	Thrown when a thread is waiting, sleeping, or otherwise paused for a long time and another thread interrupts it using the interrupt method in class Thread.

<b><u>Package javax.comm.SerialPort</u></b>	
<b>Method Summary</b>	
abstract void <b>addEventListener</b> (SerialPortEventListener)	Registers a SerialPortEventListener object to listen for SerialEvents.
abstract int <b>getBaudRate</b> ()	Gets the currently configured baud rate.
abstract void <b>notifyOnDataAvailable</b> (boolean)	Expresses interest in receiving notification when input data is available.

<b><u>Package javax.comm.CommPortIdentifier</u></b>	
<b>Method Summary</b>	
void <b>addPortOwnershipListener</b> (CommPortOwnershipListener)	Registers an interested application so that it can receive notification of changes in port ownership.
static CommPortIdentifier <b>getPortIdentifier</b> (CommPort)	Obtains the CommPortIdentifier object corresponding to a port that has already been opened by the application.
static Enumeration <b>getPortIdentifiers</b> ( )	Obtains an enumeration object that contains a CommPortIdentifier object for each port in the system.
<b><u>Package javax.comm.SerialPortEvent</u></b>	
<b>Field Summary</b>	
static final int <b>DATA_AVAILABLE</b>	Data available at the serial port.
<b>Method Summary</b>	
int <b>getEventType</b> ( )	Gets the type of this event
<b><u>Package javax.comm.SerialPortEventListener</u></b>	
<b>Method Summary</b>	
abstract void <b>serialEvent</b> (SerialPortEvent)	Propagates a SerialPortEvent event.
<b><u>Package javax.comm.PortInUseException</u></b>	
<b>Field Summary</b>	
String <b>currentOwner</b>	Describes the current owner of the communications port.

## **APPENDIX E MIND MAP OF CONCEPTS**

### **Data Transportation**

The structure of the data transportation is developed to increase the adaptation to the different transportation technologies. Some of the issues associated with the data transportation are discussed.

### **PDA**

The PDA is HP iPAQ H5550 and its technical specification are studied

### **ECG**

This project is to deal with QRS complex in a frame of ECG data, therefore the first objective is to understand ECG, and find out how many ECG detection algorithms available and which one is the best for this project.

The available algorithms are classified as following: a) Approaches [153,119,120,121] Based on Signal Derivatives and Digital Filters; b) Algorithms Based on Digital Filters [95,122,123]; c) Wavelet-Based QRS Detection [124,125,126]; d) Filter-Bank Methods [108,96]; e) Neural Network Approaches [127,128]; f) Hidden Markov Models [129,130]; g) Mathematical Morphology [131,132]; h) Matched Filters [133]; i) Genetic Algorithms [134]; j) Hilbert Transform-Based QRS Detection [135,136]; k) Length and Energy Transforms [137]; l) Syntactic Methods [138,139]; m) QRS

Detection Based on MAP Estimation [140]; n) Zero-Crossing-Based QRS Detection [80]. However the basic knowledge of the heart can not be ignored.

### **Java**

The Java APIs are studied and employed in this project for the system implementation.

### **Context Collection**

Collect the context information – ADLs is at the highest priority, so the design and implementation issues are detailed.

### **Evaluation**

The author needs to evaluate the value of the wireless technology and the context collection and find out how valuable of this application.



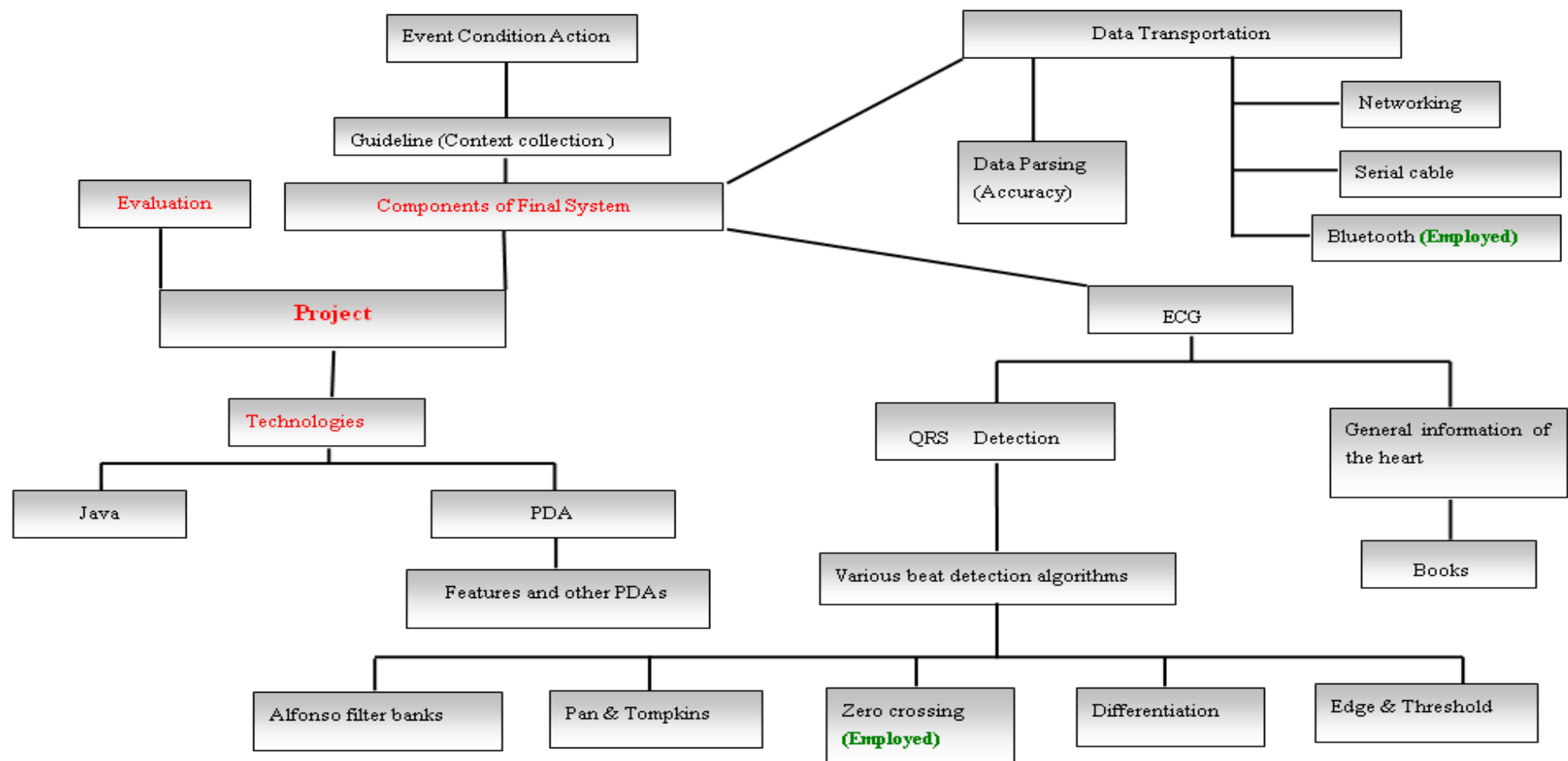


Figure E.1 Mind map (needs updating)

## **APPENDIX F OPEN DISTRIBUTED PROCESSING (ODP)**

There are five viewpoints of ODP, which are enterprise, information, computational, engineering, and technology. And their relationship is the following,

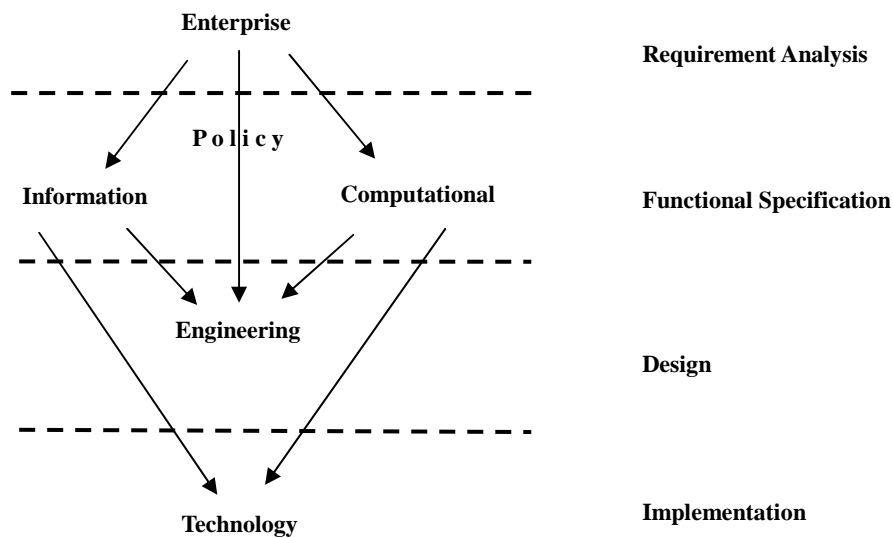


Figure F.1 ODP viewpoints [141]

Kerry Raymond [141] defines the five viewpoints as the following,

Enterprise Viewpoint = purpose, scope and policies

Information Viewpoint = semantics of information and information processing

Computational Viewpoint = functional decomposition

Engineering Viewpoint = infrastructure required to support distribution

Technology Viewpoint = choices of technology for implementation

## APPENDIX G JSR 82

Java Specification Request 82 (JSR-82), defines the optional package for Bluetooth wireless technology for Java 2 Platform, Micro Edition (J2ME). The goal of this specification is to define the architecture and the associated APIs required to enable an open, third party Bluetooth application development environment. This API is designed to operate on top of the Connected, Limited Device Configuration (CLDC), which is described in Connected, Limited Device Configuration (JSR-30), Sun Microsystems, Inc.

JSR82 has an expert group as shown below:

*Extended Systems*  
*IBM*  
*Mitsubishi Electric*  
*Motorola (specification lead)*  
*Newbury Networks*  
*Nokia*  
*Parthus Technologies*  
*Research in Motion*  
*Rococo Software*  
*Sharp Laboratories of America*  
*Sony Ericsson Mobile Communications*  
*Smart Fusion*  
*Smart Network Devices*  
*Sun Microsystems*  
*Symbian*  
*Telecordia*  
*Vaultus*  
*Zucotto*

The Bluetooth stack is shown in the following figure,

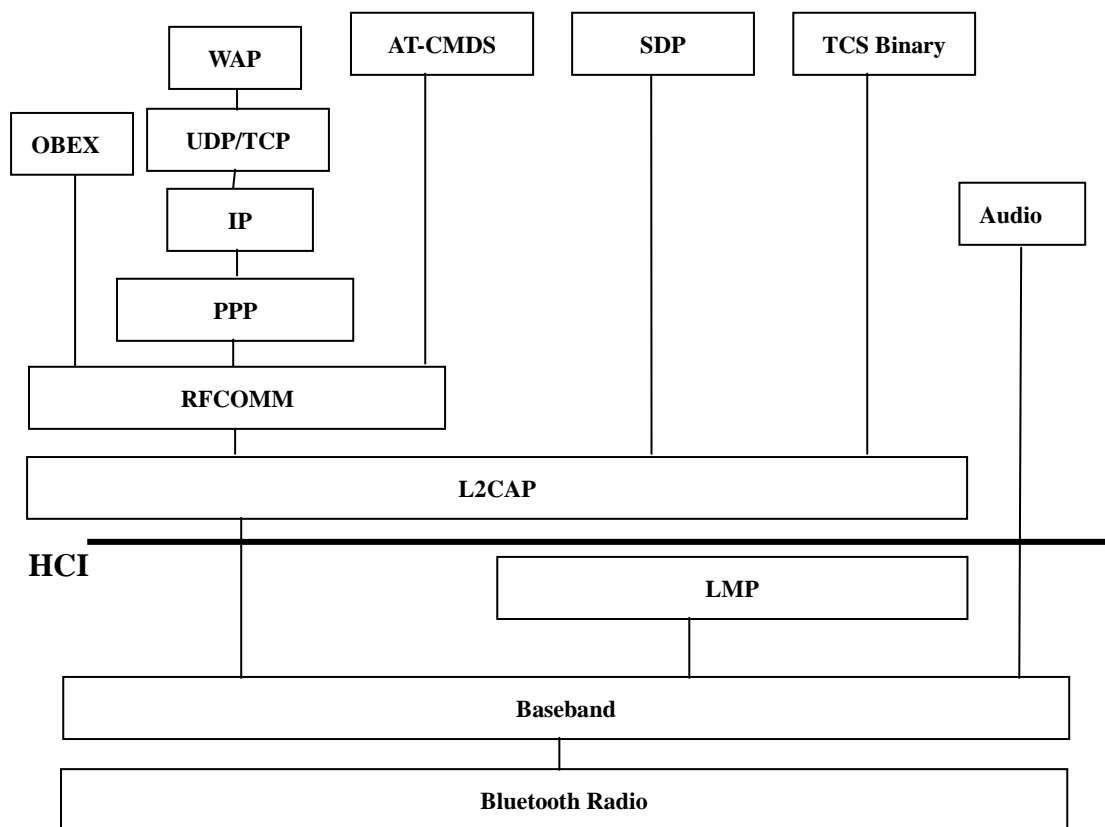


Figure G.1 The Bluetooth Protocol Stack

In addition to the protocols, the Bluetooth SIG has defined Bluetooth Profiles. A Bluetooth Profile defines standard ways to use selected protocols and protocol features that enable a particular usage model. A Bluetooth device may support one or more profiles. The four “generic” profiles are the Generic Access Profile (GAP), the Serial Port Profile (SPP), the Service Discovery Application profile (SDAP), and the Generic Object Exchange Profile (GOEP). The following figure illustrates the profiles,

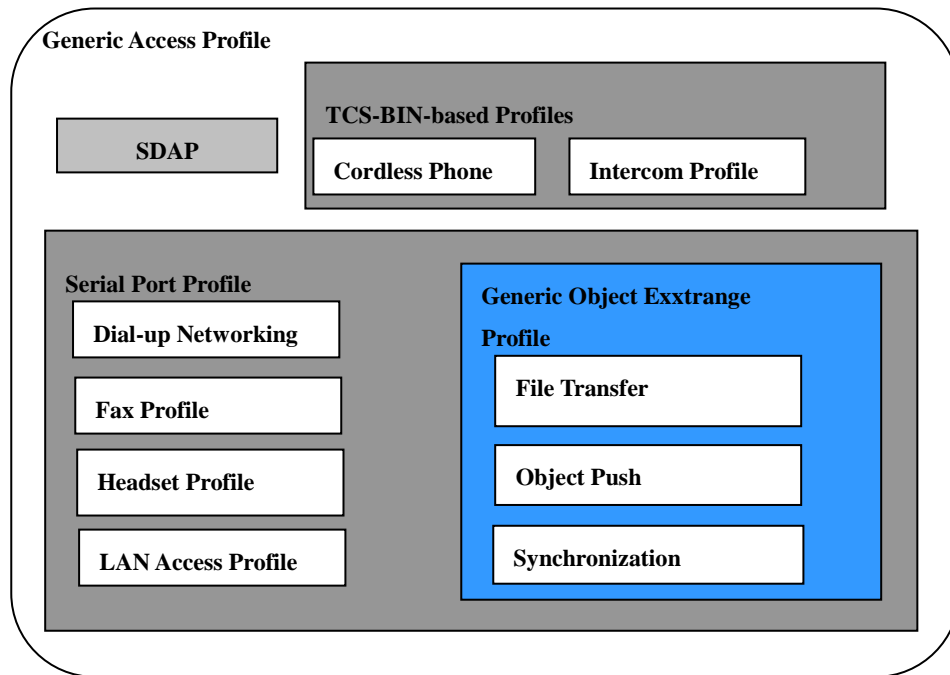


Figure G.2 Bluetooth profiles

## APPENDIX H TESTING ENVIROMENT

The system is tested in a laboratory room that is on the ground floor shown in Figure H.1.

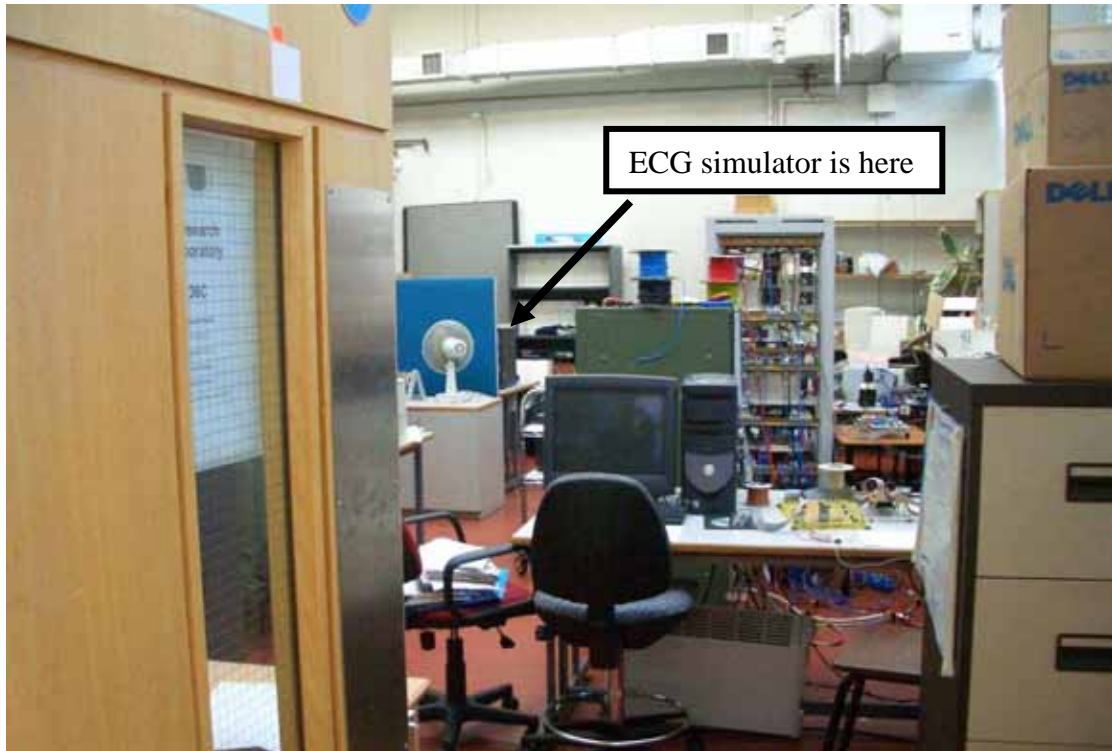


Figure H.1 Laboratory view and the ECG simulator

Outside the lab, it looks as Figure H.2.



Figure H.2 The campus

The hall is along the lab that is shown in Figure H.3.



Figure H.3 The hall along the lab

## APPENDIX I UML DIAGRAMS

### I.1 CLASS DIAGRAM

The purpose of a class diagram is to depict the classes within a model. In an object oriented application, classes have attributes (member variables), operations (member functions) and relationships with other classes. The UML class diagram can depict all these things quite easily. The fundamental element of the class diagram is an icon that represents a class. This icon is shown in Figure I.1.

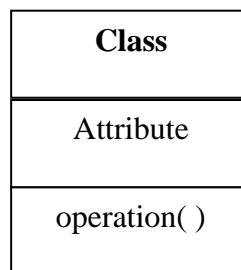


Figure I.1 The class icon

A class icon is simply a rectangle divided into three compartments. The topmost compartment contains the name of the class. The middle compartment contains a list of attributes (member variables), and the bottom compartment contains a list of operations (member functions). In many diagrams, the bottom two compartments are omitted. Even when they are present, they typically do not show every attribute and operations. The goal is to show only those attributes and operations that are useful for the particular diagram.

The inheritance relationship in UML is depicted by a peculiar triangular arrowhead. This arrowhead, that looks rather like a slice of pizza, points to the base class. One or more lines proceed from the base of the arrowhead connecting it to the derived classes, which is shown



in Figure I.2.

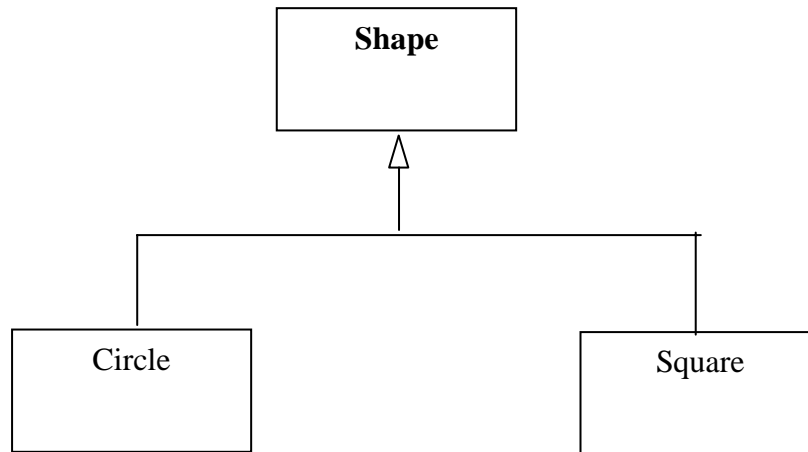


Figure I.2 Inheritance

In UML modeling, a realisation relationship is a relationship between two model elements, in which one model element (the client) realises the behavior that the other model element (the supplier) specifies. A realisation is displayed in the diagram editor as a dashed line with an unfilled arrowhead towards the supplier. Realisations can only be shown on class diagrams.

## I.2 SEQUENCE DIAGRAM

UML sequence diagrams are used to represent or model the flow of messages, events and actions between the objects or components of a system and a simple diagram is shown below,

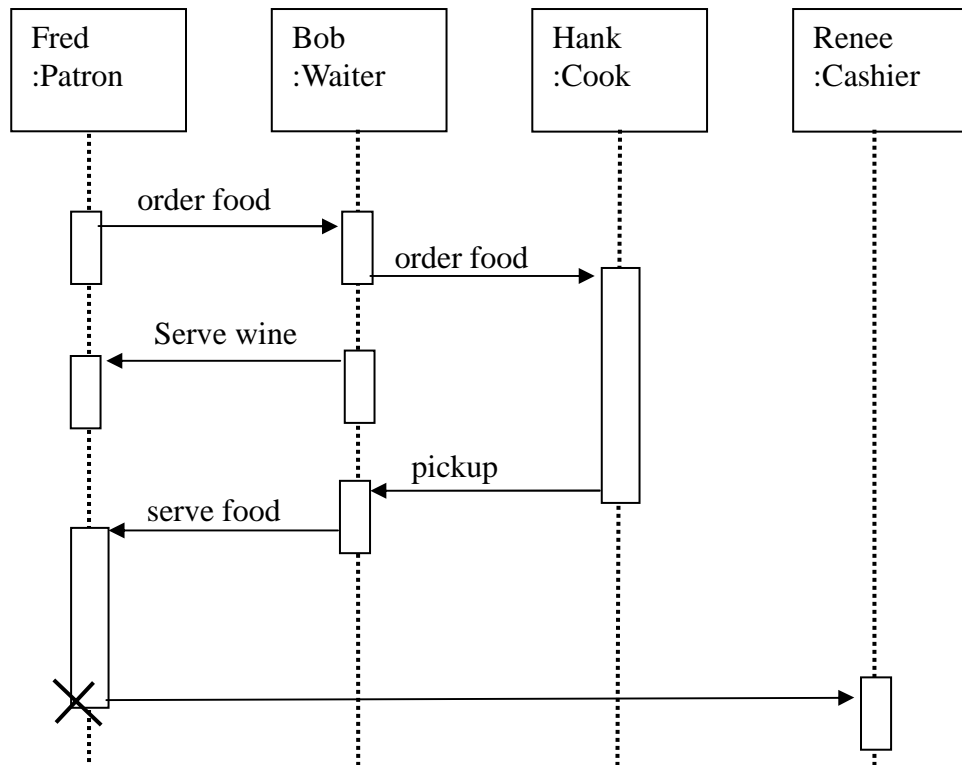


Figure I.3 Sequences diagram

## APPENDIX J BLUETOOTH ISSUSE IN HP IPAQ

HP iPAQ 5550 has well reported memory problems associated with Bluetooth manager on Windows CE and often requires a soft-boot before it can launch the winCE manager.

Figure 8.1 ‘Bluetooth manager radio failed to turn ON due to insufficient device memory...’

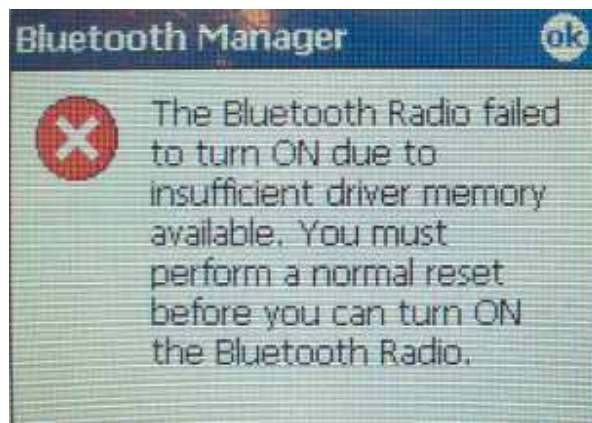


Figure J.1 Bluetooth error

One possible reason for the memory error could be that Bluetooth stack is bulky [6]. This Bluetooth issue is one that has plagued iPAQ from the beginning. And the problem is a result of a number of decisions by various groups as followed [48], 1) Microsoft decided to allocate only one block of memory for drivers. In Windows Mobile 2003, that was modified to two blocks, one for system drivers and one for application drivers. That helped a bit, but does not completely solve the problem. Bluetooth is considered a system driver; 2) HP decided not to allocate the driver memory until the Bluetooth radio was turned on. That gives user access to that memory when the radio is off, but, if another program gets the RAM first, the user gets this annoying message. Actually, HP put the

message in there to warn the user; 3) The Bluetooth consortium decided that memory for every profile that is supported in a machine will be allocated when the Bluetooth stack is loaded. That means that, even if you only use one Bluetooth profile, memory is allocated for every one supported.

As a result, if a user turns on the Bluetooth radio right after a soft-reset, he/she will always be able to use it. If user waits, or turns it off. Other drivers, or even programs, may use the memory that the Bluetooth stack needs. Since all the system drivers load into one specific block, even if user has tons of free memory, if drivers or programs have taken enough of the system driver block that the Bluetooth stack can't load, user will get the warning message and the only thing that makes it to work is to do a soft-reset [48].

## APPENDIX K FINITE IMPULSE RESPONSE FILTER

A Finite Impulse Response (FIR) filter is a type of a digital filter. The FIR filter is widely used in various DSP applications such as signal preconditioning and video convolution functions. The filter is implemented with a well known formula of digital convolution as follows,

$$y[k] = \sum_m^M x[n - m] \bullet h[m] \quad \text{Equation K.1}$$

where  $x[n]$  is the input signal and  $h[m]$  is the table of filter coefficients.  $M$  is the filter order or number of taps. The general filter form cannot be optimized in terms of number of additions and multiplications. There are many optimizations based either on assumptions on input signal, or assumptions on filter coefficients. An  $M$ -order FIR filter requires  $M$  multiplications and  $M$  additions for every output signal sample.

Equation K.1 can be implemented by a MATLAB function,

$$y = \text{conv}(x, h) \quad \text{Equation K.2}$$

In this convolution sum in Equation K.2, notice that at each  $n$  we need access to  $x(n)$ ,  $x(n - 1)$ ,  $x(n - 2)$ ...  $x(n - M)$ . We can maintain this set of values by cascading a set of latches to form a **delay line**, as shown in Figure K.1,

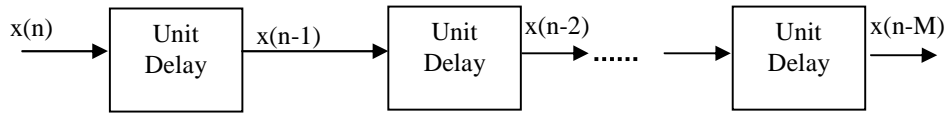


Figure J.1 Delay in a FIR filter

For each integer  $n$ , the output sample is the values in the delay line scaled by  $h(0)$ ,  $h(1) \dots h(M)$ . To obtain these values, we simply **tap** the delay line, as shown in Figure J.2,

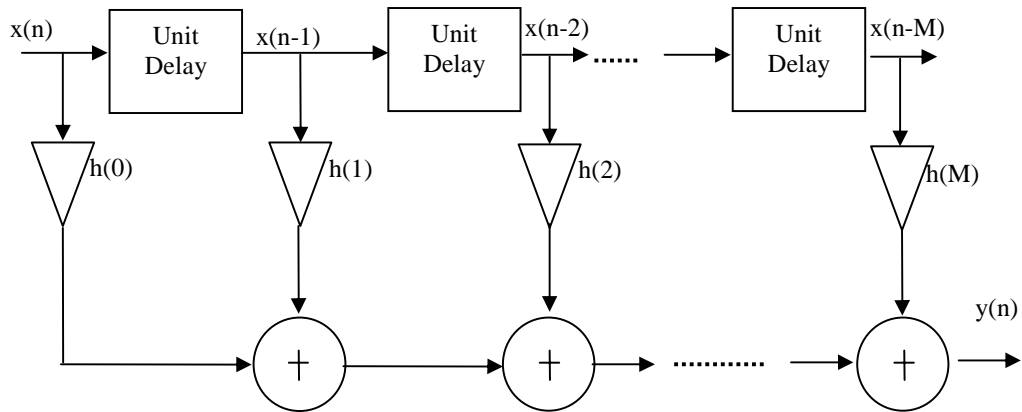


Figure J.2 FIR filter

The FIR filter illustrated in Figure J.2 can be obtained by the following generalized Java code,

```

class FIR {
    private int length;
    private double[] delayLine;
    private double[] impulseResponse;
    private int count = 0;

    FIR(double[] coefs) {
        length = coefs.length;
        impulseResponse = coefs;
        delayLine = new double[length];
    }

    double getOutputSample(double inputSample) {
        delayLine[count] = inputSample;
        double result = 0.0;
        int index = count;
        for (int i=0; i<length; i++) {
            result += impulseResponse[i] * delayLine[index--];
            if (index < 0) index = length-1;
        }
        if (++count >= length) count = 0;
        return result;
    }
}

```

# **4th European Congress for Medical and Biomedical Engineering 2008**

Li J.P., Berry D., Hayes R

School of Electrical Engineering Systems

Dublin Institute of Technology

## **ABSTRACT**

Preventative health management represents a shift from the traditional approach of reactive treatment-based healthcare towards a proactive wellness-management approach where patients are encouraged to stay healthy with expert support when they need it, at any location and any time.

This work represents a step along the road towards proactive, preventative healthcare for cardiac patients. It seeks to develop a smart mobile ECG monitoring system that requests and records context information about what is happening around the subject when an arrhythmia event occurs. Context information about the subject's activities of daily living will, it is hoped, provide an enriched data set for clinicians and so improve clinical decision making. As a first step towards a mobile cardiac wellness guideline system, the authors present a system which can receive bio-signals that are wirelessly streamed across a body area network from Bluetooth enabled electrocardiographs. The system can store signals as they arrive while also responding to significant changes in Electrocardiogram activity. The authors have developed a prototype on a handheld computer that detects and responds to changes in the calculated heart rate as detected in an ECG signal.

Although the general approach taken in this work could be applied to a wide range of bio-signals, the work focuses on ECG signals. The components of the system are,

- A Bluetooth receiver, data collection and storage module
- A real-time ECG beat detection algorithm.
- An Event-Condition-Action (E-C-A) rule base which decides when to request context information from the user.
- A simple user interface which can request additional information from the user.

A selection of real-time ECG detection algorithms were investigated for this work and one algorithm was tested in MATLAB and then implemented in Java. In order to collect ECG signals (and in principle any signals), the generalised data collection architecture has also been developed using Java and Bluetooth technology. An



Event-Condition-Action (ECA) rule based expert system evaluates the changes in heart beat interval to decide when to interact with the user to request context information.

**Keywords**

Bluetooth, ECG, beat detection, event condition action, guideline, context collection

Corresponding Author:

Mr Damon Berry

Damon.Berry@dit.ie