

DEPARTMENT OF ELECTRONIC AND TELECOMMUNICATION  
ENGINEERING  
UNIVERSITY OF MORATUWA



## GROUP PROJECT REPORT

### EN1093 - LABORATORY PRACTICE

---

# Portable Heart Rate Monitor

---

*Authors:*

K.K. Herath

H.M.U.D Herath

R.U. Hettiarachchi

M.N. Hettiaratchchi

*Registration Number:*

170213V

170215U

170221T

170222X

This is submitted as a partial fulfillment for the module EN1093

—  
December 30, 2018

## 1 Abstract

The proposed project measures the heart rate of a person using optical sensors. The optical sensor detects the variation of blood volume at the fingertip. In our project the sensor will be an infrared light emitting diode (IR LED) which will be on the same side of the finger. The underlying principle is that the intensity of the reflected infrared light varies on the blood volume at an instance of the fingertip, which changes proportionately to each cardiac cycle. The lighting condition in the environment is very important to avoid distortion in the signal so we have taken the following measures,

- To design a special enclosure to cover the optical sensor for accurate measurements
- To use a band filter to remove unnecessary wavelengths.

Based on the refined output signal, we used an Atmel microcontroller to calculate and display the heart rate through the LCD Display.

## 2 Acknowledgement

We would like to express our gratitude to our supervisor Mr.Asanka Rathnayake for giving us technical advice and guidance.

Our sincere thanks goes to Mr.Thilina Sameera Ambagahawaththa for the inspiration and advice given to us to compile this report in LaTeX.

We pay our gratitude to all the lecturers, instructors and other academic staff who intimately welcomed us to share their knowledge and experiences. We are very grateful to the personnel who are in charge of laboratories for allowing us to use the laboratories when needed and for the support given to solve technical problems.

## Contents

<b>1</b>	<b>Abstract</b>	<b>1</b>
<b>2</b>	<b>Acknowledgement</b>	<b>1</b>
<b>3</b>	<b>Introduction</b>	<b>3</b>
<b>4</b>	<b>Methodology</b>	<b>4</b>
4.1	Sensor . . . . .	4
4.2	Amplification and Filtering . . . . .	5
4.2.1	Calculations . . . . .	5
4.3	Calculating BPM from the analog signal . . . . .	7
4.3.1	Microcontroller . . . . .	7
4.3.2	Obtaining a threshold value . . . . .	7
4.3.3	Algorithm . . . . .	7
4.4	Enclosure and PCB Design . . . . .	9
<b>5</b>	<b>Implementation and Results</b>	<b>10</b>
<b>6</b>	<b>Conclusion</b>	<b>12</b>
<b>7</b>	<b>Online Materials</b>	<b>13</b>

### 3 Introduction

This is a simple report template with the UCT logo. Feel free to use/modify it to suit your needs. Variables that need to be altered have been commented to make modifications easier. For example if you need to change the university logo, look for the comment `% University Logo` in this file and then make appropriate modifications in that line. A Table of Contents and a bibliography have also been implemented. To add entries to your bibliography, simply edit `biblist.bib` in the root folder and then use the `\cite{...}` command in `main.tex` [4]. The Table of Contents will be updated automatically. I hope that you find this template both visually appealing and useful.

— Linus

## 4 Methodology

The methodology can be summarized to the following block diagram.

### 4.1 Sensor

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

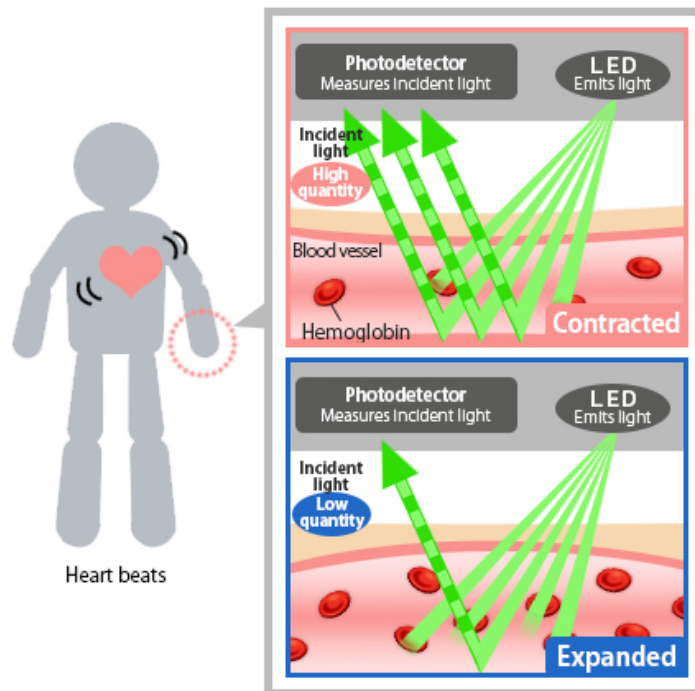


Figure 1: How IR LED light is reflected from the blood vessels

## 4.2 Amplification and Filtering

To amplify the PPG signal we designed an amplifier circuit. However we observed that without proper filtering we cannot get a clear signal due to noise. So we had to determine the necessary values for the filter circuit.

The typical heart rate of an adult is between 60-100bpm[3]. In frequency terms, these values correspond to the range 1-1.7 Hz. So an active bandpass filter circuit was designed to provide the following characteristics using LM358.

### 4.2.1 Calculations

$$\text{Cutoff frequency } (f) = \frac{1}{2\pi RC}$$

$$\text{Lower cutoff frequency} = \frac{1}{2\pi(680 \text{ k}\Omega)(100 \text{ nF})} = 2.34051 \text{ Hz}$$

$$\text{Upper cutoff frequency} = \frac{1}{2\pi(47 \text{ k}\Omega)(4.7 \text{ }\mu\text{F})} = 0.720484 \text{ Hz}$$

So we have,

- Gain of 10
- Lower Cutoff Frequency of 0.72 Hz
- Upper Cutoff Frequency of 2.34 Hz

To have a low powered operational amplifier we selected LM358 integrated circuit.

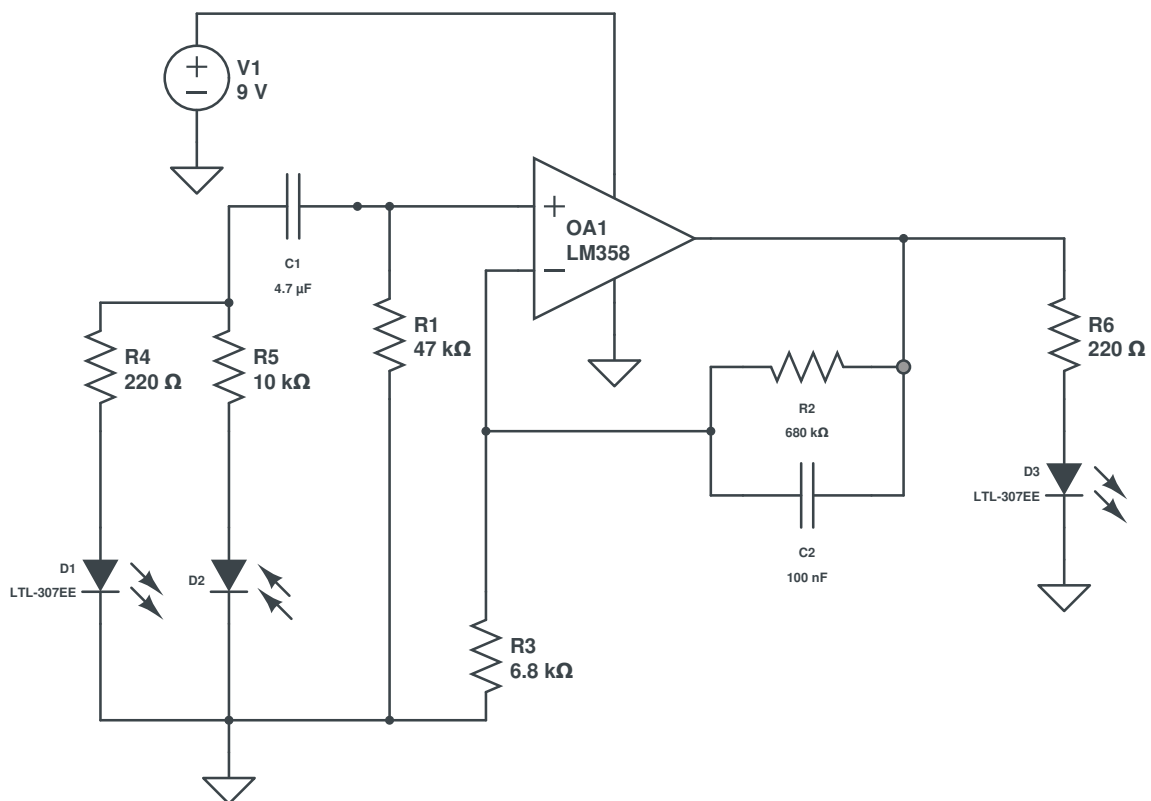


Figure 2: Active bandpass filter for the pulse sensor

### 4.3 Calculating BPM from the analog signal

The ATmega32 microcontroller was used to sample the analog signal to 0-1023 voltage levels. Then an algorithm was developed to identify peaks and thereby calculate the beats per minute. We used Atmel Studio to compile and flash the hex file to the microcontroller.

#### 4.3.1 Microcontroller

Basic info about the microcontroller goes here.

#### 4.3.2 Obtaining a threshold value

Give basic info on how threshold was determined.

#### 4.3.3 Algorithm

can include the flowchat, and code here

```
1      /* Group 7 | UoM | ENTC 17Batch */
2      #ifndef F_CPU
3      #define F_CPU 16000000UL // 16 MHz clock speed
4      #endif
5      #define D0 eS_PORTD0
6      #define D1 eS_PORTD1
7      #define D2 eS_PORTD2
8      #define D3 eS_PORTD3
9      #define D4 eS_PORTD4
10     #define D5 eS_PORTD5
11     #define D6 eS_PORTD6
12     #define D7 eS_PORTD7
13     #define RS eS_PORTC6
14     #define EN eS_PORTC7
15
16     #include <avr/io.h>
17     #include <util/delay.h>
18     #include "stdlib.h"
19     #include "string.h"
20     #include "lcd.h"
21
```



```
22     char disp[16]="0000000000000001";
23     char result[8] = "00000001";
24
25     void lcd_disp(char arr[],int r,int c,char w[]){
26         if(w=="clear")Lcd8_Clear();
27         Lcd8_Set_Cursor(r,c);
28         Lcd8_Write_String(arr);
29     }
30
31     void ADC_Init(){
32         DDRA=0x0;          /* Make ADC port as input */
33         ADCSRA = 0x87;      /* Enable ADC, fr/128 */
34         ADMUX = 0x40;       /* Vref: Avcc, ADC channel: 0 */
35     }
36
37     int ADC_Read(char channel){
38         int Ain,AinLow;
39
40         ADMUX=ADMUX|(channel & 0x0f); /* Set input channel to read */
41
42         ADCSRA |= (1<<ADSC); /* Start conversion */
43         while((ADCSRA&(1<<ADIF))==0); /* Monitor end of conversion
44                                     interrupt */
45         _delay_us(10);
46         AinLow = (int)ADCL; /* Read lower byte*/
47         Ain = (int)ADCH*256; /* Read higher 2 bits and
48                             Multiply with weight */
49         Ain = Ain + AinLow;
50         return(Ain); /* Return digital value*/
51     }
52
53
54     int main(void){
55
56         DDRD = 0xFF; // #
57         DDRC = 0xFF; //for lcd
58         DDRA = 0x00; //Analog input
59
60         ADC_Init();
61
62         Lcd8_Init(); //Initializing the LCD screen
63         lcd_disp("Starting Pulse ~",1,0,"");
```

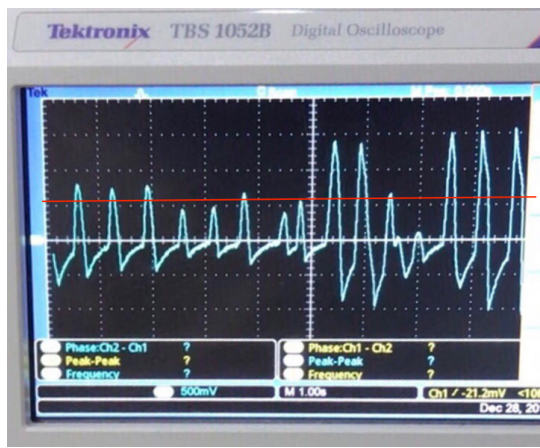
```
64     lcd_disp("Meter",2,0,"");
65
66     _delay_ms(3000);
67     int i;
68
69     for(i=0;i>=0;i++){
70
71         char temp[11]="Analog - ";
72         char val[4]; // 0 - 255 value
73
74         itoa(ADC_Read(0),val,10);
75
76         strcat(temp,val);
77
78         lcd_disp(temp,1,0,"clear");
79         _delay_ms(300);
80
81     }
82 }
```

Based on this [1] and this data-sheet [2]

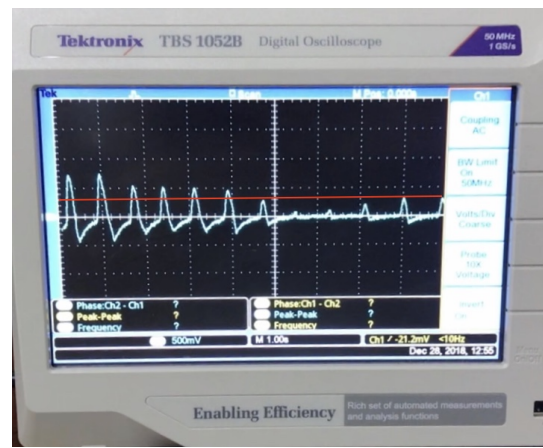
## 4.4 Enclosure and PCB Design

## 5 Implementation and Results

While testing, we observed that due to various lighting conditions, the amplitude of the output waveform varies. Therefore we came to a conclusion that a hard-coded threshold value will not count the number of peaks accurately. Therefore, we implemented a calibration algorithm.



(a) Varying amplitudes



(b) Amplitude Gradually decreasing

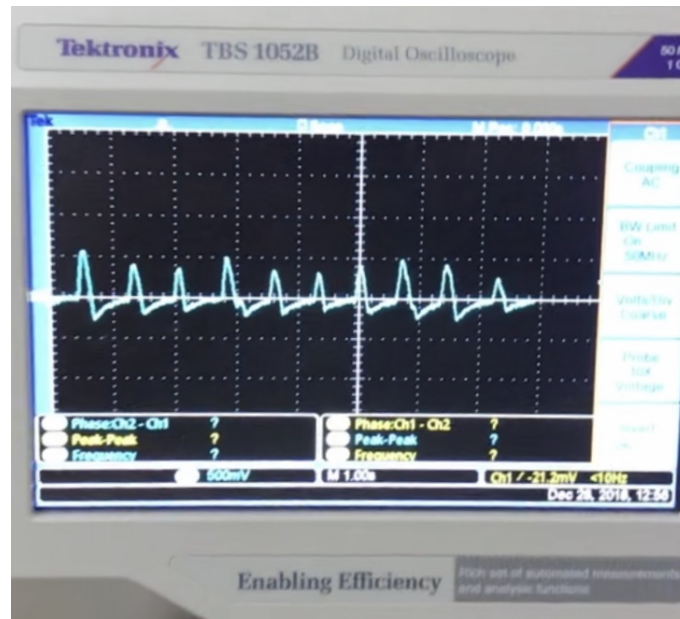
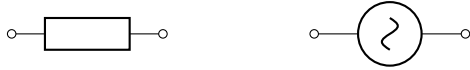


Figure 3: Filtered Waveform




Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

## 6 Conclusion

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

## 7 Online Materials

The project has been open sourced on Github. All the diagrams, schematics, PCB designs can be found through our repository.

 <https://github.com/ramithuh/Pulse-Sensor>.

## References

- [1] Analog readings with avr. <https://maker.pro/custom/tutorial/how-to-take-analog-readings-with-an-avr-microcontroller>. Accessed: 2018-12-28.
- [2] Atmega32 datasheet. <http://ww1.microchip.com/downloads/en/devicedoc/doc2503.pdf>. Accessed: 2018-11-02.
- [3] Harvard heart letter. <https://www.health.harvard.edu/heart-health/want-to-check-your-heart-rate-heres-how>. Accessed: 2018-12-30.
- [4] Photoplethysmogram - wikipedia article. <https://en.wikipedia.org/wiki/Photoplethysmogram>. Accessed: 2018-10-12.