I, Aparna Rakurthi, hereby submit this original work as part of the requirements for the degree of Master of Science in Computer Engineering.

It is entitled:
Development of an Optical Heart Rate Monitor using a Microchip PIC24-microcontroller based development board

Student's name: Aparna Rakurthi

This work and its defense approved by:

Committee chair: Fred Beyette, PhD

Committee member: Wen Ben Jone, PhD

Committee member: Carla Purdy, PhD
Development of an Optical Heart Rate Monitor using a Microchip PIC24-microcontroller based development board

A thesis submitted
to the Graduate School of the University of Cincinnati
in partial fulfillment of the requirements for the degree of

Master of Science

in the School of Electronic and Computing Systems
of the College of Engineering and Applied Science

April 2012

By

Aparna Rakurthi

M.S.(Electrical Engineering),
University of Cincinnati, Cincinnati, 2010

Thesis Advisor and Committee Chair: Dr. Fred R. Beyette Jr.
Abstract

The primary goal of this thesis is to develop a prototype device that can be used to demonstrate some key and fundamental concepts related to the disciplines of computer and electrical engineering to freshman engineering students. This objective is achieved through the development of a heart rate monitor that primarily consists of a sensor module for heart beat detection and a Microchip PIC-microcontroller based development board for heart rate calculation.

The central element of the sensor module is an optical detection system that consists of a light-emitting diode and photodiode setup to detect heart beat from a measurement site with strong pulse like a fingertip. The obtained heart beat signal is then passed through multiple amplification and filtering stages to obtain a clean and strong heart beat signal. Using this optical sensor module implemented on a printed circuit board (PCB) we can demonstrate several concepts related to electrical engineering such as: basic electronics, semiconductor devices, analog circuit design, optical electronics, PCB design, etc.

The voltage signal obtained from the sensor module is sent to a Microchip Explorer 16 development board for further signal processing and heart rate calculation. The development board contains a 16-bit PIC microcontroller with a built in 10-bit analog-to-digital converter that is used to digitize the analog voltage signal and calculate heart rate as beats per minute using a heart rate calculation algorithm. The calculated heart rate is finally displayed on an alphanumeric liquid-crystal display (LCD) display that is included on the development board. This module of the prototype heart rate monitoring system is designed to be interactive and provide user-control to the heart rate calculation process while demonstrating some concepts related to computer
engineering such as: microcontrollers, embedded systems, software development, hardware/software co-design, etc.

The heart rate monitor developed in this research work has thus been designed to present a prototype device that can be used to demonstrate several key concepts related to computer and electrical engineering and this thesis document contains design and implementation details related to the development of this device.
Acknowledgements

I would like to express my sincere gratitude to my advisor Dr. Fred Beyette for his guidance and constant encouragement through the course of this research work. I would like to particularly acknowledge and thank him for his kind cooperation and support during some critical phases of my graduate studies and helping us achieve my academic goals. I would also like to thank Dr. Wen-Ben Jone and Dr. Carla Purdy for taking the time to read and comment on my thesis and for serving on my defense committee.

Thanks to all my friends in Cincinnati for their friendship and support. Special thanks to Bharath, Deepika, Harika, Himaja, Raja, Sandeep, Sudhir, Swetha and Vikram who have been my family here in the US. Also, thanks to all my friends in POCSDL for all the lively discussions and fun during the course of this research work.

Finally, I dedicate this thesis to my family. I am what I am and achieved what I have to date because of their constant love and support. Thank you!
# Table of Contents

Chapter 1 – Introduction ................................................................................................................. 1

1.1 Background information ............................................................................................................. 1

1.2 Methods of heart beat detection ................................................................................................. 2

1.2.1 Electrodes method .................................................................................................................. 3

1.2.2 Optical method ....................................................................................................................... 3

1.3 Heart rate monitor model ......................................................................................................... 5

1.4 Research goals .......................................................................................................................... 6

1.5 Thesis outline ........................................................................................................................... 7

Chapter 2 – Optical Sensor Module ............................................................................................... 8

2.1 Overview of the sensor module ................................................................................................. 8

2.2 Sensor circuit design ............................................................................................................... 9

2.2.1 LED setup ............................................................................................................................ 9

2.2.2 Photodiode detector module ................................................................................................. 10

2.2.3 AC-coupled inverting amplifier ......................................................................................... 12

2.2.4 Band-pass filter .................................................................................................................. 16

2.2.5 Non-inverting amplifier ...................................................................................................... 18

2.2.6 Low-pass filter .................................................................................................................... 18

2.2.7 Biased-positive clipper ........................................................................................................ 19

2.2.8 ADC capacitor charging setup ............................................................................................ 20
2.3 PCB layout ........................................................................................................................................... 23

Chapter 3 – Explorer 16 Development Board......................................................................................... 25

3.1 Overview of the Explorer 16 development board........................................................................... 25

3.2 Software development ....................................................................................................................... 28

3.2.1 Heart rate calculation algorithm ................................................................................................... 29

3.2.2 Flowchart ...................................................................................................................................... 30

3.3 PIC24FJ128GA010 microcontroller ................................................................................................. 34

3.3.1 I/O Ports ...................................................................................................................................... 35

3.3.2 Oscillators .................................................................................................................................... 37

3.3.3 Timers ......................................................................................................................................... 39

3.3.4 Interrupts ...................................................................................................................................... 43

3.3.5 ADC Converter ............................................................................................................................. 46

3.3.6 Parallel Master Port ...................................................................................................................... 56

Chapter 4 – Results ................................................................................................................................. 60

4.1 Test Setup ....................................................................................................................................... 60

4.2 Results and Discussion: .................................................................................................................. 61

Chapter5 – Conclusion and Future Work ............................................................................................. 67

5.1 Conclusion ...................................................................................................................................... 67

5.2 Suggestions for future work ........................................................................................................... 68

References ............................................................................................................................................. 70
List of Figures

Figure 1. ECG waveform showing the QRS complex .......................................................... 2
Figure 2. Functional block diagram ................................................................................. 5
Figure 3. Microchip Explorer 16 development board ....................................................... 6
Figure 4. Block diagram of the optical sensor circuit ....................................................... 8
Figure 5. LED setup ......................................................................................................... 9
Figure 6. Photodiode detector module ............................................................................. 11
Figure 7. Light absorption through living tissue .............................................................. 12
Figure 8. AC-coupled, biased, inverting amplifier .......................................................... 13
Figure 9. Non-inverting amplifier .................................................................................... 14
Figure 10. Inverting amplifier ........................................................................................ 14
Figure 11. Band-pass filter ............................................................................................. 17
Figure 12. Inverting amplifier ........................................................................................ 18
Figure 13. Low-pass filter .............................................................................................. 19
Figure 14. Positive, biased clipper circuit ..................................................................... 19
Figure 15. Voltage regulator circuit ................................................................................ 20
Figure 16. ADC analog input model .............................................................................. 21
Figure 17. ADC capacitor charging setup ..................................................................... 21
Figure 18. Sensor circuit and development board interface setup .................................. 22
Figure 19. PCB layout of the optical sensor circuit ......................................................... 23
Figure 20. PCB assembly of the optical sensor circuit .................................................... 24
Figure 21. High-level block diagram of the Explorer 16 development board ............... 25
Figure 22. Explorer 16 development board layout ......................................................... 27
Figure 23. Flow of control during heart rate calculation ........................................ 28
Figure 24. Graphical representation of the heart rate calculation algorithm .................. 30
Figure 25. Heart rate calculation flowchart ................................................................. 32
Figure 26. PIC24FJ128GA family clock diagram ......................................................... 38
Figure 27. Timer control register .............................................................................. 41
Figure 28. Implemented interrupt vectors ................................................................... 44
Figure 29. ADC block diagram .................................................................................... 47
Figure 30. A/D control register 1 ................................................................................ 50
Figure 31. A/D control register 2 ................................................................................ 51
Figure 32. A/D control register 3 ................................................................................ 52
Figure 33. ADC sample/conversion sequence .............................................................. 53
Figure 34. ADC transfer function .............................................................................. 55
Figure 35. PMP module ............................................................................................... 57
Figure 36. PMP controlled LCD module ..................................................................... 58
Figure 37. Test setup .................................................................................................... 61
Figure 38. Test case 1- Individual 1 before exercise ...................................................... 63
Figure 39. Test case 2- Individual 1 after exercise ........................................................ 63
Figure 40. Test case 3- Individual 2 before exercise ...................................................... 64
Figure 41. Test case 4- Individual 2 after exercise ........................................................ 64
Figure 42. Erroneous peak detection due to hand or finger movement ....................... 66
Chapter 1 – Introduction

As the title suggests, a prototype heart rate monitor has been developed that could be used to demonstrate several fundamental concepts related to the fields of electrical and computer engineering to freshman engineering students. The heart rate monitor developed in this project is designed to be interactive and can be used to introduce the students to different areas in electrical and computer engineering such as analog and digital electronics, semiconductor devices, embedded systems, biomedical electronic devices, hardware/software co-design, etc.

Before discussing the details related to the design and implementation of this prototype device, some background information pertaining to heart beat detection is provided next.

1.1 Background information

Heart rate, which is typically expressed as beats per minute (BPM), is the number of heart beats per unit time. All events related to blood flow or pressure between the beginning of one heart beat and the beginning of the next are referred to using the term cardiac cycle\(^1\). A cardiac cycle involves five major stages, where the first stage involves the flow of blood from atria in the heart to the ventricles, the next three stages involve the depolarization of the ventricles and the last stage involves the repolarization of the ventricles. This cardiac activity is clearly represented by the electrocardiogram (ECG) tracing shown in Figure 1. An ECG records the electrical activity of the heart and, as shown in the figure, is typically composed of five deflections named P, Q, R, S and T. The P wave represents the atrial contractions, the QRS complex represents the ventricular contractions while the T wave represents the ventricular repolarization for the next contraction\(^2\). In this project, the heart rate is calculated by measuring the time interval between
the R peaks of two consecutive cardiac cycles. This time interval is averaged over multiple cycles and is used to calculate BPM using a microcontroller. A brief description of different methods available to detect heart beat is provided in section 1.2. The heart rate monitor developed in this thesis consists of a sensor circuit to detect heart beat, an analog-to-digital converter (ADC) for signal digitization, a 16-bit PIC24FJ128GA010 microcontroller for data processing and an LCD to display the calculated heart rate as BPM. Further details related to the design and implementation of this prototype device are discussed in section 1.3 and chapters 2 and 3.

1.2 Methods of heart beat detection

Heart beat can be monitored using different methods: acoustically using a stethoscope, electrically using an ECG, mechanically using a sphygmomanometer or optically. In this section, we take a closer look at two of the above listed methods to detect heart beat. The most popular
method is the electrical method, where an electrode system is employed to study the electrical activity of the heart and thus monitor heart beat continuously. The second method is the optical method where an optical sensor circuit is used to detect heart beat by studying the changes in the absorption of light in the blood flow with every heart contraction. Details related to both these methods are provided next.

1.2.1 Electrodes method

The electrode or electrical heart beat detection system is based on detecting the electrochemical impulses generated by the heart with the polarization and depolarization of the cells of the heart. This electrical activity combined with the conductive fluid in the body results in an electrochemical action that can be measured at the surface of the body using an electrode system. By placing electrode contacts at various points on the body, voltage potential can be measured to study the cardiac activity. The chest wall and four limbs have been identified as standard sites for electrode placement to study cardiac activity in different parts of the heart.

Electrocardiography, the best example for an electrode-based heart rate monitoring system, is defined as “the graphical display of electrical potential differences of an electrical field originating in the heart as recorded at the body surface.” Other examples of the electrode-based heart beat monitoring systems include commercial heart rate monitors which consist of a chest strap with electrodes and wrist receiver, exercise machines with built-in heart rate monitors, etc.

1.2.2 Optical method

The optical method for heart beat monitoring is based on the principle of photoplethysmography, which states that the intensity of visible light passing through body tissue is modulated based on the blood content. When light is directed onto a particular site with
strong pulse during a heartbeat cycle, vascular changes in the tissue bed due to arterial pulsations
during a cardiac cycle will result in a variation of the intensity of the measured light\textsuperscript{5,6}. Pulse
oximetry, a non-invasive method used to measure and monitor arterial oxygen saturation, is
another popular technique based on photoelectric photoplethysmography\textsuperscript{7}. Light sources in the
near red (600-700nm) wavelength region are effective light sources for heart beat monitoring
since hemoglobin, which is the major protein present in blood, is a strong absorber of light at that
wavelength\textsuperscript{8}. Thus, changes in the amount of oxygenated hemoglobin in blood during a cardiac
cycle will result in a distinct change in light intensity that can be studied to detect heartbeat.

There are two methods in which light can be directed through the site at which the
measurement is to be taken: by transmission or reflectance\textsuperscript{9,10}. In the transmission method, the
measurement site like a fingertip is placed in between a light source and a detector and the light
transmitted through the finger is monitored for intensity variations. In the reflection mode, the
light source and detector are placed next to each other such that the light source is directed onto
the measurement site and the detector is used to observe the variations in the reflected light to
monitor cardiac activity. While the transmission mode is the popular method, it imposes more
restrictions on the number of measurement sites suitable for studying heart beat than the
reflection method. For both methods, care should be taken with the light source and detector
setup so as to minimize artifacts associated with probe movement and avoid artifacts due to
ambient light interference, etc.

The optical method of heart beat monitoring presents certain advantages over the electrode
or electrical method. First, unlike the electrode method there is no electrical contact to the body
in the optical method, making it safer. Second, there is less electromagnetic interference to the
obtained signal, which is a significant issue in the electrode method. The human body acts as a
huge antenna that picks up a lot of noise, including a dominant 50-Hz/60-Hz line-frequency noise. With a contact method like the electrode system, these power-line signals add significant noise to heart beat signal\textsuperscript{11}. In this thesis, an optical sensor system has been developed based on the above discussed photoplethysmography principle and reflection method to detect heartbeat.

1.3 Heart rate monitor model

In this section we present a model of the heart rate monitor developed in this project, as shown in the functional block diagram presented in Figure 2. The prototype heart rate monitor consists of two main parts: 1) an optical sensor circuit for heart beat detection and 2) a Microchip Explorer 16 development board used for signal processing, heart rate calculation and output display.

![Functional block diagram](image)

**Figure 2. Functional block diagram**

The optical sensor circuit designed in this thesis primarily consists of a light-emitting diode (LED) and photodiode setup based on the reflection-mode, photoplethysmography principle to obtain heart beat signal from a fingertip. The obtained signal is passed through multiple amplifier and noise-filtering stages to obtain a clean signal with distinct peaks that represent heart beats. A Microchip Explorer 16 development board, shown in Figure 3\textsuperscript{12}, has been used in this project to process the signal obtained from the optical sensor circuit and obtain heart rate as BPM\textsuperscript{13}. A 16-
bit PIC24FJ128GA010 flash microcontroller is available on the board with a 10-bit ADC that is used for signal digitization and heart rate calculation\textsuperscript{14}. Finally, the calculated BPM is displayed on a 2 x 16 alphanumeric LCD module available on the development board. Detailed discussions on each module shown in the functional diagram are presented in chapters 2 and 3.

Figure 3. Microchip Explorer 16 development board

1.4 Research goals

As discussed in the beginning of this chapter, the goal of this thesis work is to design and develop a heart rate monitoring system that is interactive and can be used to introduce some key
concepts related to electrical and computer engineering to freshman engineering students. The optical sensor module in the prototype can be used to demonstrate various concepts related to analog electronics, semiconductor devices and optical electronics, while the microcontroller development board module can be used to introduce concepts related to digital electronics, microcontrollers, embedded systems, hardware and software co-design, etc.

1.5 Thesis outline

The rest of this thesis documentation has been organized as follows:

Chapter 2 – Optical Sensor Module, describes various design and implementation details related to the development of the optical sensor circuit including PCB design.

Chapter 3 – Explorer 16 Development Board, discusses operational and implementation details related to several key modules available on the board including software development.

Chapter 4 – Results, contains details related to the test setup and experimental validation of the prototype device through different test cases.

Chapter 5 – Conclusion and Future work, concludes this documentation of the prototype device development with some suggestions for future work.
Chapter 2 – Optical Sensor Module

2.1 Overview of the sensor module

In this chapter, design and implementation details related to the optical sensor circuit are discussed in detail. Shown in Figure 4, is a functional block diagram of the optical sensor circuit designed in this thesis. As shown in the figure, first an LED and photodiode setup is used to obtain a heartbeat signal from the fingertip of a person. The obtained signal is given to an AC-coupled inverting amplifier module to amplify only the voltage fluctuations in the signal, caused due to cardiac activity during a heartbeat. Following this amplification stage, the signal is passed through a band-pass filter stage to filter out unwanted frequencies and noise in the amplified signal. The filtered signal is then passed to another amplifier stage to obtain the signal in a voltage range that is within the electrical ratings of the PIC24 microcontroller to which the signal is to be sent for further signal processing. Following this second stage of voltage amplification, the signal is passed through a low-pass filter to filter out any noise that is injected during amplification. The analog I/O pins of the PIC24 microcontroller have a maximum voltage rating of approximately 3.3V. A biased, positive clipper circuit has been added in the sensor module following the low-pass filter to ensure that the voltage of the signal sent to the Explorer 16 development board is below 3.3V at all times. Finally, before the signal is sent to the ADC.

Figure 4. Block diagram of the optical sensor circuit
module of the PIC24 microcontroller on the development board, it is passed through a capacitor charging setup to supply the current needed to charge the sample/holding capacitor in the ADC module for accurate signal sampling\textsuperscript{15}. Details pertaining to the design of each of these blocks are presented in the following section.

\subsection*{2.2 Sensor circuit design}

\subsubsection*{2.2.1 LED setup}

The circuit shown in Figure 5 depicts the light source setup that is used to direct light on a highly vascular surface like a fingertip. Together with the photodiode detector module, the LED-photodiode setup is used to detect heartbeat from the pulsatile blood flow based on the photoplethysmography principle.

![Figure 5. LED setup](image)

A super red, 3mm diameter LED with a peak wavelength of 660nm has been chosen since light near this wavelength has been established to be best suited for heartbeat detection through blood flow, as previously discussed in section 1.2.2 in chapter 1. An LED functions like a normal
diode where the current flowing through the diode increases exponentially with applied voltage\textsuperscript{16}. Thus, in case of application of high voltages, the LED can be potentially destroyed due to high currents that are over the maximum rating for that LED. To solve this problem, generally a current limiting resistor is added in series with an LED to control the LED current. The formula to calculate the resistance value for the resistor is given by\textsuperscript{17}:

\[
\text{Resistance} = \frac{(\text{Supply voltage} - \text{LED forward voltage drop})}{\text{LED current}}
\] (1)

The red LED used in this project has a typical forward voltage drop of 1.8V and desired current of 20mA. Since the sensor circuit board designed in this thesis is powered by 9V power supply, the resistance is calculated to be 360\textOmega. Based on these calculations, a current limiting resistor of 360\textOmega has been used in series with the LED, as shown in Figure 5. Increasing or decreasing the series resistance, decreases or increases the intensity of light emitted from the LED accordingly.

\textbf{2.2.2 Photodiode detector module}

The photodiode detector circuit implemented in this thesis is shown below in Figure 6. A photodiode is a diode that converts light incident on it into current flowing through the diode\textsuperscript{18}. The photodiode together with the LED is used in the reflection-based, photoplethysmography method discussed in section 1.2.2 in chapter 1, to detect cardiac activity based on the light reflected by the pulsatile blood flow at the heartbeat measurement site.
A PIN photodiode with a 2.65 x 2.65 mm$^2$ radiant sensitive area and spectral range of sensitivity between 400-1100nm, which includes the 660nm wavelength of our red light source, has been used in this module. PIN photodiodes expand the response range of a normal photodiode by the inclusion of an intrinsic layer between the normal P and N regions of a diode. While the output signal from a photodiode can be studied both as current or voltage signal, a current signal is generally used since it has been seen to offer better linearity, offset and bandwidth performance.

Since we need a voltage signal for further signal processing, the current signal obtained from the photodiode is converted into a voltage signal using a current-to-voltage (I-to-V) converter circuit, as shown in Figure 6. When light is incident on the virtually shorted photodiode, the induced current is forced through the feedback resistor and a voltage signal is obtained at the output. The operational amplifier(op-amp) in the circuit tries to keep the input pins at the same voltage independent of the incident light and hence keep the photodiode zero-biased. This setup

Figure 6. Photodiode detector module
of zero-bias and high-impedance loading to the photodiode has been observed to result in reduced influence by dark current and a broad linear range of photocurrent in relation to the intensity of incident light\textsuperscript{20}. A 30\text{K}\Omega resistor has been used as a feedback resistor as shown in Figure 6. The output voltage, which depends on the value of the feedback resistor used in the circuit, is given by the following formula:

\[ V_o = I_f \times R \quad (2) \]

2.2.3 AC-coupled inverting amplifier

The voltage signal obtained from the above LED-photodiode module has to be amplified and filtered before further signal processing. The signal obtained from the photodiode I-to-V setup has a small AC component (due to variation in the reflected light intensity based on changes in pulsatile blood flow) and a comparatively large DC component (due to constant reflected light intensity from the non-pulsatile blood flow, absorption through the finger tissue, etc\textsuperscript{21}, as shown in Figure 7.

![Figure 7. Light absorption through living tissue\textsuperscript{21}](image.png)
Since only the alternating component of the signal is of interest to us, the signal is passed through a capacitor $C_{\text{block}}$ in series with the amplifier circuit as shown in Figure 8. This process is called AC coupling or DC blocking where only the AC or time-varying part of the signal is passed through to the amplifier while the DC component is suppressed. The coupling capacitor together with the input impedance of the op-amp will act as a high-pass filter, degrading our low frequency heart beat signals which can have a maximum frequency of approximately 4Hz (250 beats per minute). To overcome this signal degradation issue and obtain adequate low frequency response, a high capacitance of 470µF is used for the $C_{\text{block}}$ capacitor to overcome forming a high-pass filter at the op-amp input terminal. The 470µF capacitor along with the 56Ω R1 resistor offers a cut-off frequency that is adequate for the signal requirements in this project. The cut-off frequency is calculated using the formula:

$$f_c = \frac{1}{2\pi RC} \quad (3)$$
The DC-blocked signal is next passed to a biased, inverting amplifier as shown in Figure 8. An op-amp with negative feedback can be used as an amplifier with gain characteristics that are dependent on the feedback circuit used in the circuit design\(^2^3\). The negative feedback-based amplifier can be operated in two primary configurations: as a non-inverting amplifier or as an inverting amplifier.

![Non-inverting amplifier diagram](image1)

**Figure 9. Non-inverting amplifier**

![Inverting amplifier diagram](image2)

**Figure 10. Inverting amplifier**

As shown in Figure 9, in a non-inverting amplifier configuration, the input signal is given to the non-inverting terminal of the op-amp while the inverting terminal is connected to the output terminal using a feedback resistor \( R_f \). The resistors \( R_f \) and \( R_g \) form a voltage divider circuit, and the voltage drop that appears across \( R_g \) is given as input to the inverting terminal of the op-amp\(^2^2\). The difference between the voltages at both the terminals is then amplified by the open-loop gain of the circuit. The voltage divider rule can be used to calculate the gain as follows\(^2^4\):

\[
V_i = V_o \times \frac{R_g}{R_f + R_g}
\]

\[
Gain = \frac{V_o}{V_i} = 1 + \frac{R_f}{R_g} \quad (4)
\]
An inverting amplifier, as the name suggests, gives an amplified and inverted or 180° phase shifted signal as output. As shown in Figure 10, the input signal is given to the inverting terminal of the op-amp while the non-inverting terminal is grounded. Negative feedback is setup by connecting the inverting input terminal to the output terminal using a feedback $R_f$ resistor. Since the non-inverting amplifier is grounded, the feedback network inverts the input signal to a virtual ground\textsuperscript{24}. The current flow in the input terminals is assumed to be zero and hence current through resistors $R_g$ and $R_f$ are equal. Using Kirchoff's law:

$$I_g = -I_f$$

$$\frac{V_i}{R_g} = -\frac{V_0}{R_f}$$

$$Gain = \frac{V_0}{V_i} = -\frac{R_f}{R_g}$$  \hspace{1cm} (5)$$

Shown in Figure 8 is the biased, inverting amplifier circuit that has been implemented for signal amplification. Since all signal voltages in this system are required to be within the 3.3V rating of the PIC24 microcontroller at all times, the amplifier is biased to provide a steady-state output voltage of 2V. Once the AC input signal is applied, based on the gain of the setup the output will oscillate about the bias point of 2V. We calculate the resistances used for this bias setup, using the following formula:

$$V_{bias} = \frac{R_4}{R_3 + R_4} * Supply\ voltage$$  \hspace{1cm} (6)$$

Since we have a supply voltage of 9V and need a bias voltage of 2V, the circuit is designed to have $R3 = 3.5K\Omega$ and $R4 = 1K\Omega$ using equation 6. Further, this voltage divider biasing setup
along with the $470\mu F$ $C_{bias}$ capacitor gives a cut-off frequency much lesser than the lowest cut-off frequency requirements of the whole setup, ensuring that the impedance of the bias network will not add significantly at the desired frequencies. Finally, the gain obtained from the inverting amplifier is determined by resistors R1 and R2 as derived previously in equation 5 and is given by the following formula:

$$Gain = -\frac{R_2}{R_1}$$

As shown in Figure 8, we have used a 40KΩ resistor as the feedback resistor R2 in the circuit with the 56Ω R1 resistor. Along with providing the necessary signal amplification, this 40KΩ feedback resistance is small enough to avoid “motorboat” oscillations that were observed in the output signal from the AC-couple op-amp circuit implemented on the PCB, due to the introduction of parasitic capacitances and leading phase shifts in the signal$^{24}$.

2.2.4 Band-pass filter

The signal obtained from the amplifier stage needs to be filtered since it may consist of noise from several external and internal sources. It may contain unwanted noise introduced into the signal due to electrical pick-up from the 60Hz power lines or low frequency variations induced into the signal due to respiratory activity, tremor or involuntary finger movements (motion artifacts)$^9$. To filter out these undesired frequencies, a band-pass filter has been implemented after signal amplification.

Since, the heartbeat signal is expected to be between 1Hz to 4Hz, the filter has been designed to filter out frequencies approximately lower than 1Hz and higher than 4Hz. As shown in Figure 11, the band-pass filter implemented in this thesis consists of a low-pass filter followed
by a high-pass filter. The low-pass RC filter section has a cut-off frequency of approximately 0.5Hz and the high-pass RC filter has a cut-off frequency of approximately 5Hz, where the cut-off frequency is calculated using equation 3. Thus, signals with frequencies below 0.5Hz and above 5Hz are suppressed using this filter.

One issue with the RC band-pass filtering technique implemented in this circuit is that there is some signal attenuation in the pass band frequency range due to the cascading of low-pass and high-pass passive RC filters\textsuperscript{25}. This loss (in dB) is given by the following equation, where $w$ is the frequency and $n$ is the order of the filter\textsuperscript{26}:

\[
\text{Loss}(dB) = -20 \log_{10}(w) \quad (7)
\]

Hence, the filtered signal obtained from the band-pass filter stage has some amplitude loss and needs to be amplified before being sent to the ADC module for further signal processing.

![Figure 11. Band-pass filter](image-url)
2.2.5 Non-inverting amplifier

The filtered and attenuated signal obtained from the band-pass filter is amplified using a non-inverting amplifier stage. Based on the discussion in section 2.2.3 regarding non-inverting amplifier configuration, the amplifier circuit used for further signal amplification is shown in Figure 12. Gain of this circuit can be calculated using equation 4.

![Inverting amplifier circuit diagram](image)

**Figure 12. Inverting amplifier**

2.2.6 Low-pass filter

Following the second amplification stage, the amplified signal obtained from the opamp-based amplifier circuit has been observed to contain some noise peaks introduced into the signal when observed on an oscilloscope. Since we need a clean signal with distinct peaks for heat beat detection, an additional low-pass filter has been included after the amplification stage. The RC low-pass filter, as shown in Figure 13, has a cut-off frequency of 6Hz and suppresses any noise frequencies above that. The cut-off frequency is again calculated using equation 3.
As previously stated, the PIC24 microcontroller has a maximum voltage rating of approximately 3.3V for any analog I/O pin\textsuperscript{14}. To ensure that the voltage signal being sent to the ADC module of the microcontroller is below 3.3V at all times, a biased, positive clipper circuit has been included in the design. Shown in Figure 14 is the biased, shunt positive clipper circuit implemented in this project that clips any signal voltage above an approximate cut-off voltage of 2.8V - 3V.
When the voltage of the input signal given to the clipper circuit is less than the bias voltage, the diode remains OFF and the input signal is observed at the output of the clipper circuit. But when the voltage of the input signal is more than the bias voltage, the diode turns ON and a constant voltage that is equal to the bias voltage plus the threshold voltage of the diode is obtained at the output of the clipper circuit\(^2\), \(2\). Thus, using this biased, positive clipper circuit we ensure that the voltage signal being sent to the development board is within the maximum voltage rating. A 2.5V bias voltage is provided using a 2.5V voltage regulator setup as shown in the Figure 15.

![Voltage regulator circuit](image)

**Figure 15. Voltage regulator circuit**

### 2.2.8 ADC capacitor charging setup

Finally, before sending the amplified and filtered voltage signal to the ADC module to be digitized, the signal is passed through an ADC capacitor charging setup that is necessary for accurate ADC sampling. Shown in Figure 16 is the analog input model of the 10-bit ADC module that primarily consists of \(C_{\text{HOLD}}\) (Sample/Hold Capacitance), \(C_{\text{PIN}}\) (input capacitance), \(R_{\text{IC}}\) (Interconnect Resistance), \(R_{\text{SS}}\) (Sampling Switch Resistance).
For the ADC module to operate at the desired accuracy, the charge holding capacitor $C_{\text{HOLD}}$ needs to be fully charged to the voltage level on the analog input pin. The source impedance ($R_s$), $R_{\text{IC}}$ and $R_{\text{SS}}$ together affect the time required to charge $C_{\text{HOLD}}$ and hence the impedance of the analog source needs to be small enough to completely charge the holding capacitor within the chosen sample time. Often the source impedance of the input analog signal is high, resulting in pin leakage currents that affect the accuracy of the ADC module. The maximum recommended source impedance is 2.5K$\Omega$\textsuperscript{15}. For high impedance sources where the input signal changes slowly, it is recommended to use a charge capacitor before the analog

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{adc_capacitor_setup.png}
\caption{Figure 17. ADC capacitor charging setup}
\end{figure}
input pin, so that the capacitor charges to the analog voltage being sampled and supplies the instantaneous current required to charge the internal holding capacitor $C_{\text{HOLD}}$. Based on this recommendation, an external charge capacitor setup has been included to ensure proper operation of the ADC module, as shown in Figure 17.

The signal obtained from this final charge block of the sensor circuit is sent to channel AN3 of the ADC module through the PICtail™ Plus connector slot on the development board using a setup of jumper wires and a prototype daughter board that is plugged into the PICtail™ Plus slot, as shown in Figure 18, for stable board to board connection.

Figure 18. Sensor circuit and development board interface setup
2.3 PCB layout

Shown in Figure 19 is the PCB layout of the sensor circuit we have discussed in this chapter. The PCB has been designed using EAGLE layout editor\textsuperscript{28} and fabricated by the PCB fabrication house Advanced Circuits\textsuperscript{29}. Key features of the designed 90 mm x 80 mm PCB include:

- 15mil trace widths for interconnections.
- Minimum 15mil spacing between traces, vias, drill holes and components.

Figure 19. PCB layout of the optical sensor circuit
- Two-layered design:
  - Top copper layer is shown in RED and bottom copper layer is shown in BLUE in Figure 19.
  - Top silk layer, shown in YELLOW in Figure 19, is used to indicate component outlines and related details.
- Ground plane to reduce noise and provide a common reference ground potential to all the circuits in the design.

Shown in Figure 20 is the fabricated sensor board with the all the components populated onto the PCB.

**Figure 20. PCB assembly of the optical sensor circuit**
Chapter 3 – Explorer 16 Development Board

3.1 Overview of the Explorer 16 development board

The voltage heartbeat signal obtained from the optical sensor circuit is next processed using various modules provided by the Microchip Explorer 16 development board to calculate the heart rate. Shown in Figure 21 is a high-level block diagram of the Explorer 16 board with the modules used in this thesis high-lighted in red. The key element we use on the board is the 16-bit PIC24FJ128GA010 microcontroller which has an in-built 10-bit ADC for signal processing and heart rate calculation. The 16 x 2 LCD display is used to display the calculated heart rate in beats.

Figure 21. High-level block diagram of the Explorer 16 development board

13
per minute (BPM). Also shown in the figure is the PICtail™ Plus connector slot that is used to connect the optical sensor circuit to the development board via a Prototype PICtail™ Plus daughter board. Further, one of the switches/pushbuttons available on the board is used to provide user-control to the heart rate calculation. The development board kit includes a Microchip MPLAB ICD 3 programmer/in-circuit debugger(ICD) module that is used in conjunction with the Microchip MPLAB IDE (Integrated Development Environment) to download the code needed to calculate heart rate into the PIC24 microcontroller. Microchip’s MPLAB C30 C Compiler has been used for all the software development in this project. The compiler generates a HEX file from the source code written in C and the ICD 3 module is used to program the generated HEX code into the flash memory of the microcontroller.

Shown in Figure 22 is a more comprehensive layout of the Explorer 16 development board. Again, the features used in this project are highlighted in red and their corresponding descriptions are provided next:

1. **PIC24FJ128GA010 Processor Installation Module (PIM)** – a 16-bit, 100-pin microcontroller mounted on a 100-pin PIM riser on the development board. More details related to the PIC24 microcontroller are provided in section 3.3.

2. **9V DC direct power supply** – Provides +3.3V and +5V regulated voltage to all the modules on the development board.

3. **Power LED** – Indicates if the board is powered.

7. **ICD connector** – Standard 6-wire ICD connector to connect the MPLAB ICD 3 programmer/debugger module to the board.
9. **LCD display** – The 2-line by 16-character alphanumeric LCD display module.

11. **Push Buttons** – For device RESET and supplying user-defined inputs to the development board. One push button for RESET and four general-purpose buttons are available.

13. **Indicator LEDs** – Eight indicator LEDs available, which are used to indicate the completion of heart rate calculation.

18. **PICtail™ Plus connector** – Socket and edge connector slots available for PICtail™ Plus expansion card compatibility. The socket connector slot has been used in this project to connect the sensor circuit to the development board via a prototype PICtail™ Plus daughter board.

---

*Figure 22. Explorer 16 development board layout*\(^\text{13}\)
3.2 Software development

Before discussing the details related to some of the important modules on the development board that have been used in this project, we present some information pertaining to the flow of operational control and software development details related to heart rate calculation in this section.

![Flow of control during heart rate calculation](image)

Figure 23. Flow of control during heart rate calculation

Shown in Figure 23 is a block diagram depicting the flow of operation of the different modules on the development board during heart rate calculation. First, the signal obtained from the optical sensor circuit has to be sent to an ADC module to be digitized. Thus, the voltage signal from sensor circuit is sent to the 10-bit, built in ADC that is available with the PIC24 microcontroller on the development board. The sampled and digitized voltage readings of the heartbeat signal are then used to calculate the heart rate in real-time using the PIC24 microcontroller. The algorithm developed and implemented to calculate heart rate in this project is discussed in detail in sections 3.2.1 and 3.2.2. Finally, the calculated heart rate is displayed in BPM on the 2x16 LCD available on the development board.
3.2.1 Heart rate calculation algorithm

A number of heart rate calculation techniques have been presented in literature that vary in computational complexity and generally involve QRS wave detection using techniques in the time domain or spectral domain or both. As mentioned in section 1.1 in chapter 1, in this thesis the heart rate is estimated by measuring the time interval between the R peaks in the QRS complex of two consecutive cardiac cycles. The algorithm developed to perform this calculation involves three main parts which will be described with the help of Figure 24 as follows:

1. As the voltage signal from the sensor circuit is sampled by the 10-bit ADC module, a running average is maintained of all the converted voltage readings to obtain an average voltage level against which the sampled voltages are compared and an R peak is identified. This average level is depicted by the dashed line in Figure 24.

2. For the detection of the R peak in a single cardiac cycle, another 8-reading average is maintained using the ADC sampled and converted voltage readings. In this part of the algorithm, an average of eight consecutive ADC sampled values is maintained. Whenever the 8-reading average is greater than 3 times the running average value, an approximate detection of an R peak is made. This process of 8-reading average and peak detection is depicted graphically in Figure 24.

3. For the purpose of accuracy, in this project we detect six R peaks over six cardiac cycles and the time taken between the detection of the first and last peak is divided by 5 to get an average estimate of the time interval between two consecutive R peaks. Beats per minute (BPM) is then calculated using the following formula:

\[
BPM = \frac{60}{Time \ between \ two \ peaks}
\]
3.2.2 Flowchart

In this section a flowchart is presented that depicts all the events that occur during signal processing and heart rate calculation using different modules on the development board. The flowchart is shown in Figure 25 and all the steps are explained next:

Step 1: On programming the microcontroller, various registers and ports utilized to access and operate different modules on the board are initialized. Some of the key initializations are given next:

i. **ADC Initialization**: The in-built ADC module available with the PIC24 microcontroller is setup for operation by initialization of various control and configuration registers dedicated for ADC operation. Detailed discussion related to the ADC module operation is presented in section 3.3.5 in this chapter.
ii. **LCD Initialization**: The LCD module present on the development board is driven by a Parallel Master Port (PMP). The LCD and PMP modules need to be initialized using associated special function registers. More details related to both the modules are presented in section 3.3.6.

iii. **Timer Initialization**: Two 16-bit timers, Timer1 and Timer2, are used in the code for heart rate calculation and their control registers are initialized in this step. Details related to Timer operation and control are presented in section 3.3.3.

iv. **External Interrupt Initialization**: The PIC24 offers up to five external interrupt sources for user-defined interrupts. In this project, external interrupt 0 is used to generate an interrupt on a push of the button S3. Flag and enable bits related to this interrupt 0 are initialized here. Information related to the usage of interrupts, interrupt service routines and other implementation details are presented in section 3.3.4.

v. **Port Initializations**: In the 100-pin PIC24 microcontroller, PORTB is used to access the ADC module, PORTD is used to access the push-buttons and PORTA is used to access the eight indicator LEDs on the development board. Details related to the access and control of these ports and their initialization as input or output ports using their data direction registers are discussed further in section 3.3.1.

**Step 2**: Pushbutton S3 (pin RD6) on the development board is used to provide user-control to the heart rate calculation. When the button is pushed the heart rate calculation procedure begins.

**Step 3**: Following the button push, the ADC module starts sampling the voltage signal supplied by the sensor circuit to pin AN3/RB3 through PORTB. The ADC is used to sample the signal every 5ms such that we obtain approximately 200 samples per second.
Figure 25. Heart rate calculation flowchart
Step 4: The ADC reading obtained from the above step is used to maintain a running average of all the ADC readings obtained since the button S3 has been pushed indicating the start of heart rate calculation.

Step 5: Before we start trying to detect a peak, the first 200 ADC sampling cycles in every test run are dedicated to obtaining a stable running average over a period of approximately 1 second.

Step 6: The ADC reading obtained in step 3 is also used to maintain an 8-reading average of eight consecutive ADC readings. Once we pass the 200 initial sample limit, the 8-reading average is calculated using the current ADC reading and seven past consecutive ADC readings.

Step 7: We check if the 8-reading average is greater than 3 times the running average, signifying the detection of a peak. If the check fails, we loop back to step 3.

Step 8: Whenever a peak is detected in the signal, a counter variable is incremented to denote peak detection. Following every peak detection, a 250ms delay is inserted prior to any further calculations. This dead delay is included to avoid any erroneous peak detections in the signal due to motion artifacts and duplicate peak detection during sampling of the downward slope of a QRS peak. Further, a signal with a time period of 250ms would indicate a heartbeat signal representing 240 BPM, higher than the maximum heart rate of 220 BPM possible for a human heart. Thus, with this 250ms delay while we ensure that there is no erroneous detection of noise peaks, we also ensure that we do not miss detection of any R peaks by inserting too long a delay.

Step 9: In this step, we check if the peak counter variable is equal to six. If it is not, the operation loops back to step 3. If it is, it implies the detection of six peaks in the heartbeat signal and we
move to calculation of heart rate as BPM. Timer1 is used to obtain the time period between
detection of the first peak and the sixth peak.

**Step 10:** Using the time period obtained from Timer1 in step 9, we average it to get the average
time period between the peaks of two consecutive cardiac cycles. The heart rate is then
calculated as BPM using equation 8.

**Step 11:** Finally, the calculated BPM is displayed on the LCD module and the eight indicator
LEDs are turned ON to indicate completion of heart rate calculation.

To attain a better understanding of all the operations performed in the above flowchart,
detailed discussions on some of the key modules used in the microcontroller and the
development board are presented in the following section.

### 3.3 PIC24FJ128GA010 microcontroller

The 16-bit PIC24FJ128GA010 microcontroller belongs to the Microchip PIC24FJ128GA
family of general purpose, flash microcontrollers. The 100-pin, RISC PIC24FJ128GA010
microcontroller offers an extensive peripheral feature suite and superior computational
performance. Key device and peripheral features of the microcontroller that have been utilized in
this project are listed as follows:

- 128Kbytes of flash program memory
- 8Kbytes of SRAM data memory
- 16-bit data and 24-bit address paths
- 84-I/O pins on 7 ports
- 8MHz internal oscillator
- C compiler optimized Instruction Set Architecture (contains 76 base instructions and flexible addressing modes)
- Five 16-bit Timers
- Five external interrupt sources
- 10-bit Analog-to-Digital converter
- Parallel Master Port

### 3.3.1 I/O Ports

As stated in the previous section, the 100-pin PIC24FJ128GA010 microcontroller contains 84 general purpose, bi-directional I/O pins that are distributed among 7 ports. Some of these I/O pins are multiplexed with alternate functions to add flexibility and functionality to the microcontroller. These additional functions depend on the peripheral features that are multiplexed to the pins. Shown in Table 1 is the I/O pin distribution among the seven I/O ports that are available in the PIC24 microcontroller.

<table>
<thead>
<tr>
<th>PORT NAME</th>
<th>PIN COUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>PORTA</td>
<td>12</td>
</tr>
<tr>
<td>PORTB</td>
<td>16</td>
</tr>
<tr>
<td>PORTC</td>
<td>8</td>
</tr>
<tr>
<td>PORTD</td>
<td>16</td>
</tr>
<tr>
<td>PORTE</td>
<td>10</td>
</tr>
<tr>
<td>PORTF</td>
<td>10</td>
</tr>
<tr>
<td>PORTG</td>
<td>12</td>
</tr>
</tbody>
</table>

**Table 1. I/O pin distribution**
All the above listed I/O ports have four registers that are related to the operation of the port and each I/O pin that belongs to a particular port has an associated bit in those four registers. The four registers are listed as follows, where ‘x’ in the register name denotes a particular I/O port A through G:

1. **TRISx register**: This is a data direction register whose control bits determine if the individual pins of a port function as an input pin or an output pin. If a TRIS bit associated with an I/O pin is set to ‘1’, then the pin functions as an input pin. If it is ‘0’, the pin functions as an output pin.

2. **PORTx register**: This is an I/O Port register that is used to access data on an I/O pin. To read data from a particular I/O pin, the corresponding bit in the associated PORT register can be read into a user-defined variable. An important point to be aware of is that, while a “read” of the PORT register reads the value of the I/O pin, a “write” to the PORT register will write the value to the port data latch. This might result in erroneous results for read-modify-write operations when a pin that was an input is changed to an output.

3. **LATx register**: This is an I/O Latch register that can be used to eliminate the read-modify-write problem. Unlike the PORT register, a “read” in the LAT register reads the value from the port output latch. A “write” operation is similar to that in a PORT register, where the LAT register writes the value to the port latch.

4. **ODCx register**: This is an Open-Drain Control register where the I/O pins in a port can be configured to give a normal digital output or an open-drain output. The open-drain feature is used to generate output voltages that are higher than $V_{DD}$ for any digital output pin by using external pull-up resistors. If the bit associated with a particular pin is set to ‘1’, then the pin works as an open-drain output. If it is a ‘0’, then the pin works like a normal digital output.
For the operations required in this project, the TRIS and PORT registers are used to access and control the following ports:

- **PORTA**: Used to control the 8 indicator LEDs.
- **PORTB**: Used to access input pin RB3/AN3 of the ADC module.
- **PORTD**: Used to poll pushbutton S3 for user-controlled heart rate calculation.

### 3.3.2 Oscillators

In the PIC24 microcontroller, the system clock source can be provided by one of following listed sources:\(^{32}\):

- **Primary Oscillator (POSC)** – external oscillator that can be configured to accept external clock input or external crystal on the OSC1 and OSC2 pins of the microcontroller.
- **Secondary Oscillator (SOSC)** – low-power external oscillator that is designed to work with a 32.768 KHz crystal on the SOSCI and SOSCO pins of the microcontroller.
- **Internal Fast RC Oscillator (FRC)** – internal oscillator that provides an 8MHz output, with an option to obtain clock speeds as low as 31KHz using the postscaler feature.
- **Internal Low-Power RC Oscillator (LPRC)** – low-power internal oscillator that gives a fixed 31KHz output useful for low-power and time-insensitive applications.

Shown in Figure 26 is a simplified block diagram of the oscillator system depicting all the clock sources available in the PIC24FJ128GA010 microcontroller. Using the four oscillator options up to 11 different clock modes can be obtained. From the above listed sources, any selected clock source can be used to generate the clock sources for the processor and selective peripherals. The processor clock source is divided by two to produce the internal instruction
cycle clock, denoted as $F_{CY}$ or $F_{OSC}/2$. In this project, the 8MHz FRC oscillator is used as the clock source for the different timer modules that are utilized during device operation. The clock source selection can be done using the FNOSC oscillator configuration bits in Configuration Word 2\(^{32}\). The FRC internal oscillator has been highlighted in red in Figure 26.

Figure 26. PIC24FJ128GA family clock diagram\(^{14}\)
The operation of the oscillator system is controlled by the following three special function registers:

1. **OSCCON register**: This is the main control register of the oscillator system that is used to monitor the clock sources and control switching between different clock sources. By default, this register is set to enable the FRCDIV option as the current clock source.

2. **CLKDIV register**: This clock divide register is used to control clock source features such as postscaler configurations, operation in power-saving mode, etc. When using the FRC oscillator, the default setting of the register results in an undivided 8MHz clock source.

3. **OSCTUN register**: This FRC Oscillator Tune register is used to allow user-defined fine tuning of the oscillator within ±12% range. By default, the register sets the oscillator to run at the center frequency that is factory calibrated.

### 3.3.3 Timers

The PIC24 microcontroller offers up to five 16-bit timers/counters, some of which can be combined to form a 32-bit timer. Key features of the timer modules include selectable prescaler settings, ability to generate interrupts based on the value set in a 16-bit period register, continued operation in CPU Idle and Sleep modes, etc. Each timer module in the microcontroller has the flexibility to be operated in any of the four modes listed next:

- Timer
- Gated timer
- Synchronous counter
- Asynchronous counter
In this project, the Timer1 and Timer2 modules are used in the 16-bit timer mode during device operation. Timer1 is used to maintain a record of the time period between the R peaks in the heart rate calculation algorithm discussed in section 3.2. Timer2 is used to insert the 5ms delay between ADC samplings and 250ms dead delay after every R peak detection as discussed in steps 3 and 8 in the flowchart. Each timer module in the PIC24 microcontroller can be accessed and operated using three special functions registers, which are listed next. The ‘x’ in each register name is used to denote a specific timer module the registers control.

1. **TMRx register**: A 16-bit timer count register that contains the timer count based on the configuration set in the TxCON register.

2. **PRx register**: This 16-bit register is a period register associated with each timer that contains the value up to which the timer counts to before resetting to 0000h.

3. **TxCON register**: This is a 16-bit Timer Control register that is used to configure the associated timer for desired operation.

   Shown in Figure 27 is a bit-level representation of the TxCON control register\(^{33}\). The settings used for Timer operation in this project are highlighted in red. Both Timer1 and Timer2 TxCON registers are set to a prescale value of 1:256. Further, each timer module has the ability to generate interrupts. Following is a list of bits associated with interrupt control for each timer module\(^{14}\):

   - **Interrupt Enable Control bit (TxIE)**: Present in the Interrupt Enable Control Registers. When this bit is set to ‘1’, the associated timer module is enabled to generate interrupts. When it is ‘0’, the associated timer module cannot generate interrupts.
**Figure 27. Timer control register**

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>TON: Timer1 On bit</td>
</tr>
<tr>
<td>14</td>
<td>Unimplemented: Read as '0'</td>
</tr>
<tr>
<td>13</td>
<td>TSDL: Stop in Idle Mode bit</td>
</tr>
<tr>
<td>12-7</td>
<td>Continue module operation in Idle mode</td>
</tr>
<tr>
<td>6</td>
<td>TGATE: Timer1 Gated Time Accumulation Enable bit</td>
</tr>
<tr>
<td>5-4</td>
<td>TCKPS1:TCKPS0: Timer1 Input Clock Prescale Select bits</td>
</tr>
<tr>
<td>3</td>
<td>Unimplemented: Read as '0'</td>
</tr>
<tr>
<td>2</td>
<td>TSYNC: Timer1 External Clock Input Synchronization Select bit</td>
</tr>
<tr>
<td>1</td>
<td>TCS: Timer1 Clock Source Select bit</td>
</tr>
<tr>
<td>0</td>
<td>Unimplemented: Read as '0'</td>
</tr>
</tbody>
</table>

Legend:
- R = Readable bit
- W = Writable bit
- U = Unimplemented bit, read as '0'
- n = Value at POR
- '1' = Bit is set
- '0' = Bit is cleared
- x = Bit is unknown
- **Interrupt Flag Status bit (TxIF):** Present in the Interrupt Flag Status Registers, this bit is first initialized to be ‘0’. When a timer module generates an interrupt, the associated flag bit is set to ‘1’ and the program control is passed to the corresponding Interrupt Service Routine (ISR). An interrupt can be generated by any 16-bit timer module whenever there is a period match with value in the PRx register or on the falling edge of an external gate signal. This bit has to be cleared in software after the interrupt has been served.

- **Interrupt Priority Control bits (TxIP<2:0>):** Present in the Interrupt Priority Control Registers, interrupts related to the timer module can be assigned up to 7 levels of priority using the three assigned bits.

More details pertaining to interrupts and interrupt control will be presented in section 3.3.4.

**Steps used to configure and operate the Timer modules:**

1. Turn on the Timer module by setting the TON bit to ‘1’ in the associated TxCON register.

2. Select the desired clock source using the TCS bit in the TxCON register. The internal clock source (F_{osc}/2) has been used in our project, and hence set the bit to ‘0’.

3. Select the desired prescaler ratio using the TCKPS1:TCKPS0 <5:4> in the TxCON register.

4. Load the associated PRx register with the desired timer period value where necessary.

5. For the Timer1 module, its associated interrupt control registers are configured to allow interrupt generation, as previously discussed.
### 3.3.4 Interrupts

The PIC24 microcontroller architecture provides a rich interrupt system with provision for about 118 distinct internal or external sources of interrupts. Each interrupt source has a unique pointer or “vector” associated with it that contains the starting address to an ISR that performs the required action to handle the interrupt\(^\text{34}\). All the interrupt vectors are stored in a table called an Interrupt Vector Table (IVT). The IVT contains a total of 126 vectors, including 8 non-maskable trap vectors in addition to the 118 interrupt sources\(^\text{14}\). The IVT resides in the program memory of the PIC24 microcontroller starting at location 000004h. The complete PIC24 interrupt vector table can be found in the PIC24FJ128GA family data sheet. A list of the implemented interrupt vectors with their associated vector numbers, IVT addresses and interrupt control bits is presented in Figure 28.

In the PIC24 microcontroller, a total of 28 registers are available to access and control different interrupt sources. These 28 registers include: INTCON1, INTCON2, Interrupt Flag Status registers (IFSx), Interrupt Enable Control registers (IECx) and Interrupt Priority Control registers (IPCx)\(^\text{14}\). INTCON1 and INTCON2 registers are primarily used to provide global interrupt control. As shown in Figure 28, each interrupt source listed in the table has control bits in three special function registers\(^\text{14}\):

- **Interrupt Flag Status (IFS) registers**: These registers maintain the interrupt request flags for all the interrupt sources. As discussed in the previous section, the flag bit associated with each interrupt source is set by the respective peripherals or the external signal causing the interrupt.
<table>
<thead>
<tr>
<th>Interrupt Source</th>
<th>Vector</th>
<th>IVT Address</th>
<th>AVT Address</th>
<th>Interrupt Bit Locations</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADC1 Conversion Done</td>
<td>13</td>
<td>00002Eh</td>
<td>00012Eh</td>
<td>IFS0&lt;13&gt; IEC0&lt;13&gt; IPC3&lt;8:4&gt;</td>
</tr>
<tr>
<td>Comparator Event</td>
<td>18</td>
<td>00003Fh</td>
<td>00013Fh</td>
<td>IFS1&lt;2&gt; IEC1&lt;2&gt; IPC4&lt;10:8&gt;</td>
</tr>
<tr>
<td>CRC Generator</td>
<td>67</td>
<td>00009Ah</td>
<td>00019Ah</td>
<td>IFS4&lt;3&gt; IEC4&lt;3&gt; IPC16&lt;14:12&gt;</td>
</tr>
<tr>
<td><strong>External Interrupt 0</strong></td>
<td>0</td>
<td>000014h</td>
<td>000114h</td>
<td>IFS0&lt;0&gt; IEC0&lt;0&gt; IPC0&lt;2:0&gt;</td>
</tr>
<tr>
<td><strong>External Interrupt 1</strong></td>
<td>20</td>
<td>00003Ch</td>
<td>00013Ch</td>
<td>IFS1&lt;4&gt; IEC1&lt;4&gt; IPC5&lt;2:0&gt;</td>
</tr>
<tr>
<td><strong>External Interrupt 2</strong></td>
<td>29</td>
<td>00004Eh</td>
<td>00014Eh</td>
<td>IFS1&lt;13&gt; IEC1&lt;13&gt; IPC7&lt;8:4&gt;</td>
</tr>
<tr>
<td><strong>External Interrupt 3</strong></td>
<td>53</td>
<td>00007Eh</td>
<td>00017Eh</td>
<td>IFS3&lt;5&gt; IEC3&lt;5&gt; IPC13&lt;8:4&gt;</td>
</tr>
<tr>
<td>External Interrupt 4</td>
<td>54</td>
<td>000050h</td>
<td>000150h</td>
<td>IFS3&lt;6&gt; IEC3&lt;6&gt; IPC13&lt;10:8&gt;</td>
</tr>
<tr>
<td>I2C1 Master Event</td>
<td>17</td>
<td>00003Fh</td>
<td>00013Fh</td>
<td>IFS1&lt;17&gt; IEC1&lt;17&gt; IPC4&lt;6:4&gt;</td>
</tr>
<tr>
<td>I2C1 Slave Event</td>
<td>16</td>
<td>00003Fh</td>
<td>00013Fh</td>
<td>IFS1&lt;0&gt; IEC1&lt;0&gt; IPC4&lt;2:0&gt;</td>
</tr>
<tr>
<td>I2C2 Master Event</td>
<td>50</td>
<td>000078h</td>
<td>000178h</td>
<td>IFS3&lt;2&gt; IEC3&lt;2&gt; IPC12&lt;10:8&gt;</td>
</tr>
<tr>
<td>I2C2 Slave Event</td>
<td>49</td>
<td>000070h</td>
<td>000170h</td>
<td>IFS3&lt;1&gt; IEC3&lt;1&gt; IPC12&lt;6:4&gt;</td>
</tr>
<tr>
<td>Input Capture 1</td>
<td>1</td>
<td>000016h</td>
<td>000116h</td>
<td>IFS0&lt;16&gt; IEC0&lt;16&gt; IPC0&lt;8:4&gt;</td>
</tr>
<tr>
<td>Input Capture 2</td>
<td>5</td>
<td>00001Eh</td>
<td>00011Eh</td>
<td>IFS0&lt;5&gt; IEC0&lt;5&gt; IPC1&lt;4:0&gt;</td>
</tr>
<tr>
<td>Input Capture 3</td>
<td>37</td>
<td>00005Eh</td>
<td>00015Eh</td>
<td>IFS2&lt;5&gt; IEC2&lt;5&gt; IPC9&lt;4:0&gt;</td>
</tr>
<tr>
<td>Input Capture 4</td>
<td>38</td>
<td>000060h</td>
<td>000160h</td>
<td>IFS2&lt;6&gt; IEC2&lt;6&gt; IPC9&lt;10:8&gt;</td>
</tr>
<tr>
<td>Input Capture 5</td>
<td>39</td>
<td>000062h</td>
<td>000162h</td>
<td>IFS2&lt;7&gt; IEC2&lt;7&gt; IPC9&lt;14:12&gt;</td>
</tr>
<tr>
<td>Input Change Notification</td>
<td>19</td>
<td>00003Ah</td>
<td>00013Ah</td>
<td>IFS1&lt;3&gt; IEC1&lt;3&gt; IPC4&lt;14:12&gt;</td>
</tr>
<tr>
<td>Output Compare 1</td>
<td>2</td>
<td>000018h</td>
<td>000118h</td>
<td>IFS0&lt;2&gt; IEC0&lt;2&gt; IPC0&lt;10:8&gt;</td>
</tr>
<tr>
<td>Output Compare 2</td>
<td>6</td>
<td>000020h</td>
<td>000120h</td>
<td>IFS0&lt;6&gt; IEC0&lt;6&gt; IPC1&lt;10:8&gt;</td>
</tr>
<tr>
<td>Output Compare 3</td>
<td>25</td>
<td>00004Fh</td>
<td>00014Fh</td>
<td>IFS1&lt;9&gt; IEC1&lt;9&gt; IPC6&lt;8:4&gt;</td>
</tr>
<tr>
<td>Output Compare 4</td>
<td>26</td>
<td>00004Fh</td>
<td>00014Fh</td>
<td>IFS1&lt;10&gt; IEC1&lt;10&gt; IPC8&lt;10:8&gt;</td>
</tr>
<tr>
<td>Output Compare 5</td>
<td>41</td>
<td>000066h</td>
<td>000166h</td>
<td>IFS2&lt;9&gt; IEC2&lt;9&gt; IPC10&lt;6:4&gt;</td>
</tr>
<tr>
<td>Parallel Master Port</td>
<td>45</td>
<td>000000h</td>
<td>000100h</td>
<td>IFS2&lt;13&gt; IEC2&lt;13&gt; IPC1&lt;10:8&gt;</td>
</tr>
<tr>
<td>Real-Time Clock/Calendar</td>
<td>62</td>
<td>000000h</td>
<td>000100h</td>
<td>IFS3&lt;14&gt; IEC3&lt;14&gt; IPC15&lt;10:8&gt;</td>
</tr>
<tr>
<td>SPI1 Error</td>
<td>9</td>
<td>000022h</td>
<td>000122h</td>
<td>IFS0&lt;9&gt; IEC0&lt;9&gt; IPC2&lt;2:0&gt;</td>
</tr>
<tr>
<td>SPI1 Event</td>
<td>10</td>
<td>000028h</td>
<td>000128h</td>
<td>IFS0&lt;10&gt; IEC0&lt;10&gt; IPC2&lt;10:8&gt;</td>
</tr>
<tr>
<td>SPI2 Error</td>
<td>32</td>
<td>000054h</td>
<td>000154h</td>
<td>IFS2&lt;0&gt; IEC2&lt;0&gt; IPC8&lt;2:0&gt;</td>
</tr>
<tr>
<td>SPI2 Event</td>
<td>33</td>
<td>000056h</td>
<td>000156h</td>
<td>IFS2&lt;1&gt; IEC2&lt;1&gt; IPC8&lt;8:4&gt;</td>
</tr>
<tr>
<td>Timer1</td>
<td>3</td>
<td>00001Ah</td>
<td>00011Ah</td>
<td>IFS0&lt;3&gt; IEC0&lt;3&gt; IPC0&lt;14:12&gt;</td>
</tr>
<tr>
<td>Timer2</td>
<td>7</td>
<td>000022h</td>
<td>000122h</td>
<td>IFS0&lt;7&gt; IEC0&lt;7&gt; IPC1&lt;14:12&gt;</td>
</tr>
<tr>
<td>Timer3</td>
<td>8</td>
<td>000024h</td>
<td>000124h</td>
<td>IFS0&lt;8&gt; IEC0&lt;8&gt; IPC2&lt;2:0&gt;</td>
</tr>
<tr>
<td>Timer4</td>
<td>27</td>
<td>00004Ah</td>
<td>00014Ah</td>
<td>IFS1&lt;11&gt; IEC1&lt;11&gt; IPC6&lt;14:12&gt;</td>
</tr>
<tr>
<td>Timer5</td>
<td>28</td>
<td>00004Ch</td>
<td>00014Ch</td>
<td>IFS1&lt;12&gt; IEC1&lt;12&gt; IPC7&lt;2:0&gt;</td>
</tr>
<tr>
<td>UART1 Error</td>
<td>65</td>
<td>000096h</td>
<td>000196h</td>
<td>IFS4&lt;1&gt; IEC4&lt;1&gt; IPC18&lt;8:4&gt;</td>
</tr>
<tr>
<td>UART1 Receiver</td>
<td>11</td>
<td>00002Ah</td>
<td>00012Ah</td>
<td>IFS0&lt;11&gt; IEC0&lt;11&gt; IPC2&lt;14:12&gt;</td>
</tr>
<tr>
<td>UART1 Transmitter</td>
<td>12</td>
<td>00002Ch</td>
<td>00012Ch</td>
<td>IFS0&lt;12&gt; IEC0&lt;12&gt; IPC3&lt;2:0&gt;</td>
</tr>
<tr>
<td>UART2 Error</td>
<td>66</td>
<td>000096h</td>
<td>000196h</td>
<td>IFS4&lt;2&gt; IEC4&lt;2&gt; IPC18&lt;10:8&gt;</td>
</tr>
<tr>
<td>UART2 Receiver</td>
<td>30</td>
<td>000050h</td>
<td>000150h</td>
<td>IFS1&lt;14&gt; IEC1&lt;14&gt; IPC7&lt;10:8&gt;</td>
</tr>
<tr>
<td>UART2 Transmitter</td>
<td>31</td>
<td>000052h</td>
<td>000152h</td>
<td>IFS1&lt;15&gt; IEC1&lt;15&gt; IPC7&lt;14:12&gt;</td>
</tr>
</tbody>
</table>

Figure 28. Implemented interrupt vectors\textsuperscript{14}
• **Interrupt Enable Control (IEC) registers**: These control registers maintain the interrupt enable bits of all the interrupt sources to individually enable interrupts from the peripherals or external signals.

• **Interrupt Priority Control (IPC) registers**: These control registers are used to assign priority levels for each source of interrupt, where each interrupt source is provided with three bits that can be used to assign up to seven (1 to 7) priority levels. At power on, all interrupt sources are assigned a default priority level of four by initializing the IPC registers, while the processor priority is initially set at level zero using the Interrupt Priority Level(IPL<2:0>) bits in the CPU Status register (SR). If the priority control bits associated with a particular interrupt source are cleared, then the interrupt source is effectively disabled.

If two interrupts occur at the same time, the interrupt event with the higher priority is served first. Further, if interrupts with the same user-assigned priority level occur at the same time, then the conflict is resolved based on the natural order of priority of the interrupt sources as listed in the IVT and shown in Figure 28. Nesting of interrupts is available in PIC24 to allow the interruption of a lower-priority interrupt service routine by a higher-priority routine during device operation. This interrupt nesting is controlled using the NSTDIS bit in the INTCON1 register of the PIC24. By default, all interrupts are nestable. A ‘0’ in the NSTDIS bit denotes that interrupt nesting is enabled, while setting it to ‘1’ disables interrupt nesting during device operation.

Among the various internal and external sources of interrupt available, in this project two interrupts sources are used during device operation: Timer1 and External Interrupt 0, which are highlighted in red in the Figure 28. Timer1 is an internal interrupt source that is used to generate an interrupt whenever Timer1 reaches the value specified in the PR1 period register. In the
source code, a timer count variable is incremented whenever the 1:256 pre-scaled Timer1 reaches the value 15625 specified in the period register, signifying the lapse of one second. Next, the External Interrupt 0 is used to generate an interrupt whenever the button S3 is pushed to begin the heart rate calculation. Up to five external interrupt sources are available in the PIC24 microcontroller interrupt system that can be used to generate interrupts on the occurrence of a user-defined interrupt event. In this project, the External Interrupt 0 source and its service routine have been configured to generate an interrupt when button S3 is pushed and start the heart rate calculation process.

3.3.5 ADC Converter

The PIC24 microcontroller contains an in-built 10-bit, high-speed ADC module that is used to sample the heart beat voltage signal obtained from the sensor module and map it to a 10-bit digital value that is used for further processing\textsuperscript{36}. Since the ADC module is one of the key modules whose proper configuration and operation is essential for the accurate functioning of our prototype device, important features related to it will be discussed in detail in this section. Some key features of the integrated ADC module are listed next\textsuperscript{15}:

- 16 external analog input channels available.
- Provision for external voltage reference inputs.
- Successive Approximation Register (SAR) conversion mechanism employed.
- Contains a unipolar differential Sample and Hold (S/H) Amplifier.
- 16-bit wide, 16 word conversion Result Buffer.
- User-definable buffer fill modes.
- Continued ADC module operation during CPU Sleep and Idle modes.
(i) **ADC Block Diagram**

As listed above and shown in Figure 29, the ADC module is equipped to handle up to 16 analog inputs and separate external reference voltage inputs. The ADC samples a single input signal at a time from the multiple input channels using a combination of the different input multiplexers, as shown in the figure below.

![Figure 29. ADC block diagram](image)

Figure 29. ADC block diagram\textsuperscript{14}
This 10-bit ADC present in the PIC24 microcontroller is a successive approximation type of ADC which provides fast conversions of up to 500K samples per second. In an SAR type of ADC, instead of counting up in a binary sequence the converter outputs a final digital value of the sampled value by performing a binary search through different quantization levels\(^\text{37}\). This results in a converter that can provide results faster than a regular ADC. The SAR type of ADC consists of four main parts: a sample and hold amplifier, an analog voltage comparator, a 10-bit SAR register and an internal digital-to-analog converter (DAC), as shown in Figure 29.

- **Sample and Hold (S/H) Amplifier:** The S/H amplifier, as the name suggests, samples the analog signal and holds the value constant during the conversion process. During the sampling phase the sample/hold \(C_{\text{HOLD}}\) capacitor is charged to the input voltage and during the conversion phase it is disconnected from the analog input pin, as shown in the input model of the ADC module in Error! Reference source not found.\(^\text{38}\).

- **Comparator:** The analog voltage comparator is used to compare the sampled input voltage value and the output from the DAC module, based on the result of which the conversion logic associated with the ADC module is applied and the SAR register is updated.

- **10-bit SAR:** This register contains the approximate 10-bit digital value of the sampled input voltage which is then supplied to the DAC module as an input.

- **DAC module:** This internal DAC is used to provide the comparator with the analog voltage equivalent of the digital value contained in the SAR register.

More details related to the design and operation of a SAR type of ADC can be found in these references\(^\text{37,39}\). Once the conversion is done and the finalized digital value is written to the 10-bit SAR, the data can be modified into desired format and written into one of the sixteen result
registers as 8-bit or 16-bit words. All the data formatting bits and buffer mode selection bits can be accessed using the assigned configuration bits in the ADC control registers. More details related to the ADC control and result registers will be presented next. By configuring the various control registers associated with the ADC module, the timing and control functions can be set to allow automatic channel scanning, select the desired conversion trigger source, enable interrupts, etc.

(ii) Registers

The ADC module has 6 special function registers and 16 result buffers associated with it that are all mapped in the data memory space. The result buffers are implemented as a 16-word, dual port RAM in the data memory and each location, named ADC1BUF0 through ADC1BUFF, is separately accessible to read and write ADC results. The six A/D special function registers used to configure and control the operation of the ADC module are:

- **A/D Control Register 1 (AD1CON1):** The first A/D control register, shown in Figure 30, is used to turn on the converter, select integer data output format, enable and control sampling, and configure the A/D module for auto-conversion in the source code.

- **A/D Control Register 2 (AD1CON2):** The second A/D control register, shown in Figure 31, is configured to use MUX A, access result register as 16-bit buffer and use the input rails AVDD and AVSS as the ADC reference inputs.

- **A/D Control Register 3 (AD1CON3):** The third A/D control register, as shown in Figure 32 and discussed in the following section, is used to select the A/D conversion clock source, sampling and conversion times.
**Figure 30. A/D control register**

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit 15</td>
<td>ADON: A/D Operating Mode bit</td>
<td>1 = A/D Converter module is operating, 0 = A/D Converter is off</td>
</tr>
<tr>
<td>Bit 14</td>
<td>Unimplemented: Read as ‘0’</td>
<td></td>
</tr>
<tr>
<td>Bit 13</td>
<td>ADSIDL: Stop in Idle Mode bit</td>
<td>1 = Discontinue module operation when device enters Idle mode, 0 = Continue module operation in Idle mode</td>
</tr>
<tr>
<td>Bit 12-10</td>
<td>Unimplemented: Read as ‘0’</td>
<td></td>
</tr>
<tr>
<td>Bit 9-8</td>
<td>FORM1:FORM0: Data Output Format bits</td>
<td>11 = Signed fractional (sddd dddd dd00 0000), 10 = Fractional (ddd dddd dd00 0000), 01 = Signed integer (ssss ssss ssss ssss), 00 = Integer (0000 00dd dddd dddd)</td>
</tr>
<tr>
<td>Bit 7-5</td>
<td>SSRC2:SSRC0: Conversion Trigger Source Select bits</td>
<td>1110 = Reserved, 110x = Reserved, 101x = Reserved, 1000 = Reserved, 1001 = Timer3 compare ends sampling and starts conversion, 0011 = Active transition on INT0 pin ends sampling and starts conversion, 0000 = Clearing SAMP bit ends sampling and starts conversion</td>
</tr>
<tr>
<td>Bit 4-3</td>
<td>Unimplemented: Read as ‘0’</td>
<td></td>
</tr>
<tr>
<td>Bit 2</td>
<td>ASAM: A/D Sample Auto-Start bit</td>
<td>1 = Sampling begins immediately after last conversion completes. SAMP bit is auto-set, 0 = Sampling begins when SAMP bit is set</td>
</tr>
<tr>
<td>Bit 1</td>
<td>SAMP: A/D Sample Enable bit</td>
<td>1 = A/D sample/hold amplifier is sampling input, 0 = A/D sample/hold amplifier is holding</td>
</tr>
<tr>
<td>Bit 0</td>
<td>DONE: A/D Conversion Status bit</td>
<td>1 = A/D conversion is done, 0 = A/D conversion is NOT done</td>
</tr>
</tbody>
</table>

Legend:  
- R = Readable bit  
- W = Writable bit  
- U = Unimplemented bit, read as ‘0’  
- C = Clearable bit  
- x = Bit is unknown  
- HCS = Hardware Clearable/Settable bit  
- -n = Value at POR: ‘1’ = Bit is set, ‘0’ = Bit is cleared
Figure 31. A/D control register 2\textsuperscript{nd}
• **A/D Input Select Register** *(AD1CHS)*: This register is used to select the inputs that are to be connected to the S/H module. In this project, pin AN3 is used to send the analog signal from the sensor circuit to the ADC module through MUX A.

• **A/D Port Configuration Registers** *(AD1PCFG)*: This register is used to configure the 16 I/O pins associated with the ADC module as analog inputs or digital I/Os. A ‘0’ is used to configure a particular pin as an analog input, while ‘1’ is used to configure it as a digital I/O.

• **A/D Input Scan Select Register** *(AD1CSSL)*: This register is used to select the input channels that are to be included for sequential scanning. Since a single channel is used in this project, we do not need scanning and all the bits in this register have hence been cleared.
(iii) ADC Sample/Conversion Sequence

Shown in Figure 33 is the analog-to-digital sample and conversion sequence performed by the ADC module. The sample time is essentially the time during which the S/H module is connected to the analog input pin and the holding capacitor is charged to the sampled voltage value. The sample time can be controlled automatically using bits SAMC<4:0> in the control register AD1CON3 or controlled directly through the program code. In general, for accurate sampling of the input signal, the sampling time should be long enough to allow the holding capacitor to charge completely. Hence, in this project the maximum possible sample time of $31 \times T_{AD}$ has been used, where $T_{AD}$ is the ADC clock cycle period.

Conversion time is the time required to convert the voltage sampled by the S/H module. The trigger source to begin the A/D conversion can be controlled directly in software, via a variety of hardware sources or be set to auto-convert at the end of each sample period using the SSRC<2:0> bits in the AD1CON1 control register. While the conversion time should be as small as possible, for accurate A/D conversions the ADC clock period $T_{AD}$ is required to be greater

![Figure 33. ADC sample/conversion sequence](image)

Figure 33. ADC sample/conversion sequence
than 75ns, which is specified as parameter AD130 in the datasheet\textsuperscript{15}. This conversion clock time can be adjusted using the ADCS<7:0> bits in the AD1CON3 register and using equation 9 given below\textsuperscript{14}. In the equation, ADCS is the value given using bits ADCS<7:0> in the AD1CON register and T\textsubscript{CY} is the value given by 1/F\textsubscript{CY} or 2/F\textsubscript{OSC}. In our source code, since the A/D conversion clock source is derived from the internal system clock of 8MHz, a value of zero has been chosen for the ADCS such that a T\textsubscript{AD} of 250ns is obtained which is greater than the required minimum of 75ns.

\[ T_{AD} = T_{CY} \times (ADCS + 1) \quad (9) \]

(iv) Transfer Function

The ADC module in the PIC24 microcontroller has 10-bit resolution where any voltage value in the selected reference range is mapped to a 10-bit digital value. Using the VCFG<2:0> bits in the AD1CON2 register, the analog reference range for the ADC module can be configured to use the microcontroller’s AV\textsubscript{DD} and AV\textsubscript{SS} analog supply voltages or use the external voltages from the external V\textsubscript{REF+} and V\textsubscript{REF-} pins. In this project, we elect to use the device supply voltages such that any voltage value in the 0V – 3.3V range is mapped to a 10-bit value represented by numbers in the range 0 – 1023. The A/D module transfer function is depicted graphically in Figure 34 and the ADC output value can be calculated using equation 10, where F.S is the full-scale value of 1023\textsuperscript{38}.

\[
ADC \text{ output code} = F.S \times \frac{V_{IN+} - V_{IN-}}{V_{REF+} - V_{REF-}} = F.S \times \frac{V_{IN}}{V_{REF}} \quad (10)
\]
Steps used to configure and operate the ADC module

1. Configure the control registers AD1CON1, AD1CON2 and AD1CON3, as previously discussed, to the desired settings of sample/conversion sequence, reference voltages, output data formats, buffer fill modes, etc.

2. Configure pin AN3 as an analog input.

3. Turn ON the ADC module using bit ADON in register AD1CON1.
4. Use register AD1CHS to connect input pin/channel to the S/H module for sampling.

5. Begin ADC sampling using the SAMP bit in the AD1CON1 register and wait for auto-conversion to end by polling the DONE bit in the same register.

6. Read the ADC1BUF0 register to obtain the converted digital value of the sampled analog signal.

### 3.3.6 Parallel Master Port

The parallel master port, or PMP, is an 8-bit, parallel I/O module that can be used by the microcontroller to communicate with a variety of external parallel devices, such as LCDs, external memory devices and microcontrollers. With the availability of this separate module to access the external devices, interference with the operation of the program memory bus and data bus can be reduced. Shown in Figure 35 is a block diagram that demonstrates the use of PMP module to communicate with some external devices. Some key features of the module are:

- 8 data lines
- Up to 16 programmable address lines
- User-controllable strobe options (individual read and write strobes or single read/write strobe with an enable strobe)
- User-controllable address/data multiplexing
- User-controllable wait states

In addition to reducing the traffic on the program and data buses by using the PMP module for external device access, the special function registers associated with the module can be used to configure parameters such as read/write sequencing and timing to adapt to the speed of the
external peripheral\textsuperscript{34}. The PMP module can also be configured to function in slave mode where the current microcontroller can be attached as a peripheral to another microcontroller system. In this thesis, the PMP module is used to access the alphanumeric LCD that is available on the development board to display the calculated heart rate, as shown in Figure 36.

The 2-row and 16-character LCD module present on the development board is compatible with all industry-standard HD44780 controllers. Using its standard set of commands, our display has been initialized and configured to function properly\textsuperscript{34}. Apart from the initialization of the LCD module, the PMP module also needs to be initialized using its associated special function registers prior to normal module operation. More details related to these special function registers and their configuration to access and control the LCD display will be discussed in the rest of the section using information from reference\textsuperscript{34}.

\textbf{Figure 35. PMP module}\textsuperscript{14}
(i) Registers

In this section, details related to the configuration of several special function registers associated with the PMP module are presented. In this project, we utilize some of the following listed registers to configure and operate the PMP module.

- **Parallel Master Port Control Register (PMCON):** The main control register associated with the PMP module, PMCON is used to enable the PMP module, enable strobe signal and read/write signal, completely de-multiplex address and data lines, etc.

- **Parallel Master Port Mode Register (PMMODE):** This register is used to select the operational modes of the PMP module. The module is configured to operate in the Master mode with 8-bit data transfer and maximum wait times for data read/write operations. This register contains the status flag bit, BUSY, that is used to specify that an operation is in progress and the module is busy.

- **Parallel Master Port Address Register (PMADDR/PMDOUT1):** This register is used to provide the address to which the output data is to be written and configure chip select control bits required to address slave devices. When the PMP module is operated in slave mode, the register is referred to as PMOUT1 and functions as a buffer for data.

- **Parallel Master Port Data Output 2 Register (PMDOUT2):** Again, if the PMP module is operated in the slave mode, this register is used to buffer output data.
• **Parallel Master Port Data Input Registers (PMDIN1 and PMDIN2):** These input registers are generally used to buffer input data. While the PMDIN1 can buffer both input and output data in the Master mode, both the registers can only be used to hold the input data in the Slave mode.

• **Parallel Master Port Address Enable Register (PMAEN):** This register is used to configure the address and chip select pins associated with the PMP module. A ‘1’ in the control bit is used to assign the corresponding PIC24 pins to the PMP module, and a ‘0’ is used to assign the pins to port I/O. This register is used to enable the PMA0 pin and control the address line (RS) for register selection, as shown in Figure 36.

• **Parallel Master Port Status Register (PMSTAT):** When the PMP module is operated in the Slave mode, this status register contains information related to buffered operating modes.

(ii) **Steps used to configure and operate the PMP and LCD modules**

1. Initialize the PMP module using the PMCON, PMMODE and PMAEN registers, as previously discussed.

2. Initialize the LCD module for proper operation using the associated controller commands and setup times$^{34}$.

3. When a character is sent to be written to the LCD display, first the LCD modules and PMP modules are checked to ensure they not busy.

4. Once the LCD and PMP modules are free, the sent character is written to the data register and displayed on the LCD. Steps 3 and 4 are repeated for every character that is displayed on the LCD.
Chapter 4 – Results

In chapters 2 and 3, design and implementation details related to the sensor circuit and the PIC24-microcontroller based development board have been thoroughly discussed. Combining both these modules, we construct a heart rate monitoring system that can detect heart beat from the tip of a finger, calculate heart rate and display it as BPM on an LCD display. In this chapter, details related to the test setup of this prototype heart rate monitoring system are presented along with experimental results for different test scenarios.

4.1 Test Setup

Shown in Figure 37 is the complete test setup of the prototype heart rate monitor. Shown on the left is the PCB with the optical sensor circuit and on the right is the Explorer 16 development board. The PCB is connected to the development board using a setup consisting of a prototype daughter board and jumper wire through the PICtail™ Plus connector slot. This special interface setup has been implemented to ensure a stable connection between the PCB and the development board through its PICtail™ Plus socket connector slot.

To verify the proper functioning of this heart rate monitoring system, we study the heartbeat signal obtained from the tip of a finger at the output of the sensor circuit using an oscilloscope. For this purpose, additional test points have been included on the PCB at the output of the sensor module and a Hewlett-Packard 54600B oscilloscope is used to observe the heartbeat voltage signal. Once a stable signal is obtained from the sensor circuit, the heart rate is calculated as BPM using the prototype device and compared with the heart rate estimated from the waveform being observed on the oscilloscope. To obtain a stable heart beat signal using the current test
setup, there should be minimal interference from ambient light sources and a secure LED and photodiode setup to ensure minimal finger and hand movement. More details related to the test setup and experimental results obtained for different test cases will be presented in the following section.

4.2 Results and Discussion:

The normal resting heart rate for adults is generally between 60 to 100 BPM\textsuperscript{41}. It changes in response to various internal and external conditions and varies between individuals based on physiological conditions and differences\textsuperscript{42}. Heart rate generally increases with any muscular
exercise, rising as high as 180 BPM with vigorous physical activity. In this section we present the heart rates obtained using our prototype heart rate monitor in different test scenarios and compare them with the heart rate approximated from the heartbeat waveforms observed on the oscilloscope and the heart rate estimated from a pulse point on the neck.

Shown in Table 2 are the heart rates obtained from tip of the thumb from two individuals following different levels of physical activity: a) before exercise and b) after exercise. As shown in the table, the BPM calculations obtained from the prototype heart rate monitoring system closely match the heart rates estimated from the waveforms observed on the oscilloscope and heart rate taken from the pulse point on the neck. The heart rate estimated from the pulse point has been taken after the BPM calculations were done using the prototype.

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Subject</th>
<th>Heart Rate Monitor Estimation</th>
<th>Oscilloscope Waveform Estimation</th>
<th>Pulse Estimation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Individual 1</td>
<td>80</td>
<td>~85</td>
<td>~82</td>
</tr>
<tr>
<td>2</td>
<td>Individual 1</td>
<td>115</td>
<td>~120</td>
<td>~112</td>
</tr>
<tr>
<td>3</td>
<td>Individual 2</td>
<td>76</td>
<td>~71</td>
<td>~78</td>
</tr>
<tr>
<td>4</td>
<td>Individual 2</td>
<td>129</td>
<td>~125</td>
<td>~121</td>
</tr>
</tbody>
</table>

Table 2. Estimated heart rates as beats per minute (BPM)

Shown in Figure 38, Figure 39, Figure 40, and Figure 41 are the BPM estimations obtained using the prototype along with their corresponding oscilloscope waveforms. The settings used on the oscilloscope are voltage scale: 100mV volt/div and time scale: 200ms time/div using a 10:1 probe. The heart rate estimation from the oscilloscope waveform is done using the following formula:

\[
BPM = \frac{60}{\text{Number of time divisions} \times 200\text{ms}}
\]  
(9)
Figure 38. Test case 1- Individual 1 before exercise

Figure 39. Test case 2- Individual 1 after exercise
Figure 40. Test case 3- Individual 2 before exercise

Figure 41. Test case 4- Individual 2 after exercise
Finally, we present a few more details related to the test setup and issues encountered during the development of this heart rate monitoring system:

(i) The voltage of the heartbeat signal obtained from the optical sensor module depends on a variety of factors. The amount of pressure exerted by the finger while holding the LED and photodiode setup effects the amplitude of the signal obtained from the sensor module. This difference in the amplitude of the signal can be observed in figures Figure 38 and Figure 40 where the amplitude of the R peaks varies between individuals due to difference in the amount of pressure exerted by the test subject while holding the LED and photodiode setup. Further, interference from ambient light sources also effects the strength of the R peaks in the heart beat signal.

(ii) Movement of the LED and photodiode setup during a test due to hand tremors and finger movement also effects the quality of the heart beat signal, where additional peaks could be introduced into the signal. Shown in Figure 42 is an instance of occurrence of such peaks in the heart beat signal which could lead to erroneous peak detection and incorrect BPM estimation.

(iii) Low frequency oscillations, called "motorboat" oscillations, have been encountered during the development of this device. In AC-coupled feedback op-amp circuits, low frequency oscillations can be introduced into the signal due to accumulation of leading phase shifts produced by the capacitively coupled stages. The blocking capacitor together with the input resistance of the circuit cause leading phase shifts, and with enough loop gain the op-amp could go into low frequency oscillations. This problem can be solved by reducing the gain of the circuit. Further, motorboat oscillations can also be caused in an op-amp due to power supplies
with inadequate internal bypassing. These oscillations can be avoided by adding adequate power supply bypassing using appropriate ceramic and tantalum capacitors.

**Figure 42.** Erroneous peak detection due to hand or finger movement
5.1 Conclusion

A prototype heart rate monitoring system has been successfully developed that can detect heart beat from the tip of a finger and calculate heart rate as BPM. This prototype device has been designed and implemented with the goal of incorporating several key areas pertaining to the fields of computer and electrical engineering, as discussed in chapter 1.

The heart rate monitor is made up of two main modules: the optical sensor module and the PIC24-microcontroller based development board. The optical sensor module contains a central LED and photodiode setup together with multiple amplifier and filter circuits that are used to detect heart beat from a fingertip based on the reflection-based photoplethysmography principle. This module can be used to demonstrate some key concepts related to electrical engineering such as basic electronics, semiconductor devices, analog circuit design and PCB design. The development board module, which contains a 16-bit PIC microcontroller with an in-built ADC module, primarily calculates the heart rate based on the signal obtained from the sensor circuit and displays it as BPM on an alphanumeric LCD present on the development board. This part of the module can be used to demonstrate concepts related to computer engineering such as embedded systems, microcontrollers, software development and hardware/software co-design.

Finally, the heart rate calculations performed using the prototype device have been compared with the heart rate calculations estimated from the heart beat waveforms observed on an oscilloscope and the heart rate estimated from a pulse point, thus validating the proper operation of the device.
5.2 Suggestions for future work

The current prototype device is quite sensitive to motion artifacts such as hand tremors and finger movement. A very stable heart rate signal is required to perform the heart rate calculations using the current device setup. While our current LED and photodiode setup is functionally adequate to obtain a stable heart beat in general, the sensor setup can be improved to provide a more stable interface. The current setup allows reliable heart rate measurements to be taken only from the tip of a finger. A different setup can be designed to allow heart rate to be calculated from other measurement sites on the body with a strong pulse such as an earlobe, neck, wrist, etc.

Further, more complex algorithms can be developed for the R peak detection that take into account external and internal noise sources such as interference from motion artifacts, muscle artifacts (electromyography), baseline drift due to respiration, etc\textsuperscript{44,45}. Digital filtering and signal processing techniques can be employed to improve the accuracy of the heart rate calculations by further reducing the effect of the above discussed sources of signal interferences\textsuperscript{30}.

The prototype device developed in this thesis can be made smaller and more portable by combining the sensor circuit and the microcontroller, ADC and LCD modules onto a single PCB. Instead of using the development board, if the PCB is designed to incorporate a discrete microcontroller, ADC and LCD modules together with the optical sensor circuit, then the whole monitoring system would consist of a single PCB that is more compact.

This prototype heart rate monitoring system can be used to develop an instructional module that can be used by educators to introduce students to some key concepts related to electrical and computer engineering, as previously discussed in chapter 1.
A final idea for future work is the modification of the current system to include pulse oximetry. As discussed in chapter 1, pulse oximetry is used to study the oxygen saturation of blood and is based on the concept of photoplethysmography. The basic principle of pulse oximetry is to identify the levels of oxygenated hemoglobin and reduced hemoglobin based on their light absorption characteristics at different wavelengths to study the oxygen levels of blood\textsuperscript{46}. Typically, light sources in two wavelength regions: red(650-750nm) and infrared(900-1000nm) are used since reduced hemoglobin absorbs more red light than oxygenated hemoglobin, while oxygenated hemoglobin absorbs more infrared light than reduced hemoglobin\textsuperscript{47}. By calculating the ratio of the absorbed red light and infrared light intensities, a ratio of the oxygenated and reduced hemoglobin levels in blood can be obtained which in turn will help obtain the oxygen saturation level of blood\textsuperscript{46}. Our current prototype system can be modified to include pulse oximetry by adding required hardware in the sensor circuit and modifying the logic in the algorithm to obtain blood oxygenation levels in addition to the heart rate information.
References


13 Explorer 16 development board user’s guide, Microchip, DS51589A, 2005.


Biomedical Engineering: Applications, Basis and Communications (BME), vol. 17, no. 5, pp. 258-262, 2005.


36 Using the analog-to-digital(A/D) converter, Microchip, DS00546E, 1997.


