CHAPTER 3

ECG SIGNAL RECORDING USING LABVIEW

3.1 INTRODUCTION

The Work has been inspired by the need to find an efficient method for ECG signal recording and processing. ECG signals are non-stationary pseudo periodic in nature and whose behavior changes with time. The proper processing of ECG signal and its accurate detection is very much essential since it determines the condition of the heart. Lab VIEW based signal analysis is simple and has good accuracy and less computation time. Lab VIEW with its signal processing capabilities provides a robust and efficient environment for resolving ECG signal processing problems. Lab VIEW is one of the powerful tools in recording, denoising, analyzing, and extracting ECG signals easily and conveniently. The initial task is efficient recording of ECG signal. It actually involves the extraction of the required cardiac components by rejecting the background noise. ECG is a nearly periodic signal that reflects the activity of the heart. A lot of information on the normal and pathological physiology of heart can be obtained from ECG. However, the ECG signals being non-stationary in nature, it is very difficult to visually analyze them. Thus the need is there for Lab VIEW - based methods for ECG signal recording and Analysis. Real time monitoring plays an important role in biomedical engineering, Particularly in ECG, EMG, EEG etc. personal computer have become a standard platform for the needs of various measurement and test, standardization, performance and low cost. Use of PC in so called personal and virtual instrumentation developments enables realization of a new generation of superior devices. With their performance, this is becoming ever higher and with the increasing number of software
applications they are widely accepted as an essential tool on desk of engineer. The use of Lab VIEW and data acquisition in biomedical makes the real time monitor systems with very high performance, low cost of development, more reliable and flexible. Lab VIEW is general purpose software for virtual instrumentation in which other products like dasylab, genie, and aliment vee are followed. With Lab VIEW the maintenance and reconfiguration of created instruments are reduced significantly. PC based virtual instrumentation as a testing platform enabling recording of real time ECG introduces identification of ECG and transmission of preprocessed data to a doctor through a is tribute computation network has been proposed in [PC based monitoring system has been proposed]. In this GUI of the system has been developed in Microsoft, .NET visual C++ but it lacks the simultaneous lead illustration of ECG wave for mints selection and addition of intelligence for auto diagnose. The intelligent virtual ECG device by integrating dyadic wavelet algorithm for QRS detection, recording and identification with the facilities of the detection of heart rhythm and offline analysis of prerecorded ECG signal has been proposed. Besides all these development in biomedical engineering, the designed system in paper facilitates the automatic removal of noises and filtration of acquired signal on virtual cardiographs and this system can be used for analysis, identification of peak QRS and auto diagnose.

Lab VIEW is a platform and development environment for a visual programming language from National Instruments. The graphical language is named “G”. It is a graphical programming language with three important components data acquisition, data analysis and data visualization. This programming paradigm has been widely adopted by industry, academia and research laboratories over the world as the standard for data acquisition and instrument control software. The functions in Lab VIEW are called Virtual Instruments (VIs). VIs performs small tasks such as mathematical functions recording, configuration etc. There is a function palette, which lists all the
functions available with categories and sub categories. These VIs have inputs and outputs, which gives the user the necessary information needed in the selection of the parameters. VIs in a block diagram manner simplifies the use of classical digital signal processing. Signal processing is required to extract the useful information from the original signal by removing noise and other non-related frequencies. There are two basic components in Lab VIEW: the front panel screen and the block diagram screen. On the front panel, the running of the program and its performance can be viewed, but the actual programming takes place on the block diagram screen. The front panel is usually regarded as the user interface through which the user can visualize the performance of the program. On the block diagram screen, different VIs are selected and connected to carry out functions such as reading the data, filtering the data, storing the data etc. First, the data collected from the amplifier were transmitted to the computer for the configuration. Then, the data were passed through four filters and windows and the FFT spectrum was plotted for each band and the data were stored in the computer. A detailed description of each step is discussed below. The first few steps of the program set the parameters for proper communication between the software and the hardware. The VI sets up the channel specification and the buffer size, three channels were used for this system. The first channel was for air puff delivery, the second for tone delivery and the third for recording of the ECG. The Buffer is the storage site in RAM where the data can be continuously acquired and retrieved. Two types of problems might occur where the buffer is involved in data acquisition. The 16 data might be retrieved slower from the buffer than it is placed into it as a result, Lab VIEW overwrites the data. Therefore the size of the buffer should be decided very carefully to avoid any such dilemmas. In this particular system, a sampling rate of 1024 samples/sec was selected. Since the sampling rate was very high, the buffer was set to 100,000 to provide enough storage of the data. The first few steps of the
program set the parameters for proper communication between the software and the hardware.

3.2 DATAFLOW PROGRAMMING

The programming language used in Lab VIEW, called “G”, is a dataflow language. Execution is determined by the structure of a graphical block diagram (the LV-source code) on which the programmer connects different function-nodes by drawing wires. These wires propagate variables and any node can execute as soon as all its input data become available. Since this might be the case for multiple nodes simultaneously, G is inherently capable of parallel execution. Multi-processing and multi-threading hardware is automatically exploited by the built-in scheduler, which multiplexes multiple OS threads over the nodes ready for execution. Programmers with a background in conventional programming often show a certain reluctance to adopt the Lab VIEW dataflow scheme, claiming that Lab VIEW is prone to race conditions. In reality, this stems from a misunderstanding of the data-flow paradigm. The aforesaid data-flow (which can be “forced”, typically by linking inputs and outputs of nodes) completely defines the execution sequence, and that can be fully controlled by the programmer. Thus, the execution sequence of the Lab VIEW graphical syntax is as well defined as with any textually coded language such as C, Visual BASIC, and Python etc. Furthermore, Lab VIEW does not require type definition of the variables; the wire type is defined by the data-supplying node. Lab VIEW supports polymorphism in that wires automatically adjust to various types of data.
3.3 **GRAPHICAL PROGRAMMING**

Lab VIEW ties the creation of user interfaces (called front panels) into the development cycle. Lab VIEW programs/subroutines are called Virtual Instruments (VIs). Each VI has three components: a block diagram, a front panel and a connector panel. The latter may represent the VI as a sub VI in block diagrams of calling VIs. Controls and indicators on the front panel allows an operator to input data into or extract data from a running virtual instrument. However, the front panel can also serve as a programmatic interface. Thus a virtual instrument can either be run as a program, with the front panel serving as a user interface, or when dropped as a node onto the block diagram, the front panel defines the inputs and outputs for the given node through the connector pane. This implies each VI can be easily tested before being embedded as a subroutine into a larger program. The graphical approach also allows non-programmers to build programs by simply dragging and dropping virtual representations of the lab equipment with which they are already familiar. The Lab VIEW programming environment, with the included examples and the documentation, makes it simpler to create small applications. This is a benefit on one side but there is also a certain danger of underestimating the expertise needed for good quality “G” programming. For complex algorithms or large-scale code it is important that the programmer possess an extensive knowledge of the special Lab VIEW syntax and the topology of its memory management. The most advanced Lab VIEW development systems offer the possibility of building stand-alone applications. Furthermore, it is possible to create distributed applications which communicate by a client/server scheme, and thus is easier to implement due to the inherently parallel nature of G-code.
3.4 BENEFITS

One benefit of Lab VIEW over other development environments is the extensive support for accessing instrumentation hardware, drivers and abstraction layers for many different types of instruments and buses are included or are available for inclusion. These present themselves as graphical nodes. The proposal layers offer standard software interfaces to communicate with hardware devices. The provided driver interfaces save program development time. The sales pitch of National Instruments is, therefore, that even people with limited coding experience can write programs and deploy test solutions in a reduced time frame when compared to more conventional or competing systems. A new hardware driver topology (DAQmx Base), which consists mainly of G-coded components with only a few register calls through NI Measurement Hardware DDK (Driver Development Kit) functions, provides platform independent hardware access to numerous data acquisition and instrumentation devices. Lab VIEW includes a compiler that produces native code for the CPU platform. The graphical code is translated into executable machine code by interpreting the syntax and by compilation. The Lab VIEW syntax is strictly enforced during the editing process and compiled into the executable machine code when requested to run or upon saving. In the latter case, the executable and the source code are merged into a single file. The executable runs with the help of the Lab VIEW run-time engine, which contains some precompiled code to perform common tasks that are defined by the G language. The run-time engine reduces compile time and also provides a consistent interface to various operating systems, graphic systems, hardware components, etc. The run-time environment makes the code portable across platforms. Generally, LV code can be slower than equivalent compiled C code, although the differences often lie more with program optimization than inherent execution speed. Many libraries with a large number of functions for data acquisition, signal generation,
mathematics, statistics, signal conditioning, analysis, etc. along with numerous graphical interface elements are provided in several Lab VIEW package options.

3.5 CRITICISM

Lab VIEW is a product of National Instruments. Unlike common programming languages such as C or FORTRAN, Lab VIEW is not managed or specified by a third party standards committee such as ANSI. Obtaining a fully compatible and up to date Lab VIEW platform requires purchasing the product. There is a movement to create user-defined extensions for the development environment at Open G.org but an initial purchase of Lab VIEW is still required. Currently there is no open source, free software or alternative commercial program that can implement any portion of G-code. In addition, as of version 8, all Lab VIEW installs require customers to contact National Instruments by Internet or phone to “activate” the product. The current system, the user would no longer be able to access their code base as well as certain formats of archived data. Building a stand-alone application with Lab VIEW requires the Application Builder component which is included with the Professional Development System but requires a separate purchase if using the Base Package or Full Development System. Although this run-time engine can be freely downloaded from National Instruments website, this added requirement is in contrast to other compiled languages, such as C, where a stand-alone executable file can be created. The need for a separately installed Lab VIEW run-time engine makes the development and distribution of truly portable applications using Lab VIEW difficult. The Call by Reference Node is used to call VIs with a specified connector pane, but without to exactly specify the VI at edit time. It can be used to provide a plug-in architecture within an application, where the actual plug-in can be chosen at runtime.
3.6 HARDWARE AND SOFTWARE DESCRIPTION

3.6.1 Description of the Biokit

The Biokit Physiograph System is meant for capturing and analyzing Biomedical Signals like ECG, PCG, EEG, EMG, and Pulse etc. The captured signals can be analyzed using the Biokit Physiograph Software. The Physiograph system can be of two flavors, Built in Amplifier and Non-Amplifier Systems. The Built in Amplifier systems will have built in amplifiers for ECG, PCG, EEG and EMG etc. The System can be categorized into, Data Acquisition Unit, Amplification Unit and opto-Isolation Unit. The Amplification Unit amplifies the Bio Signals fed from external systems like Electrodes or Sensors. The Data Acquisition Unit acquires the amplified data and converts into a digital format, which can be input to the Personal Computer through a Serial Port. The Opto-isolation Unit optically isolates the Biokit Physiograph System, from the mains power, so that no harm is caused to the experimenter or the subject.

3.6.2 Sub System Layout

The amplifier unit appropriately amplifies the bio signals grabbed from the subject with the help of Electrodes and Sensors. The Date Acquisition Unit acquires the amplified data converts into a digital format. The opto- isolation Unit optically isolates the Biokit Physiograph System, from the mains power especially from the PC, so that human safety is achieved. The part of the biomedical signal related to a specific event of interest is called as an epoch. Analysis of a signal for monitoring or diagnosis requires the identification of epochs and investigation of the corresponding events. Detailed analysis of ECG waveforms will require the use of several features for accurate categorization of various cardiovascular diseases.
3.6.3 Hardware

3.6.3.1 Data acquisition unit

- Microcontroller Based
- 1000 Samples / Second / Channel
- Channels
- Baud rate of 115200 Bits/Second
- Optical isolation
- Serial Port/RS232 Connectivity
- Advanced Filters
- Battery Powered.
3.6.3.2  ECG amplifier

- Wire patient cable
- Input impedance > 5 M ohm
- CMRR > 80 Db
- Gain = x 1K Max
- Frequency Response – 1Hz to 48 Hz.

3.6.3.3  Power supply

- 12 V DC Battery operated with built in charger
- Battery Capacity – 12 V, 2.5 AH
- Charging Current < 50 Ma
- Charging Input – 230 V A.C transformer coupled
- Fuse – 500 mA slow blow
- In Use, Charging and Low Battery indicator.

3.6.3.4  Data acquisition unit

Data acquisition is the sampling of the real world to generate data that can be manipulated by a computer. Sometimes abbreviated DAQ or DAS, data acquisition typically involves acquisition of signals and waveforms and processing the signals to obtain desired information. The components of data acquisition systems include appropriate sensors that convert any measurement parameter to an electrical signal, then conditioning the electrical signal which can then be acquired by data acquisition hardware. Acquired data are displayed, analyzed, and stored on a computer, either using vendor supplied software, or custom displays and control can be developed using various
general purpose programming languages such as BASIC, C, FORTRAN, Java, Lisp, Pascal. Specialized programming languages used for data acquisition include EPICS, used to build large scale data acquisition systems, Lab VIEW, which offers a graphical programming environment optimized for data acquisition, and MATLAB which provides a programming language, and also built-in graphical tools and libraries for data acquisition and analysis.

### 3.6.3.5 NI-DAQ BNC2120

Since DAQ device acquire electrical signals, a transducer or a sensor must convert some physical phenomenon into an electrical signal. A DAQ can also simultaneously produce electrical signals. These signals can either intelligently control mechanical systems or provide a stimulus so that the DAQ can measure a response. Most DAQ devices have four standard elements: analog input (AI), analog output (AO), digital I/O (DIO), and counter/timers. The 6014E features 16 channels (eight differentials) or analog input, two channels of analog output, 1 and 8 lines of digital I/O. These devices use the National Instruments DAQ-STC system-timing controller for time-related functions. The DAQ-STC consists of three timing groups that control analog input, analog output, and general-purpose counter/timer functions. These groups include a total of seven 24-bit and three 16-bit counters and a maximum timing resolution of 50ns. The DAQ-STC makes possible such applications as buffered pulse generation, equivalent time sampling and seamless changing of the sampling rate. PC based data acquisition gives users the flexibility to develop measurement solutions for virtually any applications.

### 3.7 SOFTWARE DESCRIPTION

NI Lab VIEW is the graphical development environment for creating flexible and scalable test, measurement, and control applications rapidly and at a minimal cost. With Lab VIEW, engineers and scientists
interface with real world signals, analyze data for meaningful information, and share results and applications. Regardless of experience, Lab VIEW makes development fast and easy for all users. Lab VIEW programs are called virtual instruments, or VI, because their appearance and operation imitate physical instruments, such as oscilloscopes and multi meters. Lab VIEW contains a comprehensive set of tools for acquiring, analyzing, displaying and storing data as well as tools for troubleshooting.

3.8 CREATING THE FRONT PANEL

To create the user interface for a VI, we place the controls and data displays for our measurement system on the front panel by choosing objects from the controls palette, such as numeric displays, knobs, meters, gauges, thermometers, tanks, LEDs, charts and graphs. Then, we control our system at runtime by simply operating the various objects on the front panel, whether it is moving a slide, zooming in on a graph, a value from the keyboard.

3.9 CONSTRUCTION OF BLOCK DIAGRAM

We can construct a block diagram to define the behavior of a VI without worrying about the many syntactical details of conventional programming. We can select objects or icons from the Functions palette and connect them with virtual wires to pass data from one block to the next. These blocks range from simple arithmetic functions to advanced acquisition and analysis routines, to network and file I/O operations.

3.10 GRAPHICAL COMPILER

In many applications, execution speed is a critical consideration. Lab VIEW is the only graphical programming system with a compiler that generates optimized code with execution speeds comparable to compiled C
program. To further improve performance, you can analyze and optimize time-critical sections of code with the built-in profiler. In this way, you can increase productivity with graphical programming without sacrificing execution speed.

3.11 LABVIEW COMPETITIVE ADVANTAGE

LabVIEW users report significant productivity gains when compared to traditional development tools.

- Preserve capital investment in computer and instrumentation hardware.
- Empowers a larger group of users to develop their own solutions.
- Completes the entire application without the addition of more complicated development tools.
- Simplifies complicated development tasks with powerful add-on tools for tasks such as data analysis and visualization, report generation, and corporate database connectivity.
- Ensures successful development through National Instruments support services and a huge user network.

3.12 EXPERIMENTAL SETUP

The LabVIEW setup for visualizing the recoded ECG signal is shown in Figure 3.3.
Figure 3.2 Experimental setup using Lab VIEW

The above Lab VIEW setup can be created using the following steps

1. Go to block diagram, then function→Express→Input→Read Measurement File.
2. Go to front panel controls →express → graph indicator → chart.
3. Select wire tool then make connection between read from measurement file signals block and waveform chart.
4. Go to functions → Exec control → select → while loop.

3.13 ACQUIRING RAW ECG SIGNALS

The ring electrodes are properly placed to the different subjects and the ECG signals are acquired by various kinds of electrocardiographs. The sampling rate is typically set to 125 Hz or 250 Hz. The acquired ECG signals
can be stored in NI TDMS file type for offline analysis. The HRV Analysis Startup Kit includes a wizard to import ECG data.

Figure 3.3 Non smoker

Figure 3.4 Smoker
3.13.1 Extracting RR intervals from ECG Signals

The accurate measurement of beat to beat intervals is essential for subsequent heart rate variability analysis. Where the ECG is used to derive the intervals, timing can be affected by artifacts such as muscle noise, electrode instability and also shape changes in the QRS complex. Identifying the QRS time by correlation methods can minimize the uncertainty but the method is computationally intensive. We have developed a real time RR interval measurement system using a correlation technique. Timing resolution is ± 1ms. The correlation process uses an averaged complex from the actual ECG and has an adaptive noise threshold. The high processing speed of a DSP has proved ideal for accurate RR interval measurement. The system is described and test results with various signals to noise ratios and different types of noise are presented. RR interval signals are extracted from raw ECG signals. The extraction process usually involves the preprocessing step and the peak detection step. It is necessary to preprocess the raw ECG signals if they have noise corruption and have significant baseline trend. Then we can detect the R peaks by thresholding.

Figure 3.5 Extracted RR interval from ECG signals
The peak with the highest amplitude is called the R wave. An RR interval is the time that elapses between two successive R waves. The lower peaks are the P wave, the T wave, and the U wave, respectively. The R-R interval is also known as inter-beat-interval. The length of the cardiac cycle is termed as heart period. It can be determined by measuring the specific component of the ECG waveform. R-wave is the most common point of the cardiac cycle used. The peak of the R-wave is normally greater in amplitude than all other peaks in the ECG making it easily distinguishable. Thus R-R interval is often defined as the duration between successive R-waves. Most of the statistical and geometric measures are performed using R-R intervals.