

# DESIGN AND DEVELOPMENT OF MICROCONTROLLER BASED ECG SIMULATOR

Khandker Raska Urzoshi, ID: 09221106

Adnan Arsalan, ID: 09221178

Robi Shankar Datta ID: 09221202

Andre Dominic Paul ID: 09221183

**Department of Electrical and Electronics Engineering**


December 2010



**BRAC University, Dhaka, Bangladesh**

## Declaration

We hereby declare that this thesis is based on the results found by ourselves. Materials of work found by other researchers are mentioned by reference. This thesis, neither in whole nor in part, has been previously submitted for any degree.




08.12.10

Robi Shankar Datta

08.12.10

Andee 08-12-10

  
8.12.10

Signature of the Author

  
8/12/10  
Signature of the Supervisor

## **Acknowledgement**

With deep sense of gratitude we express our sincere thanks to our supervisor,[] Dr. A.K.M. Azad, Associate Professor, BRAC University for his valuable guidance in carrying out this work under his effective supervision, encouragement and cooperation.

We express our thanks to our fellow students Asifur Rahman and Jonayet Hossain for their suggestions, ideas and help.

We are also thankful to all the staff members of the department for their full cooperation.

## **Abstract**

For the training of doctors as well as for design, development and testing of automatic ECG(Electrocardiogram) machines, a subject with a known abnormality of heart is essentially required. ECG simulator is an electronic device used to simulate such subject for the above mentioned purpose<sup>[18]</sup>. The significance of the ECG simulator is that the subject has been replaced. The simulator is a useful tool for electrocardiograph calibration and monitoring, to incorporate in educational tasks and clinical environments for early detection of faulty behavior. The device is based on a microcontroller and generates the basic ECG wave with variable beats per minute (BPM) and arrhythmia waves. These signals are fed as an input to the ECG machine. The signals can be used for testing, servicing, calibration, and development of the ECG monitoring instruments.

## Contents

	Page No.
Declaration	ii
Acknowledgement	iii
Abstract	iv
Contents	v
Chapter 1: Introduction	7
Chapter 2: The Heart and ECG Basics	8
2.1 Introduction to ECG	8
2.2 The Heart	8
2.3 The Heart's Electrical System	9
2.3.1 The Heart Beat	9
2.3.2 Depolarization and Repolarization	10
2.4 The Basic ECG wave	11
2.5 The Duration Complexes and Interval	12
2.5.1 The P wave	13
2.5.2 The QRS Complex	13
2.5.3 The T Wave	13
2.5.4 The U Wave	13
2.5.5 PR Interval	13
2.5.6 QT Interval	14
2.5.7 ST Interval	14

2.6	The Cardiac Arrhythmias	14
2.6.1	Sinus Tachycardia	14
2.6.2	Sinus Bradycardia	15
Chapter 3: Project Overview		16
3.1	Hardware	16
3.1.1	Microcontroller (ATMega32)	16
3.1.2	Filter Network	17
3.1.3	Components and Specifications	18
3.2	Software	19
3.2.1	The Lookup Table	19
3.2.2	The AVR Studio 4 Compiler	20
3.2.3	Programming the Microcontroller	21
Chapter 4: Experimental Results		29
4.1	Generation of the basic waves	29
4.2	Generation of the basic ECG wave	30
4.3	Variable BPM option	31
4.4	Generation cardiac arrhythmias waves	32
4.5	Final Results	32
4.6	BPM Calculation	34
Chapter 5: Operation Instructions of the ECG Simulator		35
Chapter 6: Cost Analysis		38
Chapter 7: Conclusion and Future Work		40
<b>Appendix</b>		
<b>Bibliography</b>		

## CHAPTER 1

### Introduction

Electrocardiogram (ECG) is a graphic recording of the electrical potentials rhythmically produced by the heart muscle. Physicians record the ECG very easily and non invasively by placing electrodes in different places of the body<sup>[23]</sup>. ECG is used for clinical diagnosis and monitoring the heart for abnormalities.

A signal resembling the actual ECG is required to develop, and service ECG equipment. So tests are made on humans. This is very unethical. The ECG simulator is a device that can be used to test the equipment instead. It removes the potential risk to the test subject. The simulator described here produces a suitable artificial signal. This signal is fed into the ECG machine which is to be tested.

The industrial simulator is rather expensive and is not manufactured in Bangladesh. The purpose of the thesis is to design a simulator which uses a microcontroller chip and a network of capacitors, resistors and inductors to generate the signals, which will be less expensive and easily available. Our objective was to make the simulator generate the basic ECG wave with a variable beats per minute (BPM), and the two arrhythmias waves.

The following chapters give a detailed view of the entire project. Chapter 2 describes the human heart, how it works and a detailed discussion about the ECG wave, the intervals and duration of the signal, and different types of arrhythmias. Chapter 3 deals with the project overview, here we will see the block diagram and the descriptions of the hardware components used in the device, descriptions of the microcontroller, and the role of the filter network are included, and also the programming and the compilers used. We have described the experimental results we found in chapter 4, detailed description of the progress made and the pictures of the waves shapes generated are included. In Chapter 5 we will see the operation instructions of the ECG simulator we have developed. Chapter 6 gives a brief idea about the cost of our simulator compared to commercial simulators. Chapter 7 states the conclusion and future work. And finally the appendix and bibliography is stated.

## CHAPTER 2

### THE HEART AND ECG BASICS

#### 2.1 Introduction to ECG

The most functional indicator of cardiac activity is the electrocardiogram (ECG) [18]. The ECG is used around the world as a simple non invasive way for diagnosing heart conditions. The rhythmic behavior of the heart can be monitored and used as a diagnostic tool to detect heart abnormalities by acquiring the electrical activity on the body surface, across the heart. This electrical activity originates from the electrical activation of muscles in the heart, causing mechanical motion. Adhesive electrodes applied to the chest and limbs connect to the electrograph machine that detects patterns of minute electric currents in the heart muscle and print it on a chart or display it on a screen. The test does not hurt nor does it have any side effects. It does not require any preparation. The recording takes a few seconds only.

An ECG can be used to assess if the patient has had a heart attack or evidence of a previous heart attack. The doctor can determine the nature of an erratic heartbeat.

#### 2.2 The Heart

The heart is a large efficient pump which helps circulating blood throughout the body; blood enters the heart at low pressure and leaves at a higher pressure<sup>[6]</sup>. It is this high arterial pressure that provides the energy to force blood through the circulatory system. Blood returning from the body is sent to the right side of the heart and then to the lungs to pick up oxygen and release carbon dioxide<sup>[5]</sup>. The four chambers of the heart function as a double pump. The right atrium and ventricle act as one pump to propel the venous blood to the lungs for oxygenation via the pulmonary artery. At the same time, the left atrium and ventricle force the oxygenated blood into the systemic circulation via the aorta. These two pumps operate simultaneously, although they are completely separate from each other<sup>[7]</sup>.

The heart beat is triggered by electrical impulses that travel down a pathway through the heart called the heart's electrical system. It is the power source that makes this beating possible<sup>[7]</sup>.



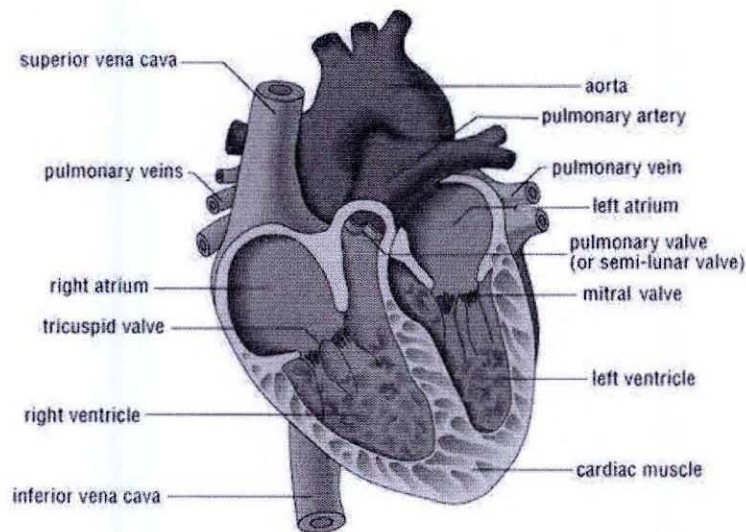


Fig 2.1 The human heart and its chambers

## 2.3 The Heart's Electrical System

The heart's electrical system is divided into three parts the sinoatrial node (SA node) the atrioventricle node (AV node) and the HIS purkinje system<sup>[7]</sup>. The pathway of the heart's electrical system is shown in figure 2.2. The SA node is located at the top right chamber or atrium. It is the natural pacemaker of the heart. The SA node initiates the heart beat by generating electrical impulses. The AV node is the bridge between the atria and the ventricles the electrical signals pass from the atria to the ventricles through this node. And the His-Purkinje fibers carry the signal throughout the ventricles.

### 2.3.1 The heart beat

A heartbeat is a complex series of events that take place in the heart. It is a single cycle in which the heart's chambers relax and contract to pump blood. This cycle includes the opening and closing of the inlet and outlet valves of the right and left ventricles of the heart. The electrical signal is created in SA node<sup>[7]</sup>. Wherever the signal is present contractions take place. The signal is generated as the two vena cava fill the heart's right atrium with blood from other parts of the body. The signal spreads across the cells of the heart's right and left atria. This signal causes the atria to contract. This action pushes blood through the open valves from the atria into both ventricles. The signal arrives at the AV node near the ventricles. It slows for an instant to allow the

heart's right and left ventricles to fill with blood. The signal is released and moves along a pathway called the bundle of His. From the bundle of His, the signal fibers divide into left and right bundle branches through the Purkinje fibers that connect directly to the cells in the walls of the heart's left and right ventricles. The signal spreads across the cells of the ventricle walls, and both ventricles contract. However, this doesn't happen at exactly the same moment<sup>[7]</sup>. The left ventricle contracts an instant before the right ventricle. This pushes blood through the pulmonary valve (for the right ventricle) to the lungs, and through the aortic valve (for the left ventricle) to the rest of the body. As the signal passes, the walls of the ventricles relax and await the next signal. This process continues over and over as the atria refill with blood and other electrical signals come from the SA node<sup>[9]</sup>.

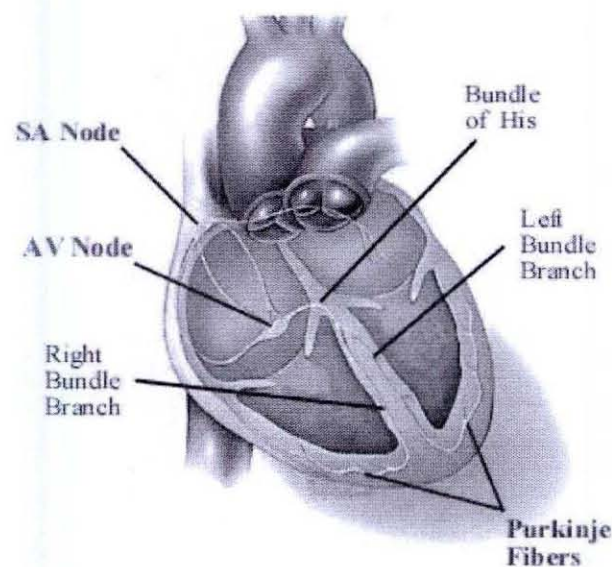


Fig 2.2 The hearts electrical system

### 2.3.2 Depolarization and repolarization

A resting heart is polarized; there is a balance of charge in and out of each cell within the heart muscle, and no electricity flows. A cell at rest has a negative charge. As a stimulus occurs, positive ions enter the cell, changing the charge to positive. This is depolarization and occurs in every cell of the heart muscle, causing the fibers to shorten. The shortening of the heart muscle fibers causes the heart muscle to

contract. The positive ions are pumped out of the cell, returning it to its normal shape, thus relaxing the heart muscle. This return of the cells to a polarized or resting state is known as repolarization. In short depolarization is the contraction whenever the impulse goes through cardiac muscle, and repolarization is the relaxation of the muscles as the impulse moves away<sup>[18]</sup>.

The chemical make-up of the body allows the potential change, which occurs during depolarization and repolarization, to be transmitted and measured at the skin surface. The ECG is able to read this electrical activity which is displayed as waveforms; P, Q, R, S, T and U<sup>[18]</sup>.

## 2.4 The Basic ECG Wave

The electric potentials generated by the heart appear throughout the body and can be measured across its surface. The typical ECG waveform is shown in figure 2.3. The signal is characterized by five peaks and valleys labeled with the successive letters PQRST and U<sup>[18]</sup>. In figure 2.4 we can see the different wave shapes obtained from different parts of the heart. By adding all these waves by Fourier series we will get the basic ECG wave shown in figure 2.3

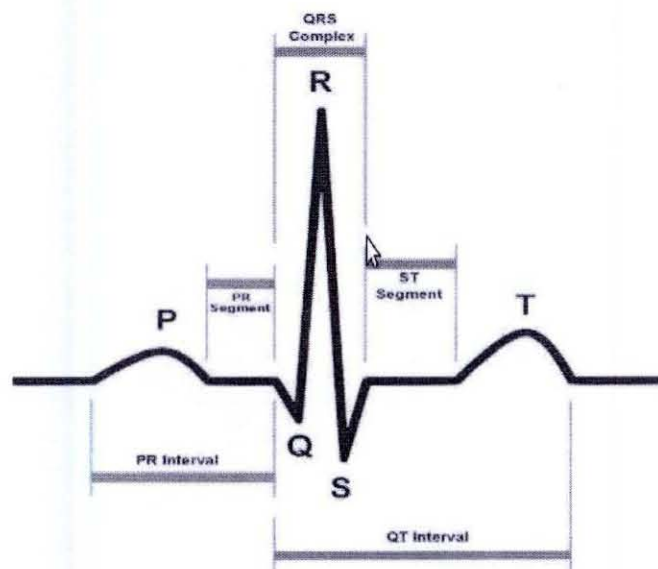


Fig 2.3 The basic ECG wave<sup>[19]</sup>

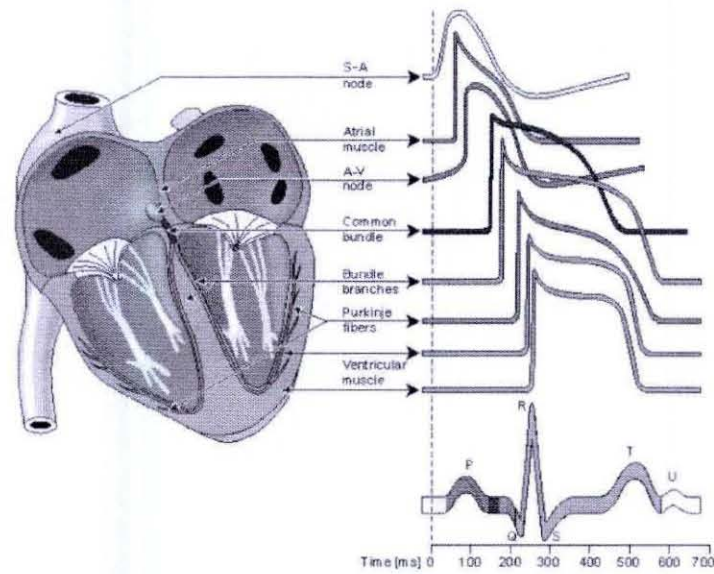


Fig 2.4 The waves generated from different points of the heart<sup>[19]</sup>

## 2.5 The Duration, Complexes and Interval

The electrocardiogram is composed of waves and complexes<sup>[8]</sup> in the normal sinus rhythm are the P wave, PR Interval, PR Segment, QRS Complex, ST Segment, QT Interval and T wave the intervals are shown in figure 2.5.

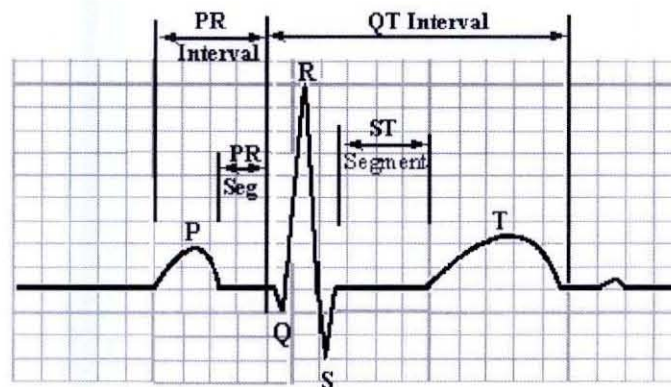


Fig 2.5 The waves and intervals of the ECG wave<sup>[18]</sup>

### 2.5.1 The P wave

The P wave is caused by atrial depolarization. In normal sinus rhythm, the SA node acts as the pacemaker. The electrical impulse from the SA node spreads over the right and left atria to cause atrial depolarization. The P wave contour is usually smooth, entirely positive and of uniform size. The P wave duration is normally less than 0.12 sec<sup>[8]</sup> and the amplitude is normally less than 0.25 mV<sup>[8]</sup>. The P wave corresponds to electrical impulses not mechanical atria contraction.

### 2.5.2 The QRS complex wave

The QRS complex represents ventricular depolarization. The duration of the QRS complex is normally 0.06 to 0.1 seconds<sup>[8]</sup>. It is measured from the beginning of the Q wave to the end of the S wave. This relatively short duration indicates that ventricular depolarization normally occurs very rapidly. The QRS voltage is as small as 0.5-0.7mV<sup>[8]</sup>.

### 2.5.3 The T wave

The T wave is caused by ventricular repolarization. The duration of the T wave is measured from the beginning of the wave to the end. The repolarization process begins before the T wave<sup>[8]</sup>. The area encompassed by the T wave may be a little smaller or larger than that encompassed by the QRS complex; it is usually about two-thirds that of the latter. The upstroke of the T wave is less steep than the down stroke<sup>[8]</sup>.

### 2.5.4 The U wave

The wave followed by the T wave is called u wave. It is not always seen in normal ECG waves. It is common in slow heart rates and reflects heart abnormality.

### 2.5.5 The PR interval

The PR interval represents the amount of time required for the depolarization process to spread from its origin in the sinus node, through the atria, to and through the atrioventricular node, down the bundle branches and their sub branches and to the ventricular muscle. It is measured from the beginning of the P wave to the beginning of the QRS complex. The difference between the intervals as measured to the beginning

of the Q wave, and as measured to the R wave, is usually about 0.02 second but may be as 0.04 second <sup>[8]</sup>.

### **2.5.6 The QT interval**

The QT interval represents the amount of time required for depolarization of the ventricles, plus the amount of time required for their repolarization. The QT interval is measured from the beginning of the Q wave of the QRS complex to the end of the T wave. The duration of the QT interval varies with age, gender, and heart rate. It should not exceed 0.40 second <sup>[8]</sup>.

### **2.5.7 The ST interval**

The duration of the ST segment represents the amount of time during which the ventricular muscles is depolarized <sup>[8]</sup>. It is determined by measuring the interval of time from the end of the S wave to the beginning of the T wave.

## **2.6 The Cardiac Arrhythmias**

The term "arrhythmia" refers to any change from the normal sequence of electrical impulses, causing abnormal heart rhythms. Arrhythmias may be completely harmless or life-threatening. Some arrhythmias are so brief that the overall heart rate or rhythm isn't greatly affected. But if arrhythmias last longer, they may cause the heart rate to be too slow or too fast or the heart rhythm to be erratic – so the heart pumps less effectively <sup>[6]</sup>.

### **2.6.1 Sinus tachycardia**

In sinus tachycardia the impulses originate regularly as they should in the sinus node. The current travels along its usual path through the atria and ventricles. The rate, however is faster than normal (100 to 160 BPM) <sup>[1]</sup>. The rate shown in fig 2.6 is 140 per minute.

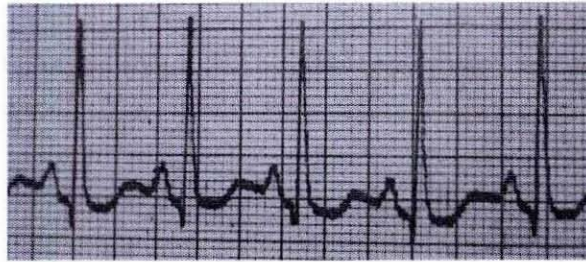


Fig 2.6 Tachycardia

### 2.6.2 Sinus bradycardia

In sinus bradycardia the pacemaker is the sinus node and conduction is normal. The rhythm is regular, but the rate, below 60, is too slow. In the tracing shown in fig 2.7 the rate is 38 per minute <sup>[1]</sup>.

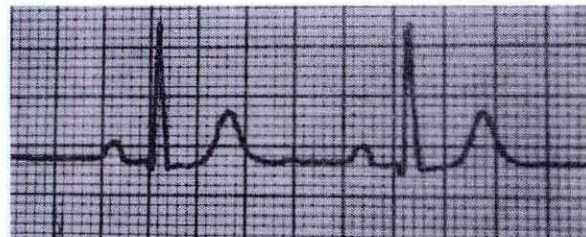


Fig 2.7 Bradycardia

## CHAPTER 3

### PROJECT OVERVIEW

#### 3.1 Hardware

Figure 3.1 shows the block diagram of the whole set up. The microcontroller is programmed to generate the normal ECG waves, bradycardia and tachycardia waves. The power supply is used to turn on the microcontroller. We use an input of 5V to power the microcontroller. Then the PWM output of the microcontroller goes through the filter network to give the desired ECG waveform.

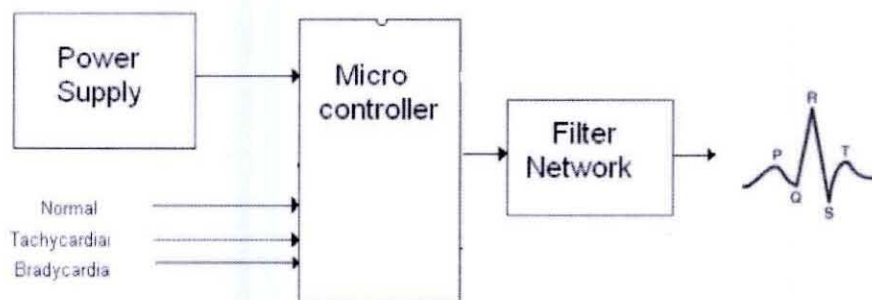


Fig 3.1 The block diagram<sup>[19]</sup>

#### 3.1.1 Microcontroller (ATMega32)

We have used Atmel ATMega32 microcontroller for this project. It is a high performance, low-power AVR 8-bit microcontroller<sup>[24]</sup>. The microcontroller has different registers for different purposes. To generate the waves we used pulse width modulation technique. To generate PWM we use register TCCR1A and TCCR1B. TCCR1A has 4 different modes, to generate PWM we use mode number 3 called fast PWM Output compare register is called OCR, it generates a value to compare the mode of output. The output compare pin gives the output wave. The microcontroller pin configuration is shown in figure 3.2. There are four ports port A, port B, port C and port D. These ports can be used as input or output ports which can either take in 5 volts or give an output of 5 volts. If not used as an input or output port these have other functions. Pin number 6, 7 and 8 are used to connect the STK 500 downloader. Pin 40 is the ADC converter pin, this was used to convert the analog voltage connected to the variable resistor. We actually get the output wave from pin 19.



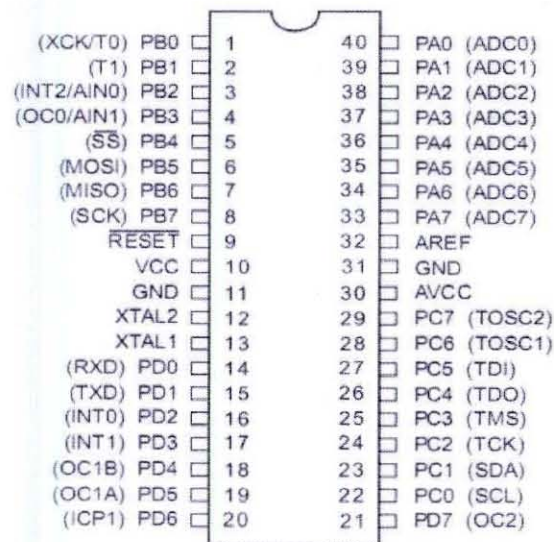


Fig 3.2 The ATmega32 microcontroller pin configuration

### 3.1.2 Filter network

The output from the microcontroller is in PWM. That is square wave pulses which is changing duty cycle. The output from the microcontroller is fed into the filter network which is responsible to give curvature to the square wave digital pulses. It is a simple RC low pass filter network circuit that passes low frequencies well but attenuates the unwanted higher frequencies. By the varying duty cycle the low pass filter charges while the PWM signal is on and discharges while the PWM is off, generating the analog output voltage. The circuit used is shown in figure 3.3.

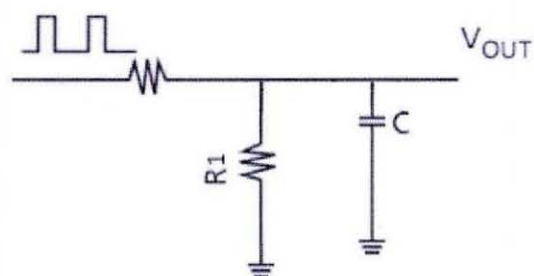


Fig 3.3 The filter network

### 3.1.2 Components and specifications

- ATmega32 microcontroller
- 1 Selector switch
- 1k $\Omega$  Resistor
- 10k $\Omega$  Resistor
- 2.2 $\mu$ F Capacitor
- 1mH Inductor
- 10k $\Omega$  Variable resistor
- LED
- STK-500 Downloader

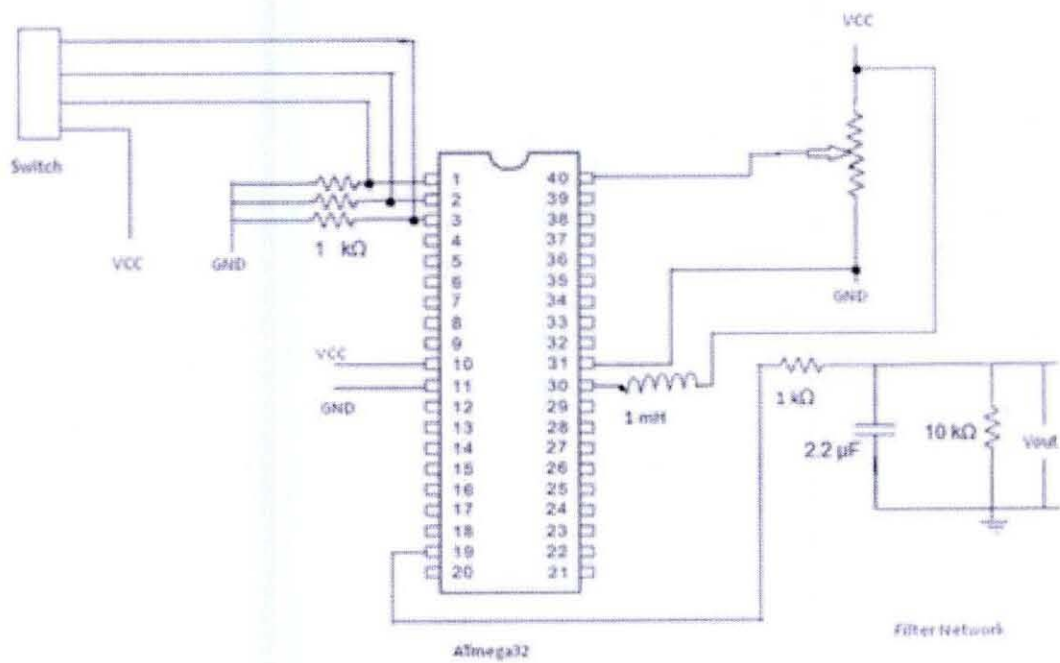


Fig 3.4 Circuit diagram

### 3.2 Software

As mentioned earlier we used a C program and compiled it into the microcontroller to give the waves. We used a lookup table in the code.

#### 3.2.1 Lookup table

The lookup table is a mathematical table to generate waves. We calculated the lookup table by drawing each ECG wave in a graph paper. The samples can be taken from 128 to 1024 on the x-axis; these samples can be taken according to our convenience. The y-axis has to be 256. Taking 256 on the y-axis generated a wave of amplitude 2.4 volts. The figure 3.5 shows one of the waves we drew on the graph paper and the calculated data is shown in figure 3.6.

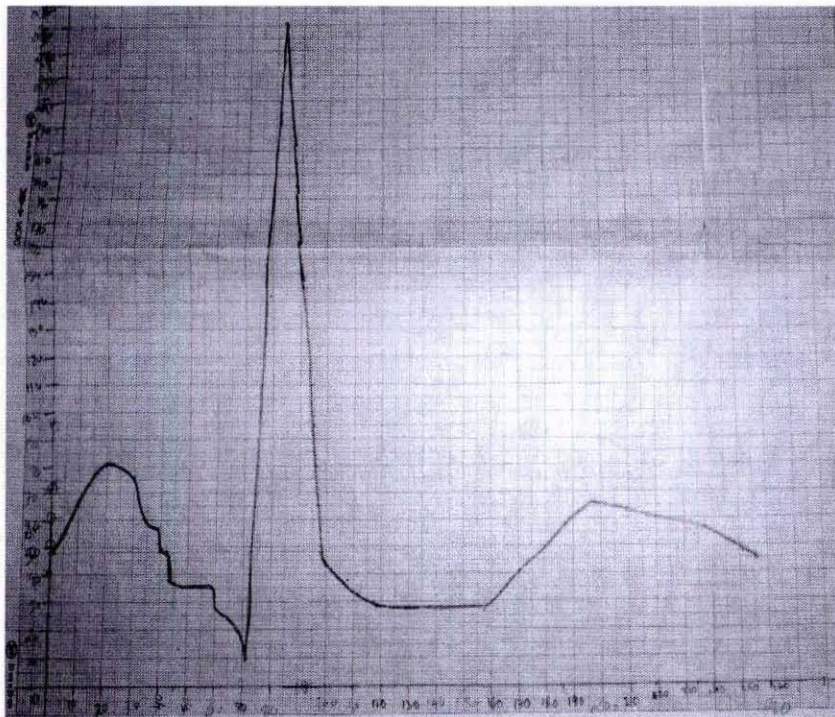


Fig 3.5 Wave drawn in graph to calculate lookup table

```

0x30,0x31,0x32,0x34,0x36,0x37,0x3a,0x3c,0x3f,0x42,0x44,0x45,0x57,0x49,0x4b,0x4c,
0x4e,0x4f,0x50,0x51,0x51,0x51,0x51,0x50,0x50,0x4f,0x4e,0x4e,0x4d,0x4c,0x4c,0x46,
0x42,0x3e,0x3c,0x3a,0x3a,0x3a,0x39,0x32,0x32,0x32,0x32,0x26,0x25,0x24,0x24,0x24,
0x24,0x24,0x23,0x23,0x23,0x23,0x23,0x23,0x23,0x23,0x23,0x23,0x23,0x1b,0x1b,0x1a,0x19,
0x18,0x18,0x17,0x15,0x14,0x13,0x0f,0x09,0x1e,0x32,0x46,0x5a,0x6e,0x82,0x96,0xaa,
0xbe,0xd2,0xe6,0xfb,0xf0,0xe1,0xd7,0xc8,0xb9,0xaa,0x9b,0x91,0x82,0x78,0x6e,0x5f,
0x50,0x41,0x30,0x2d,0x2b,0x2a,0x29,0x28,0x26,0x25,0x24,0x23,0x23,0x22,0x21,0x21,
0x20,0x1f,0x1e,0x1e,0x1e,0x1d,0x1d,0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,
0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,
0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,0x1d,
0x1e,0x1f,0x20,0x21,0x22,0x23,0x24,0x25,0x26,0x27,0x28,0x29,0x2a,0x2b,0x2c,0x2d,
0x2e,0x2f,0x30,0x31,0x32,0x33,0x34,0x35,0x36,0x37,0x38,0x39,0x3a,0x3b,0x3d,0x3e,
0x3f,0x3f,0x40,0x41,0x41,0x42,0x42,0x42,0x42,0x42,0x41,0x41,0x41,0x40,0x40,
0x40,0x3f,0x3f,0x3f,0x3e,0x3e,0x3e,0x3d,0x3d,0x3d,0x3c,0x3c,0x3c,0x3c,0x3c,0x3c,
0x3c,0x3c,0x3b,0x3b,0x3b,0x3b,0x3a,0x3a,0x3a,0x39,0x39,0x39,0x38,0x37,
0x37,0x37,0x36,0x35,0x35,0x34,0x33,0x32,0x32,0x31,0x31,0x30,0x30,0x30,0x30,0x30

```

Fig 3.6 Data calculated from the graph

### 3.2.2 The AVR studio 4 compiler

The C program was written in the AVR Studio 4 compiler and compiled there. The STK 500 downloader was connected to the computer which downloaded the program into the microcontroller.

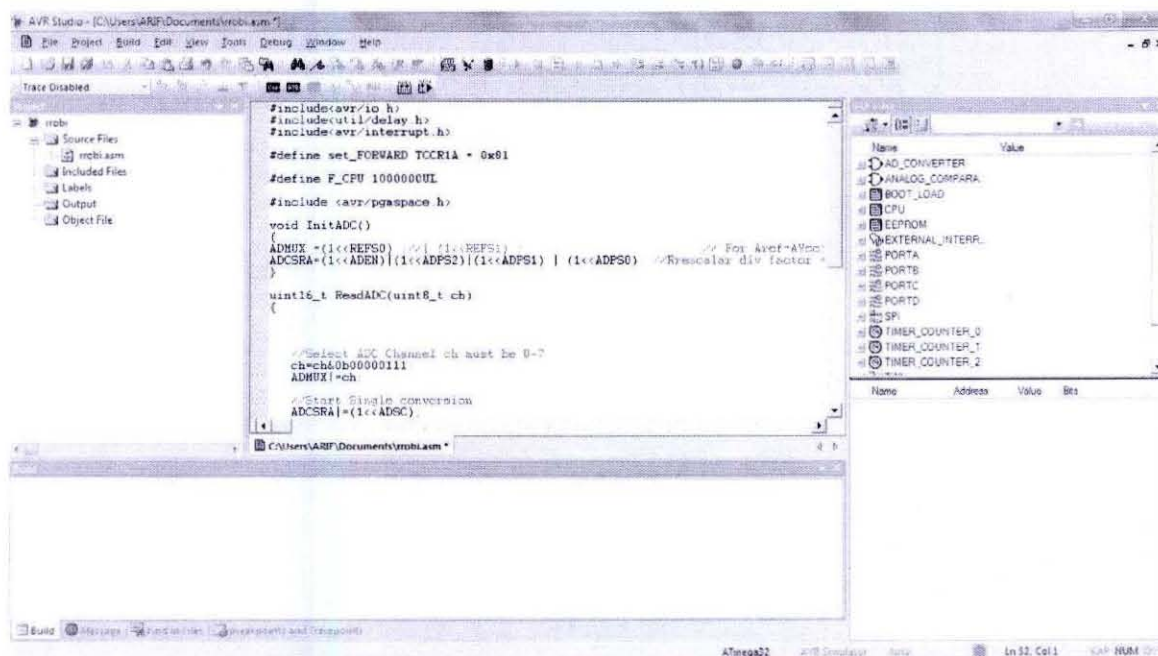


Fig 3.7 AVR Studio 4 compiler

### 3.2.3 Programming the microcontroller:

The following C code was written to program the microcontroller to generate the desired waves:

```
#include<avr/io.h>
#include<util/delay.h>
#include<avr/interrupt.h>

#define set_FORWARD TCCR1A = 0x81

#define F_CPU 1000000UL

#include <avr/pgmspace.h>

void InitADC()
{
    ADMUX =(1<<REFS0) ;//| (1<<REFS1) ;           // For Aref=AVcc;
    ADCSRA=(1<<ADEN)|(1<<ADPS2)|(1<<ADPS1) | (1<<ADPS0); //Rrescalar div factor =128
}

uint16_t ReadADC(uint8_t ch)
{

    //Select ADC Channel ch must be 0-7
    ch=ch&0b00000111;
    ADMUX|=ch;

    //Start Single conversion
    ADCSRA|=(1<<ADSC);

    //Wait for conversion to complete
    while(!(ADCSRA & (1<<ADIF)));

    //Clear ADIF by writing one to it

    ADCSRA|=(1<<ADIF);

    return(ADC);
}
```









```

0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,
0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,
0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,
0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,
0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,
0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,
0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,
0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,
0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,
0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,
};

```

```

const uint8_t tachywave[] PROGMEM= //256 values

```

```
{
```

```

0x30,0x31,0x32,0x34,0x36,0x37,0x3a,0x3c,0x3f,0x42,0x44,0x45,0x57,0x49,0x4b,0x4c,
0x4e,0x4f,0x50,0x51,0x51,0x51,0x51,0x50,0x50,0x4f,0x4e,0x4e,0x4d,0x4c,0x4c,0x46,
0x42,0x3e,0x3c,0x3a,0x3a,0x3a,0x39,0x32,0x32,0x32,0x32,0x26,0x25,0x24,0x24,0x24,
0x24,0x24,0x23,0x23,0x23,0x23,0x23,0x23,0x23,0x23,0x23,0x23,0x23,0x23,0x1b,0x1b,0x1a,0x19,
0x18,0x18,0x17,0x15,0x14,0x13,0x0f,0x09,0x1e,0x32,0x46,0x5a,0x6e,0x82,0x96,0xaa,
0xbe,0xd2,0xe6,0xfb,0xf0,0xe1,0xd7,0xc8,0xb9,0xaa,0x9b,0x91,0x82,0x78,0x6e,0x5f,
0x50,0x41,0x30,0x2d,0x2b,0x2a,0x29,0x28,0x26,0x25,0x24,0x23,0x23,0x22,0x21,0x21,
0x20,0x1f,0x1e,0x1e,0x1e,0x1d,0x1d,0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,
0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,
0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,0x1c,0x1d,
0x1e,0x1f,0x20,0x21,0x22,0x23,0x24,0x25,0x26,0x27,0x28,0x29,0x2a,0x2b,0x2c,0x2d,
0x2e,0x2f,0x30,0x31,0x32,0x33,0x34,0x35,0x36,0x37,0x38,0x39,0x3a,0x3b,0x3d,0x3e,
0x3f,0x3f,0x40,0x41,0x41,0x42,0x42,0x42,0x42,0x42,0x41,0x41,0x41,0x41,0x40,0x40,
0x40,0x3f,0x3f,0x3f,0x3e,0x3e,0x3e,0x3d,0x3d,0x3d,0x3c,0x3c,0x3c,0x3c,0x3c,0x3c,
0x3c,0x3c,0x3b,0x3b,0x3b,0x3b,0x3b,0x3b,0x3a,0x3a,0x3a,0x39,0x39,0x39,0x39,0x38,0x37,
0x37,0x37,0x36,0x35,0x35,0x34,0x33,0x32,0x32,0x31,0x31,0x30,0x30,0x30,0x30,0x30
};

```

```
void delay_ms(unsigned int ms){
```

```

    while(ms){
        _delay_ms(1.000);
        ms--;
    }

```

```
}
```

```
void delay_us(unsigned int us){
```

```

    while(us){
        _delay_us(1.000);
        us--;
    }

```

```
}
```

```
int main()
```

```
{
```

```
    int adc_val;
```

```
        int var;
```

```
    InitADC();
```

```
    DDRB=0x00;
```

```
    DDRD |= (1<<PD5) | (1<<PD4);
```

```
    adc_val=ReadADC(0);
```

```
    volatile unsigned int i=0;
```

```
    TCCR1A |= (1<<COM1A1) | (1<<COM1B1) |(1<<WGM10) | (1<<WGM11) ;
```

```
    TCCR1B |= (1<<CS10) | (1<<WGM12);
```

```
    while(1){
```

```
        if(PINB & (1<<PB0)){
```

```
            i=0;
```

```
            //delay_ms(1000);
```

```
            OCR1A = 0x00;
```

```
            set_FORWARD;
```

```
            //delay_ms(1000);
```

```
            while(i<1024)
```

```
            {
```

```
                OCR1A=pgm_read_byte(&bradywave[i]);
```

```
                i++;
```

```
                    delay_ms(13);
```

```
            }
```

```
        }
```

```
        if(PINB & (1<<PB1)){
```

```
i=0;

OCR1A = 0x00;
set_FORWARD;

while(i<256)
{
OCR1A=pgm_read_byte(&tachywave[i]);
i++;
    delay_ms(14);

}

}

    if(PINB & (1<<PB2)){

InitADC();

adc_val=ReadADC(0);
    adc_val=adc_val/16;

    if(adc_val==63){
        var=8;
    }

    if(adc_val<63 && adc_val>5){
        var=6;
    }

    if(adc_val<5 && adc_val>0){
        var=4;
    }
i=0;

OCR1A = 0x00;
set_FORWARD;
```

```
while(i<1024)
{
OCR1A=pgm_read_byte(&ecgwave[i]);
i++;
    delay_ms(var);

}

}

}
```

## CHAPTER 4

### EXPERIMENTAL RESULTS

#### 4.1 Generation of the Basic Waves

The first step was to generate the basic waves, which are saw-tooth wave and sine wave using the microcontroller. We wrote a program in C which used a look-up table. We calculated the look-up table for each waves and used it in the C program. For sine wave and saw-tooth wave we took 256 samples because it is a less complicated wave shapes. But as ECG wave is a complex wave so we needed more samples, so we took 1024 samples to get the wave shape. The look-up table is very convenient because the same C program can be used to generate all the waves, we only had to change the look-up table keeping the rest of the code same. We downloaded the program into the microcontroller using the STK-500 downloader or TOP 2005 Programmer. We used PWM mode of the microcontroller so the output was a pulse width modulation of changing duty cycle. This output from the microcontroller was passed through a low-pass RC filter network to give curvature to the pulsed square waves; we saw the output in a digital oscilloscope. Figure 4.1 and figure 4.2 show the output shown in the digital oscilloscope.

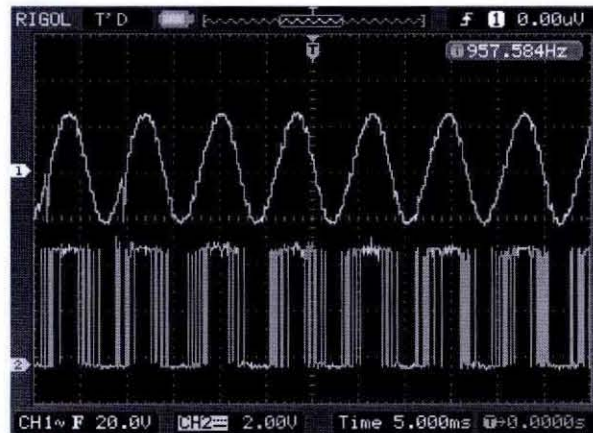


Fig 4.1 The Sine wave seen in the digital oscilloscope

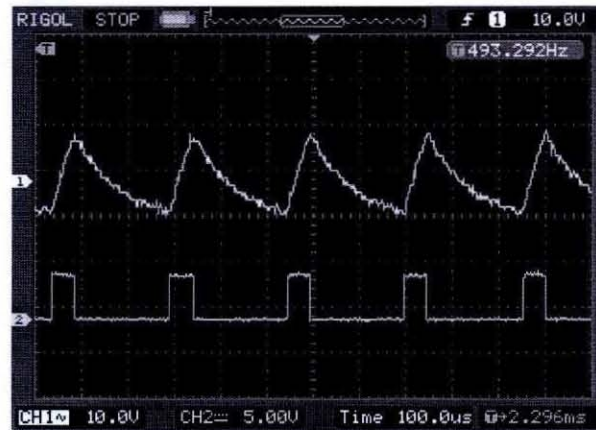


Fig 4.2 The Saw-Tooth wave

#### 4.2 Generation of the Basic ECG Wave

After we successfully generated the basic waves our focus was on the basic ECG wave. We calculated the look-up table for the ECG wave and used the same C program used to generate the basic waves only replacing the look-up table. We downloaded the program into the microcontroller and using the exact same filter network we saw the output in the digital oscilloscope. The output ECG wave we generated is shown in figure 4.3. Figure 4.4 shows the whole set up after the code was downloaded into the microcontroller.

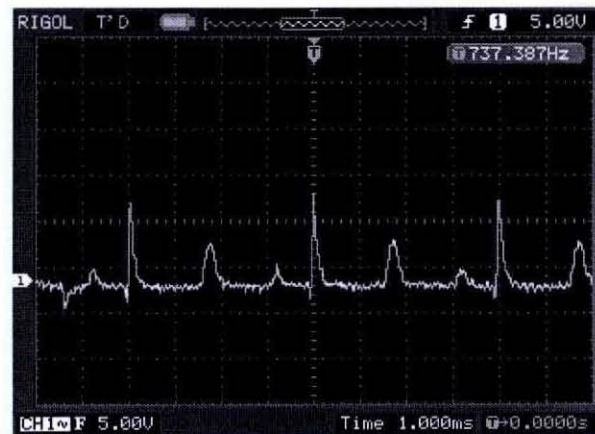


Fig 4.3 The ECG wave

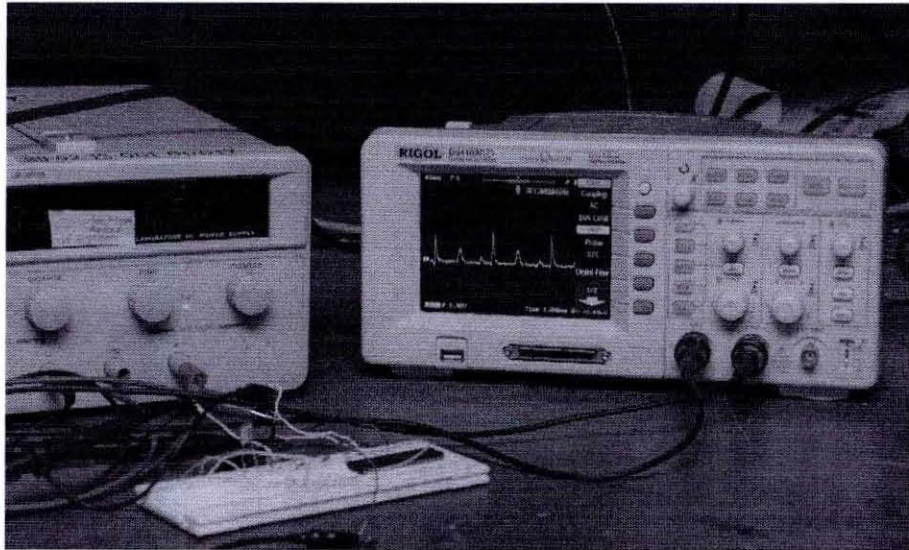


Fig 4.4 the set-up to generate ECG wave

### 4.3 Variable BPM Option

The normal heart rate is 72 beats per minute (BPM), for sinus tachycardia it is greater than 90 BPM, and for sinus bradycardia it is less than 60 BPM. Our next objective was to put a variable BPM option for the simulator. We used a 10k $\Omega$  variable resistor as a knob which allowed us to switch from different BPM outputs.

The three points of the 10k $\Omega$  variable resistor is connected to the ground and Vcc. The 5V supplied to the variable resistor is the analog voltage; it can be varied by the variable resistor. The analog voltage is converted into digital using the ADC port 0 of the microcontroller. Pin 40 of the microcontroller is connected to the third pin of the variable resistor. The analog voltage then is converted into digital voltage and controls the output from the microcontroller. Pin 30 is connected to Vcc through a 1mH inductor. The inductor is used so that high current does not pass through the microcontroller and prevents it from becoming hot.

#### 4.4 Generating Cardiac Arrhythmias Waves

After generating the normal ECG with a variable BPM option, our goal was to generate sinus tachycardia and sinus bradycardia. We used a selector switch which allows selecting three different options normal ECG wave with a knob of three options 60 BPM, 78 BPM and 120 BPM, sinus tachycardia of 136 BPM and sinus bradycardia of 38 BPM. The pins PB0, PB1 and PB2 of the microcontroller were grounded through a 10k $\Omega$  resistor to make its output zero. Then we connected 5V supply through a selector switch to these three ports. The selector switch enables 5V to any one of the three ports. Whichever port gets the 5V will be high and will give that output.

#### 4.5 Final Results

We initially implemented the circuitry in the trainer breadboard. After we got the final results we implemented the circuit in a printed circuit board (PCB). The PCB was inserted in a custom designed box. The output connector, the power input and the switch and variable resistor knob were mounted on the box for it to be user friendly and convenient. The final outputs we got are shown in the diagram below. The selector switch enables the type of wave generated bradycardia, tachycardia and normal ECG. For the normal ECG we get a variable BPM option of 60 BPM, 78 BPM, and 120 BPM. This variable BPM is achieved by the variable resistor.

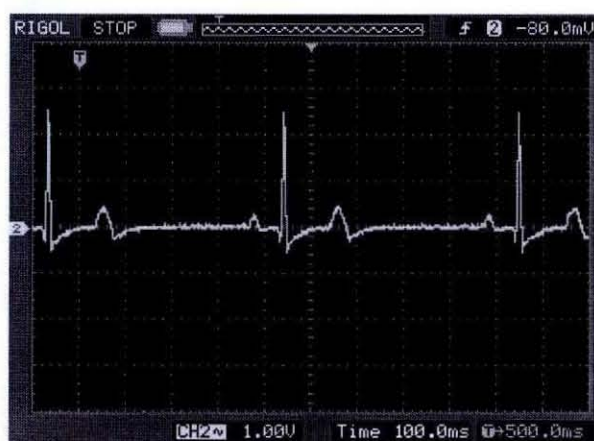


Fig 4.6 Normal ECG 120 BPM (60,000/500=120)





Fig 4.7 Normal ECG 78 BPM  $(60,000/760=78)$

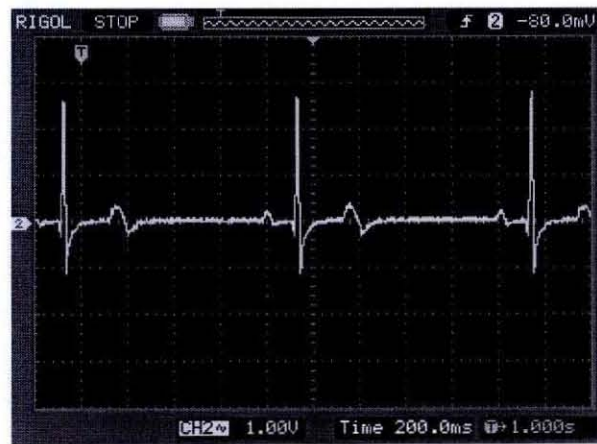


Fig 4.8 Normal ECG 60 BPM  $(60,000/1000=60)$



Fig 4.9 Tachycardia 136 BPM ( $60,000/440=136$ )

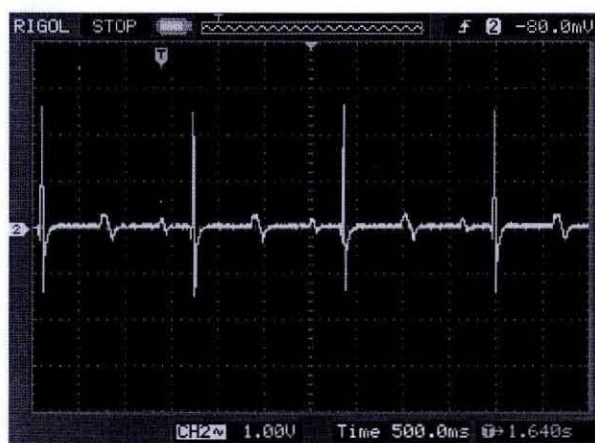


Fig 4.10 Bradycardia 38 BPM ( $60,000/1640=38$ )

#### 4.6 BPM Calculation

BPM is the number of heart beats per minute. Its is simply the number of complete ECG waves generated in one minute. The output we saw in the digital oscilloscope shows the time period of one complete wave shape in milliseconds. So we converted the milliseconds into minutes. So the BPM was calculated by dividing 60,000 by the time period of one complete cycle in milliseconds from the oscilloscope.

## CHAPTER 5

### THE ECG SIMULATOR IN OPERATION

The simulator is powered by a 5V DC supply. Once the power supply and the output leads are connected, the selector switch enables to give the desired wave shape output. The selected wave shape is also indicated by the LED. The simulator can generate bradycardia of 38 BPM, Tachycardia of 136 BPM and normal ECG wave of three variable BPM of 120,78 and 60. The diagrams below give the clear idea of the operation of the ECG simulator.

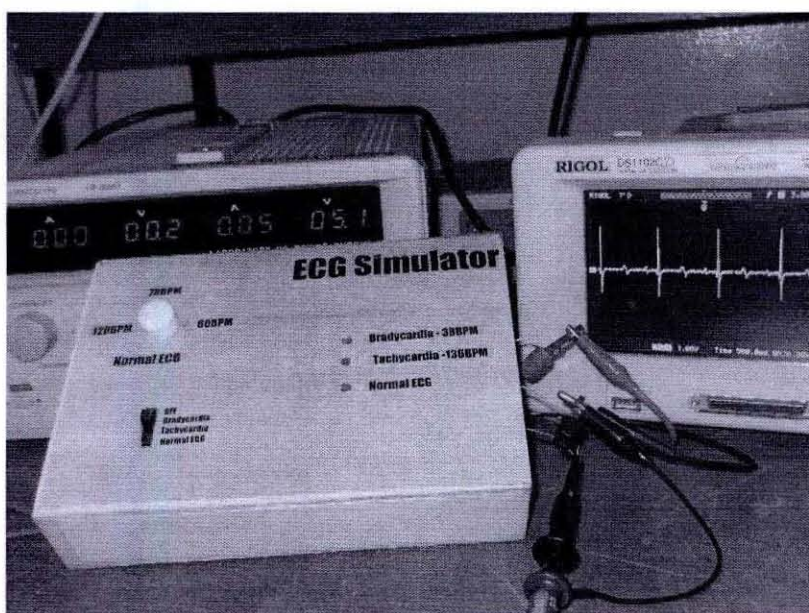


Fig 5.1 Bradycardia of 38 BPM is generated

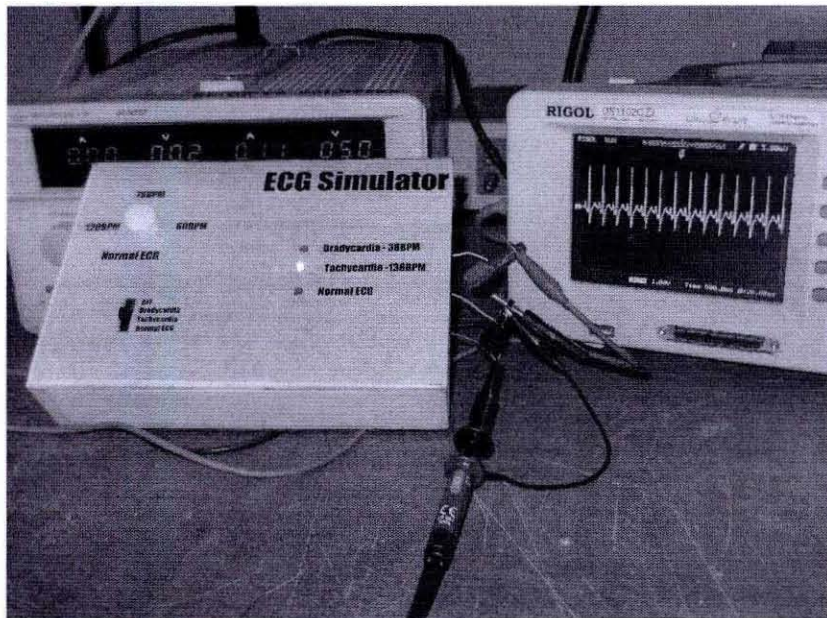


Fig 5.2 Tachycardia of 136 BPM is generated

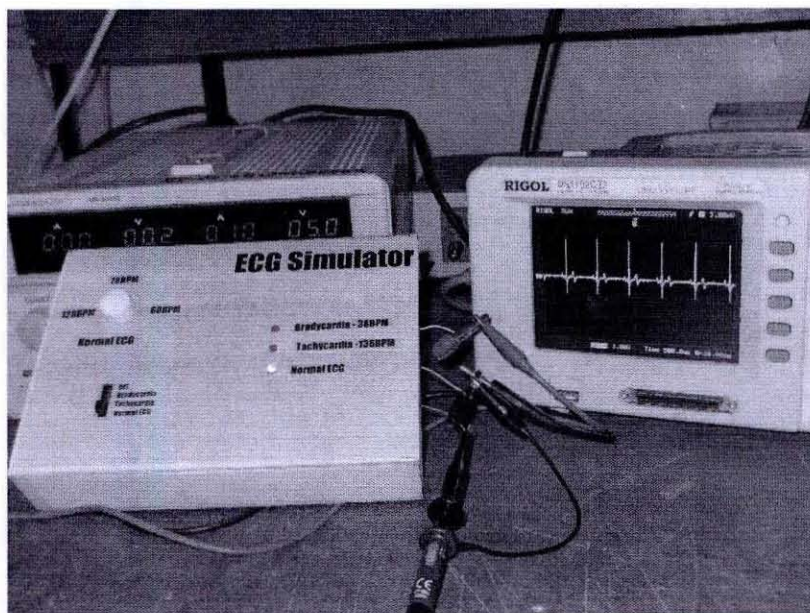


Fig 5.3 Normal ECG of 60 BPM is generated

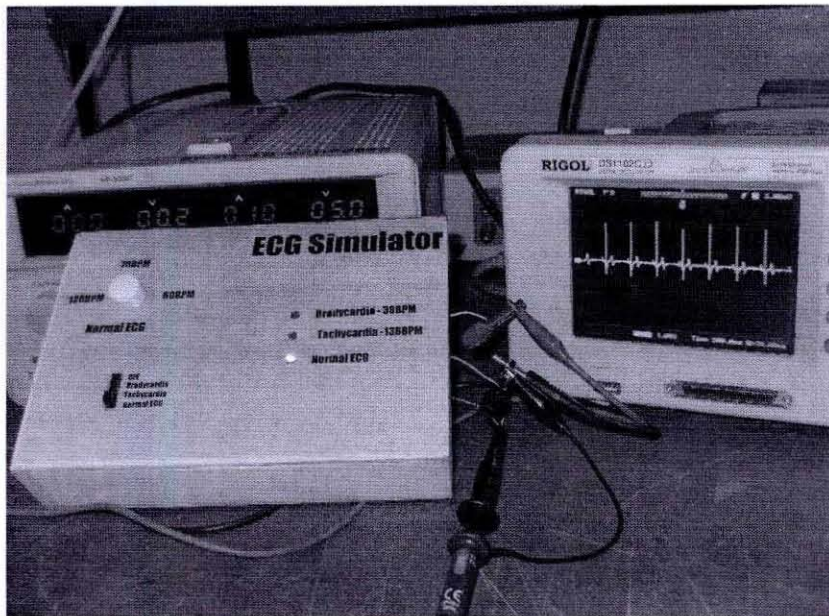


Fig 5.4 Normal ECG of 78 BPM is generated



Fig 5.5 Normal ECG of 120 BPM is generated

## CHAPTER 6

### COST ANALYSIS

The commercially available ECG simulators cost in the range of 200 USD to 1000 USD. Some of them are illustrated below:



Figure 6.1 Simulaids Stat Manikin Interactive Ecg Simulator<sup>[17]</sup>

The simulator shown in figure 6.1 costs \$709<sup>[17]</sup>



Figure 6.2 Simulaids 3-lead Simulator<sup>[17]</sup>

The simulator in figure 6.2 costs \$495<sup>[17]</sup>



Figure 6.3 Tech Patient ECG Simulator<sup>[17]</sup>

The simulator shown in figure 6.3 costs \$265<sup>[17]</sup>

These simulators have many more features and are far more sophisticated. Our ECG simulator fulfills the low range simulator features. To test the ECG monitor just a basic ECG wave is enough. And the cost of our simulator is far cheaper. The table below shows the cost of each component used in the project.

Table 6.1 the cost of each component used

Component	Cost in BDT
Atmega32 microcontroller	600
Variable resistor (10k ohm)	10
Fixed resistors (1k and 10k ohm)	10
Capacitor (2.2 uF)	2
Inductor (1mH)	3
Selector switch	10
LED	10
PCB	100
Box	55
Total	800 BDT = 11.35 USD

## CHAPTER 7

### CONCLUSION AND FUTURE WORK

Electrocardiograms are used for medical examinations, research and the development of Bio-Medical equipments. An ECG simulator thus has a large variety of applications. Some of them are as follows:

- It is used for testing ECG monitors, as it will not be necessary to supply the monitor with ECG signals generated from the surface of human body for testing. The artificially produced signals will be used to verify the operation of the device [17].
- The same simulator can be used for testing other equipments such as Halter Recorder, fetal and adult ECG monitors, and stress test systems.
- The simulator facilitates the teaching and learning of clinical aspects that would otherwise be difficult or potentially dangerous to teach.

We have successfully generated the basic ECG wave with a variable BPM option and the two arrhythmias waves. The only limitation of our simulator is the high amplitude. The standard ECG wave has amplitude of 0.7mV and our generated wave amplitude is 2.4V. We were unable to reduce the amplitude to such a low ratio, due to high noise in the system. It can be further developed to generate waves with precise amplitude and also variable amplitude option.

Our Simulator has a single output. The same simulator can be implemented and developed to have a 3 lead or a 12 lead output to connect all 12 leads of the ECG machine to the simulator.

The microcontroller based ECG simulator has not been implemented in Bangladesh as yet. The commercially available simulator is very much expensive. If it can be manufactured locally it can be used in hospitals and for medical research work. It will also make possible to open a new door to the biomedical engineering sector of Bangladesh.



## APPENDIX



Application Note AC280

## Simple Digital to Analog Converter (DAC)

## Introduction

Methods for implementing a Digital to Analog Converter (DAC) differ greatly depending on the required resolution, speed, and cost. For the many applications that do not have high resolution or speed requirements, a simple and cost-effective approach to converting digital to analog is to use a pulse width modulation (PWM) function together with an RC low-pass filter (LPF), as shown in Figure 1. An RC LPF is one of the simplest analog electronic filters that passes low frequencies well but attenuates (or reduces) the unwanted higher frequencies. By varying the duty cycle, usually controlled by a state machine or microcontroller, the LPF charges while the PWM signal is "on" and discharges while the PWM signal is "off", generating an analog output voltage. This application brief describes the techniques of PWM control (using the Actel Core8051 microcontroller) as well as design criteria for the RC filter.

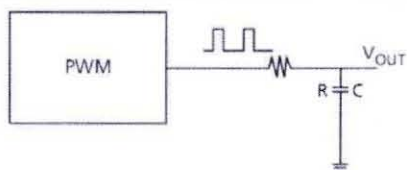


Figure 1 • Simple DAC Concept Diagram

## Application Implementation

Actel Core8051 and CorePWM are designed with integration in mind. Core8051 and CorePWM can be implemented in various Actel device families to accomplish a simple DAC application with some external circuitry (Figure 2).

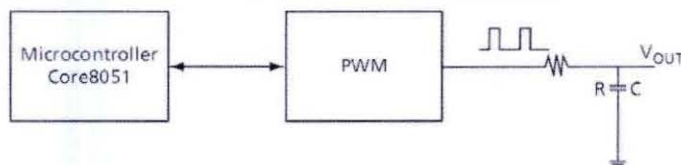


Figure 2 • Simple DAC Implementation with Core8051 and CorePWM

## Features

- High-performance, Low-power AVR<sup>®</sup> 8-bit Microcontroller
- Advanced RISC Architecture
  - 131 Powerful Instructions – Most Single-clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 16 MIPS Throughput at 16 MHz
  - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory segments
  - 32K Bytes of In-System Self-programmable Flash program memory
  - 1024 Bytes EEPROM
  - 2K Byte Internal SRAM
  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/100 years at 25°C<sup>(1)</sup>
  - Optional Boot Code Section with Independent Lock Bits
  - In-System Programming by On-chip Boot Program
  - True Read-While-Write Operation
  - Programming Lock for Software Security
- JTAG (IEEE std. 1149.1 Compliant) Interface
  - Boundary-scan Capabilities According to the JTAG Standard
  - Extensive On-chip Debug Support
  - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Four PWM Channels
  - 8-channel, 10-bit ADC
    - 8 Single-ended Channels
    - 7 Differential Channels in TQFP Package Only
    - 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x
  - Byte-oriented Two-wire Serial Interface
  - Programmable Serial USART
  - Master/Slave SPI Serial Interface
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated RC Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby
- I/O and Packages
  - 32 Programmable I/O Lines
  - 40-pin PDIP, 44-lead TQFP, and 44-pad QFN/MLF
- Operating Voltages
  - 2.7 - 5.5V for ATmega32L
  - 4.5 - 5.5V for ATmega32
- Speed Grades
  - 0 - 8 MHz for ATmega32L
  - 0 - 16 MHz for ATmega32
- Power Consumption at 1 MHz, 3V, 25°C for ATmega32L
  - Active: 1.1 mA
  - Idle Mode: 0.35 mA
  - Power-down Mode: < 1 µA



**8-bit AVR<sup>®</sup>**  
**Microcontroller**  
**with 32K Bytes**  
**In-System**  
**Programmable**  
**Flash**

**ATmega32**  
**ATmega32L**

Note: Not Recommended for new designs.

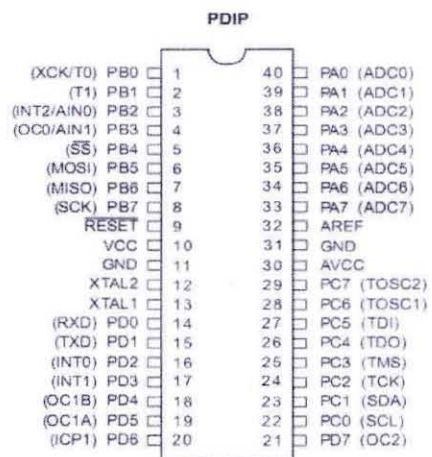
25030-AVR-07/09





## Pin Configurations

Figure 1. Pinout ATmega32



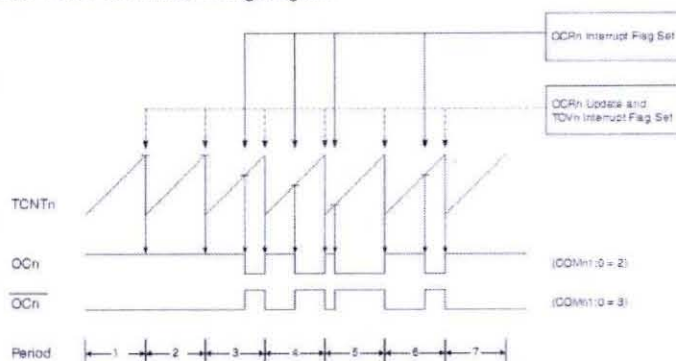
## ATmega32(L)

### Fast PWM Mode

The fast Pulse Width Modulation or fast PWM mode (WGM01:0 = 3) provides a high frequency PWM waveform generation option. The fast PWM differs from the other PWM option by its single-slope operation. The counter counts from BOTTOM to MAX then restarts from BOTTOM. In non-inverting Compare Output mode, the Output Compare (OC0) is cleared on the compare match between TCNT0 and OCR0, and set at BOTTOM. In inverting Compare Output mode, the output is set on compare match and cleared at BOTTOM. Due to the single-slope operation, the operating frequency of the fast PWM mode can be twice as high as the phase correct PWM mode that use dual-slope operation. This high frequency makes the fast PWM mode well suited for power regulation, rectification, and DAC applications. High frequency allows physically small sized external components (coils, capacitors), and therefore reduces total system cost.

In fast PWM mode, the counter is incremented until the counter value matches the MAX value. The counter is then cleared at the following timer clock cycle. The timing diagram for the fast PWM mode is shown in Figure 32. The TCNT0 value is in the timing diagram shown as a histogram for illustrating the single-slope operation. The diagram includes non-inverted and inverted PWM outputs. The small horizontal line marks on the TCNT0 slopes represent compare matches between OCR0 and TCNT0.

Figure 32. Fast PWM Mode, Timing Diagram



The Timer/Counter Overflow Flag (TOV0) is set each time the counter reaches MAX. If the interrupt is enabled, the interrupt handler routine can be used for updating the compare value.

In fast PWM mode, the compare unit allows generation of PWM waveforms on the OC0 pin. Setting the COM01:0 bits to 2 will produce a non-inverted PWM and an inverted PWM output can be generated by setting the COM01:0 to 3 (See Table 40 on page 81). The actual OC0 value will only be visible on the port pin if the data direction for the port pin is set as output. The PWM waveform is generated by setting (or clearing) the OC0 Register at the compare match between OCR0 and TCNT0, and clearing (or setting) the OC0 Register at the timer clock cycle the counter is cleared (changes from MAX to BOTTOM).

The PWM frequency for the output can be calculated by the following equation:

$$f_{OCnPWM} = \frac{f_{clk}/O}{N \cdot 256}$$

The N variable represents the prescale factor (1, 8, 64, 256, or 1024).

The extreme values for the OCR0 Register represents special cases when generating a PWM waveform output in the fast PWM mode. If the OCR0 is set equal to BOTTOM, the output will be

## BIBLIOGRAPHY

### Book:

- [1] Understanding Electrocardiography physiological and interpretive concepts.  
Edwin G. Zalis and Mary H. Conover
- [2] W.G. John, (1998) "Medical Instrumentation: Application and Design"  
Edition. 3rd, Toronto: Johan Wiley and sons
- [3] Cromwell L, Weibe F. J., "Biomedical Instrumentation and Measurements"
- [4] Vasilevich M Y 2004, "A Generator for Testing Signals of electroencephalographs  
And Electrocardiographs", Biomedical Engineering, 38(3), 162-4

### Web sites:

- [5] Wikipedia. 2010. The Human Heart [Accessed 2<sup>nd</sup> August 2010]  
Web: <[http://en.wikipedia.org/wiki/The\\_human\\_heart](http://en.wikipedia.org/wiki/The_human_heart)>
- [6] American Heart Association.[Accessed 6<sup>th</sup> December 2010]  
Web: <<http://www.americanheart.org/presenter.jhtml?identifier=34>>
- [7] National Heart Lung and Blood Institute. 2010. How the Heart Works [Accessed  
2<sup>nd</sup> August 2010]  
Web: <[http://www.nhlbi.nih.gov/health/dci/Diseases/hhw/hhw\\_electrical.html](http://www.nhlbi.nih.gov/health/dci/Diseases/hhw/hhw_electrical.html)>
- [8] Mc Gill. The EKG Waveform [Accessed 2<sup>nd</sup> August 2010]  
Web: <<http://sprojects.mmi.mcgill.ca/cardiophysio/NormalEKG.htm>>
- [9] Cardiovascular Physiology Concepts. Electrocardiogram (EKG, ECG) [Accessed  
2<sup>nd</sup> August 2010]  
Web: <<http://www.cvphysiology.com/Arrhythmias/A009.htm>>

- [10] *Heart and Circulatory, Tachycardia*, Harvard Medical School for InteliHealth.  
Source on the Internet: Aetna InteliHealth,  
Web: <<http://www.intelihealth.com/IH/ih/IH/WSIHW000/805>>
- [11] Dale Technology, ECG/Arrhythmia Simulator - DALE13, 5200 Convair Drive,  
Carson City, NV 89706, Source on the Internet: daletech.com,  
Web: <[http://www.daletech.com/main/product\\_info.php](http://www.daletech.com/main/product_info.php)>
- [12] M311 ECG Simulator, Fogg System Company, Inc. 15592 East Batavia Drive;  
Aurora, CO 80011, Source on the Internet: allwave.com,  
Web: <<http://www.allwave.com/pc-309-6-m311-ecg-simulator.aspx>>
- [13] Cardiovascular System I: The Beating Heart, Source on the Internet: BioCoach Activity,  
Web: <[http://www.phschool.com/science/biology\\_place/biocoach/cardio1/intro.html](http://www.phschool.com/science/biology_place/biocoach/cardio1/intro.html)>
- [14] Heart Electrical Activity, Source on the Internet: HeartSite.com,  
Web: [http://www.heartsite.com/html/electrical\\_activity.html](http://www.heartsite.com/html/electrical_activity.html)
- [15] The QRS Complex, The EKG Waveform, Electrocardiology and Cardiac  
Arrhythmias  
Web: <http://sprojects.mmi.mcgill.ca/cardiophysio/EKGQRScomplex.html>
- [16] Lookup Table, Wikipedia.  
Web: [http://en.wikipedia.org/wiki/Lookup\\_table](http://en.wikipedia.org/wiki/Lookup_table)
- [17] [www.amazon.com](http://www.amazon.com)

Thesis:

- [18] Gurpinder Kaur, Design and Development of Dual Channel ECG Simulator and  
Peak Detector, Masters Thesis, Thapar Institute of Engineering & Technology  
Deemed University, 2006.
- [19] Mayank Gupta, ECG Simulator, Chameli Devi Institute of Technology &  
Management.
- [20] Paul J. Michalek, an authentic ECG simulator, University of Central Florida,  
2006.

- [21] Muhammad Shah Noor bin Minir, Shamsun Nahar, Tahsina Farah, Automatic Detection of Premature Ventricular Contraction Beat in ECG Signal, Bangladesh University of Engineering & Technology, March 2009
- [22] A E Martínez, E Rossi and L Nicola Siri, Microprocessor-based simulator of Surface ECG signals, 16th Argentine Bioengineering Congress and the 5<sup>th</sup> Conference of Clinical Engineering.
- [23] Candan Caner & Mehmet Engin & Erkan Zeki Engin, The Programmable ECG Simulator, Published online: 25 March 2008, Springer Science + Business Media, LLC 2008.

Datasheet:

- [24] AT-mega32 datasheet