Heart Rate Detection System Based on PPG Signal Using MATLAB

A CAPSTONE PROJECT REPORT

Submitted in partial fulfillment of the requirement for the award of the Degree of

BACHELOR OF TECHNOLOGY
IN
ELECTRONICS AND COMMUNICATION ENGINEERING

by

SAPPARAPU DEVI SRI PRASAD (21BEC7245)

Under the Guidance of

DR. G D V SANTOSH KUMAR

SCHOOL OF ELECTRONICS ENGINEERING
VIT-AP UNIVERSITY
AMARAVATI- 522237

December 2024

CERTIFICATE

This is to certify that the Capstone Project work titled "**HEART RATE DETECTION BASED ON PPG SIGNAL USING MATLAB**" that is being submitted by **SAPPARAPU DEVI SRI PRASAD(21BEC7245)**is in partial fulfillment of the requirements for the award of Bachelor of Technology, is a record of bonafide work done under my guidance. The contents of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for award of any degree or diploma and the same is certified.

Dr.G D V SANTOSH KUMAR 16/12/24

Guide

The thesis is satisfactory / unsatisfactory

Approved by

PROGRAM CHAIR

B. Tech. ECE

DEAN

School Of Electronics Engineering

ACKNOWLEDGEMENTS

I would like to formally acknowledge the individuals and institutions whose support and contributions were instrumental in the completion of this capstone project, titled **"Heart Rate Detection System Based on PPG Signal Using MATLAB"**.

I extend my sincere gratitude to my guide, **Dr. G D V SANTOSH KUMAR**, Professor in the School of Electronics Engineering (SENSE), VIT-AP University, for his guidance and technical expertise throughout this project. His inputs and feedback have been critical to the successful execution of this work.

I also acknowledge the contributions of my team members, **Sikhatapu HariBabu (21BEC7264)** and **Bhanu Vasanth Butti (21BEC7219)**, for their collaboration and dedication in various aspects of the project.

Additionally, I wish to recognize the faculty of the SENSE department at VIT-AP University for their knowledge-sharing and support, which have significantly enhanced my understanding and technical abilities.

Finally, I appreciate the infrastructure and resources provided by VIT-AP University, which facilitated the smooth progress and timely completion of this work.

ABSTRACT

In this paper, an efficient system is presented for heart rate detection using Photoplethysmography (PPG) signals processed in MATLAB. The proposed system automates the entire process of signal acquisition, preprocessing, and heart rate calculation while ensuring the quality and reliability of the input signal. Regular monitoring of heart rate and other physiological parameters is crucial for early detection of abnormalities and maintaining cardiovascular health.

The Heart Rate Detection System processes PPG signals by removing noise, filtering undesired frequencies, and detecting peaks to calculate beats per minute (BPM). A Signal Quality Index (SQI) is calculated to assess the reliability of the signal, ensuring accurate results even under noisy conditions. MATLAB serves as the primary platform for implementing signal processing algorithms, including bandpass filtering, normalization, peak detection, and BPM estimation.

Future extensions of this work involve deploying the system on FPGA platforms using logic synthesis techniques for real-time applications. Integrating the PPG block into wearable devices will enable continuous health monitoring while ensuring low power consumption and high accuracy. This system has the potential to improve accessibility to real-time health monitoring, making it a vital tool for early detection and prevention of cardiovascular diseases.

TABLE OF CONTENTS

List of Figures

CHAPTER 1

INTRODUCTION

Health monitoring systems have become an essential part of modern healthcare. With the increasing prevalence of chronic illnesses and the rising demand for wearable devices, continuous monitoring of physiological parameters like heart rate has gained significant importance. Photoplethysmography (PPG) is a simple, non-invasive optical technique widely used in wearable devices to monitor heart rate, oxygen saturation, and other vital signs.

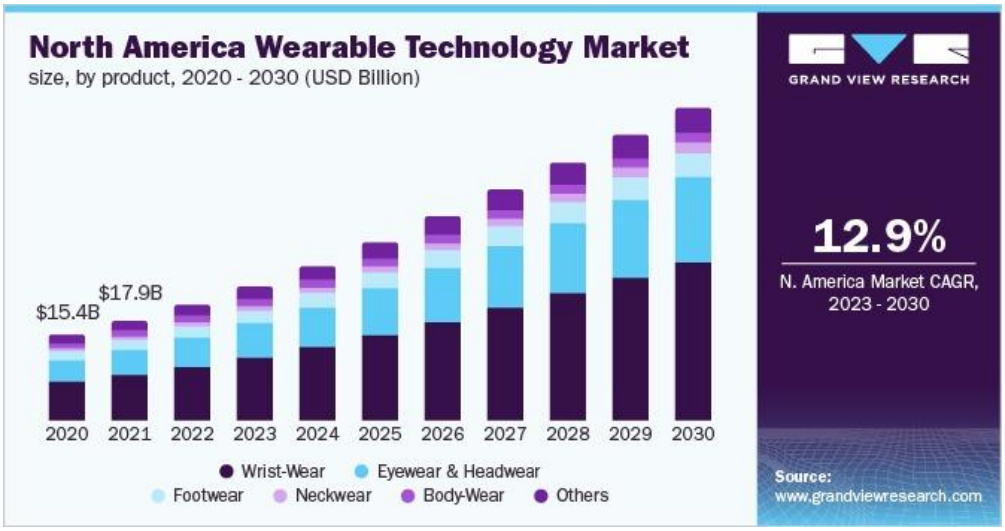

Figure 1wearable Technology usage

A significant challenge in heart rate monitoring arises due to motion artifacts, ambient light interference, and the overall quality of the acquired PPG signals. These factors can lead to incorrect heart rate estimations, which can compromise the reliability of health monitoring systems. Traditional systems often rely on simple threshold-based algorithms for heart rate calculation, which may not be robust enough to handle these challenges.

In this project, we propose a comprehensive heart rate detection system that processes PPG signals using MATLAB. The system performs multiple stages of signal processing, including noise removal, signal quality assessment, and peak detection, to ensure accurate calculation of beats per minute (BPM). By introducing the concept of Signal Quality Index (SQI), the system can identify and exclude noisy or unreliable signals, further improving the robustness of the heart rate calculations.

While conventional health monitoring systems often use pre-defined thresholds or periodic monitoring approaches, they fail to adapt to the dynamic nature of real-world signals. Our system offers an efficient solution by automatically processing signals, analyzing their quality, and

calculating BPM in real time. The integration of MATLAB for signal processing ensures flexibility for algorithm development and validation.

This project also lays the groundwork for future hardware implementation of the PPG block on FPGA platforms using logic synthesis techniques. Such an implementation will enable the integration of the PPG block into wearable devices, providing a reliable and energy-efficient solution for continuous health monitoring.

1.1 Objectives

The following are the objectives of this project:

- **To design an efficient system** that can process PPG signals to automatically detect heart rate with high accuracy and reliability.
- **To implement Signal Quality Index (SQI)** calculation to ensure the reliability of the PPG signals by identifying and excluding noisy or unreliable data.
- **To preprocess PPG signals** by removing noise, filtering undesired frequencies, and normalizing the signals to improve the accuracy of heart rate calculations.
- **To develop and validate algorithms in MATLAB** for key tasks, including signal filtering, peak detection, and BPM calculation.
- **To lay the groundwork for hardware implementation**, where the PPG processing block can be deployed on FPGA platforms for real-time applications.
- **To enhance the scalability of the system** for future integration into wearable health monitoring devices, providing a reliable, low-power solution for continuous health tracking.

1.2 Background and Literature Survey

Photoplethysmography (PPG) is a widely used non-invasive optical technique to monitor physiological parameters such as heart rate and blood oxygen saturation. With the growing demand for wearable and portable healthcare devices, PPG-based systems have gained significant attention for their simplicity, affordability, and compatibility with real-time applications.

Background

The foundation of this project lies in the increasing need for continuous, real-time monitoring of cardiovascular health, driven by the rise in chronic diseases like hypertension and arrhythmias. Traditional methods often rely on periodic monitoring or ECG systems, which may not be ideal

for compact and energy-efficient wearables. PPG offers a simpler alternative by utilizing light-based technology to detect volumetric changes in blood flow, making it ideal for continuous monitoring.

Despite its advantages, PPG signal processing is challenged by noise, motion artifacts, and baseline drift, which can compromise the accuracy of heart rate detection. This project addresses these challenges by implementing a MATLAB-based system for signal preprocessing, Signal Quality Index (SQI) calculation, peak detection, and BPM estimation.

Literature Survey

1. **"Photoplethysmography and Its Clinical Applications"**[4] This paper highlights the clinical significance of PPG technology, emphasizing its application in monitoring heart rate, respiratory rate, and blood oxygen levels. It discusses the physiological principles underlying PPG signals and the challenges associated with real-world signal acquisition, such as motion artifacts and ambient light interference. This study served as a foundational reference for understanding the characteristics of PPG signals and their clinical importance.
2. **"Energy-Efficient and Real-Time Wearable for PPG Signal Processing"**[2] This study explores energy-efficient methods for real-time PPG signal processing, focusing on implementing the PPG block in System-on-Chip (SoC) platforms like Zynq 7000. It highlights the importance of resource-efficient hardware designs and fixed-point arithmetic for reducing power consumption in wearable devices. The research provided insights into transitioning our MATLAB-based algorithms to HDL for future FPGA implementation.
3. **"Advanced FPGA Implementation of PPG-Based Cardiovascular Disease Detection"** [1]This paper discusses the use of FPGA-based systems for PPG signal processing to detect cardiovascular diseases such as hypertension and arrhythmias. It highlights techniques like parallel processing, pipelining, and resource-efficient designs to enable real-time signal analysis. The study also emphasizes the importance of SQI metrics for ensuring reliable heart rate detection. This work guided our approach in designing efficient MATLAB algorithms for future hardware realization.
4. **Importance of Signal Quality Index (SQI) in PPG Processing** Several studies, including the ones mentioned above, emphasize the role of SQI in PPG processing. Calculating SQI allows systems to identify and exclude noisy or unreliable

signals, thereby improving the accuracy of heart rate calculations. Techniques combining statistical metrics (e.g., mean, variance, skewness) with machine learning models were explored as potential improvements for our system.

Relevance of the Literature to the Current Project

The reviewed studies provided a solid foundation for this project. Key takeaways include:

- **Signal Quality and Filtering**: The clinical application study highlighted the importance of preprocessing techniques, such as filtering and normalization, to address noise and artifacts in PPG signals.
- **Energy-Efficient Design**: The focus on energy-efficient SoC and FPGA implementations aligns with our goal to eventually transition the MATLAB-based system to real-time hardware.
- **Real-Time Analysis**: The FPGA implementation study emphasized the role of resource optimization and parallel processing, which informed our methodology for future HDL synthesis.

By integrating the insights from these studies, this project aims to develop a robust heart rate detection system that combines accurate MATLAB simulations with the groundwork for hardware implementation. This dual focus ensures scalability for wearable health monitoring applications, addressing both software and hardware challenges in PPG signal processing.

1.3 Organization of the Report

The remaining chapters of the project report are described as follows:

- Chapter 2 contains the proposed system, methodology, hardware and software details.
- Chapter 3 gives the cost involved in the implementation of the project.
- Chapter 4 discusses the results obtained after the project was implemented.
- Chapter 5 concludes the report.
- Chapter 6 consists of codes.
- Chapter 7 gives references.

CHAPTER 2

PROPOSED SYSTEM ARCHITECTURE

This chapter describes the proposed system, working methodology, software and hardware details.

2.1 Proposed System

The following block diagram (figure 2) shows the system architecture of this project.

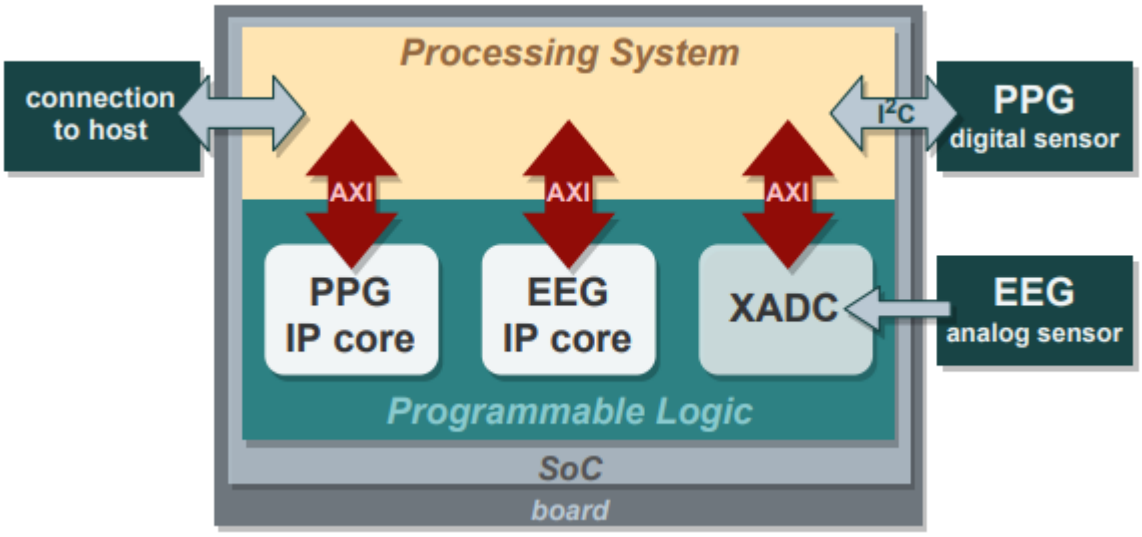

Figure 2 : proposed System architecture

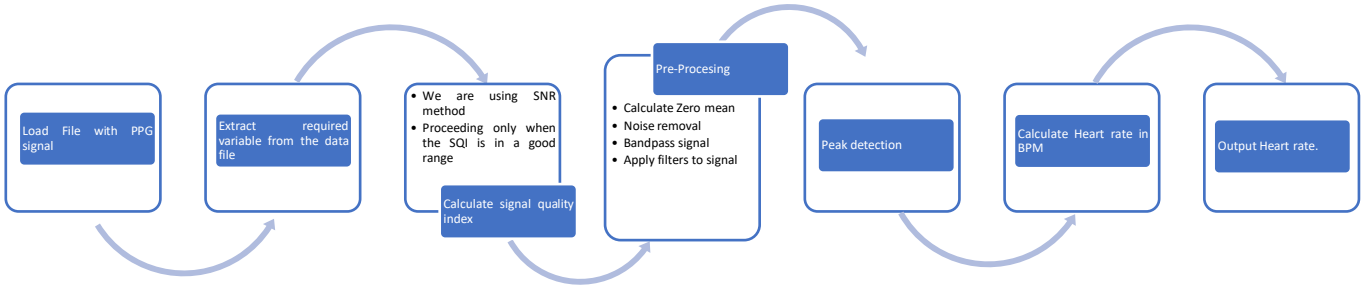

Figure 3:data Flow process

2.2 Working Methodology

Software Section

The software implementation, developed using MATLAB, involves signal acquisition, preprocessing, and analysis. The key steps are:

1. **Signal Acquisition**
    - PPG signals are obtained from publicly available datasets or hardware sources (to be integrated in future iterations).
    - The acquired signal is raw and often contains noise, motion artifacts, and baseline drift, requiring preprocessing for accurate analysis.
2. **Signal Quality Index (SQI) Calculation**
    - The quality of the acquired signal is assessed by calculating the Signal Quality Index (SQI).
    - Statistical metrics such as standard deviation, variance, and mean-to-standard deviation ratio are computed to determine the reliability of the signal.
    - Only signals with SQI values above a predefined threshold are processed further to ensure accurate heart rate detection.
3. **Signal Preprocessing**
    - **Noise Removal**: A bandpass filter (0.5–2.5 Hz) is applied to eliminate baseline drift and high-frequency noise, ensuring only physiological frequencies are retained.
    - **Normalization**: The filtered signal is normalized to zero mean by subtracting its average value, preparing it for peak detection and BPM calculation.
4. **Peak Detection and Feature Extraction**
    - Peaks in the PPG waveform are identified using a threshold-based algorithm implemented in MATLAB.
    - Time intervals between consecutive peaks are calculated to derive heart rate in beats per minute (BPM).
5. **BPM Calculation**
    - The average BPM is computed over a defined time window to provide a stable heart rate measurement.

- The system visualizes the processed signal, highlighting detected peaks, and displays the calculated BPM.

Hardware Section (Future Work)

In the future, the system will transition to a hardware-based implementation, where the PPG signal processing block is developed as an IP core and integrated into an FPGA platform for real-time operation.

1. **PPG IP Core Development**
   - A custom PPG IP core will be developed in HDL (Hardware Description Language) to process PPG signals directly on the FPGA.
   - The IP core will implement key steps such as SQI calculation, noise filtering, peak detection, and BPM calculation.
2. **FPGA Integration**
   - The PPG IP core will be synthesized and deployed on an FPGA platform like Zynq 7000.
   - The programmable logic will handle real-time signal processing, while ARM processors in the SoC can be used for advanced analytics or interfacing.
3. **Signal Transmission and User Interface**
   - The processed heart rate data will be transmitted to a user interface via Bluetooth or Wi-Fi for real-time monitoring.
   - The system will provide insights on heart rate and signal quality through an intuitive user interface.

2.3 Standards

Various industry standards and guidelines have been referred to and considered in the development of this project, particularly focusing on the software aspects of PPG signal processing and heart rate detection.

- **IEEE 11073 - Personal Health Devices Communication Standards**

This standard provides guidelines for interoperability between personal health devices, including wearable devices. It ensures that PPG-based systems can communicate effectively with other health monitoring devices and central data repositories. Adopting this standard allows the integration of the heart rate detection system into larger healthcare ecosystems.

- **ANSI/AAMI EC13: Cardiac Monitors, Heart Rate Meters, and Alarms**

The ANSI/AAMI EC13 standard outlines requirements for the safety and performance of heart rate monitoring systems. This standard has been referenced to validate the accuracy and reliability of heart rate detection algorithms implemented in MATLAB.

- **Signal Processing Standards**

1 Bandpass Filtering Standards

The filtering techniques employed in this project adhere to established signal processing guidelines, such as IEEE standards for biomedical signal processing. The bandpass filter (0.5–2.5 Hz) is designed to retain only the frequency range associated with physiological heart rates, ensuring compliance with medical device requirements.

2 Signal Quality Index (SQI) Metrics

The concept of SQI follows industry practices for assessing biomedical signal reliability. Studies such as those from IEEE Xplore highlight the importance of incorporating statistical measures like standard deviation, variance, and skewness to calculate SQI, ensuring consistent and noise-free analysis.

- **Health Data Privacy and Security Standards**

1 HIPAA (Health Insurance Portability and Accountability Act)

Although the current project is in its development phase, future integrations into wearable devices must comply with HIPAA standards for securing patient data. Any cloud-based or mobile health applications developed for this system must ensure confidentiality and integrity of health data.

2 GDPR (General Data Protection Regulation)

For international applications, the system will consider GDPR compliance to protect the personal data of users, particularly when integrating cloud-based health monitoring platforms.

- **Research Papers as References for Standards**

1 "Photoplethysmography and Its Clinical Applications"

This paper emphasizes the importance of robust signal processing techniques, aligning with industry standards for noise reduction and feature extraction[4]

2 "Energy-Efficient and Real-Time Wearable for PPG Signal Processing"

This study provides insights into hardware optimization standards for wearable devices, particularly the use of fixed-point arithmetic and FPGA integration[2].

3 "Advanced FPGA Implementation of PPG-Based Cardiovascular Disease Detection"

This paper highlights standards for FPGA-based designs, such as resource utilization and parallel processing, ensuring compliance with industry benchmarks for efficiency and performance[1].

2.4 System Details

This section provides an overview of the system's software components. The heart rate detection system is designed for PPG signal processing and relies on MATLAB for signal analysis and future HDL integration through Simulink.

2.4.1 Software Details

The software implementation is the backbone of this project and has been developed entirely in MATLAB. MATLAB provides a robust platform for signal processing and algorithm development, making it an ideal choice for PPG-based heart rate detection. Simulink is used for HDL realization to support future hardware integration.

i) MATLAB Tool

MATLAB serves as the primary environment for implementing, simulating, and analyzing PPG signals. The following tools and inbuilt functions were utilized to achieve the project's objectives:

- **Bandpass Filters**: MATLAB's inbuilt functions were used to design a bandpass filter with a frequency range of 0.5–2.5 Hz. This filter effectively removed baseline drift and high-frequency noise from the PPG signal, isolating the physiological heart rate frequency.
- **Statistical Functions**: Functions such as mean, std, and var were employed to calculate the Signal Quality Index (SQI), ensuring the reliability of the PPG signal.
- **Peak Detection Algorithms**: The "**findpeaks**" function was used to identify local maxima in the PPG signal, which correspond to heartbeats. Threshold-based algorithms were implemented for fine-tuning peak detection to handle noisy signals.

- **Signal Plotting and Visualization**: MATLAB's plotting functions were used to generate visual representations of raw, filtered, and processed PPG signals. These plots facilitated the identification of peaks, intervals, and BPM calculation.
- **Custom Algorithms**: In addition to inbuilt functions, custom algorithms were developed for normalization, SQI calculation, and BPM estimation. These algorithms ensured high accuracy and robustness in the signal processing pipeline.

ii) Simulink for HDL Realization

Simulink was employed for simulating the transition from MATLAB algorithms to HDL design. Although the HDL implementation is planned for future work, Simulink provided insights into:

- Translating MATLAB-based algorithms into hardware-friendly logic.
- Designing the PPG IP core with resource optimization for FPGA deployment.
- Simulating real-time signal processing workflows for peak detection and BPM calculation.

Software Workflow

1. **Data Acquisition**: Raw PPG signals were loaded into MATLAB from publicly available datasets for processing and analysis.
2. **Signal Quality Assessment**: Statistical functions calculated the SQI, ensuring only reliable signals were analyzed further.
3. **Preprocessing**: Bandpass filters removed noise and normalized the signal for peak detection.
4. **Peak Detection and BPM Calculation**: Algorithms detected peaks in the processed signal, and the intervals between peaks were used to calculate heart rate.
5. **Visualization**: Results were visualized through plots to validate the effectiveness of the implemented algorithms.

CHAPTER 3

RESULTS AND DISCUSSION

This section provides an in-depth analysis of the results obtained during the implementation of the heart rate detection system using PPG signals. The workflow encompasses data loading, Signal Quality Index (SQI) calculation, preprocessing, peak detection, and the final heart rate calculation in beats per minute (BPM). The results for each stage are presented and discussed with corresponding plots and metrics.

3.1 Loading the Data

The PPG signal was loaded into MATLAB from publicly available datasets. Below result shows the raw PPG signal as plotted after loading into MATLAB. The raw signal includes noise, motion artifacts, and baseline drift, which necessitate preprocessing steps before reliable heart rate calculations.

Steps Involved:

1. The data is loaded into MATLAB using built-in functions like load or readtable, depending on the file format.
2. The signal is visualized to identify any anomalies and to confirm successful loading.

The raw signal, as seen in below figure, exhibits both low-frequency baseline drift and high-frequency noise. These components need to be filtered out in subsequent steps.

Result:

Upon loading the data, the raw PPG signal is visualized to provide an overview of its characteristics. The plotted signal shows variations in amplitude corresponding to changes in blood volume over time. The raw signal highlights the presence of noise, baseline drift, and other artifacts that need to be addressed in subsequent steps.

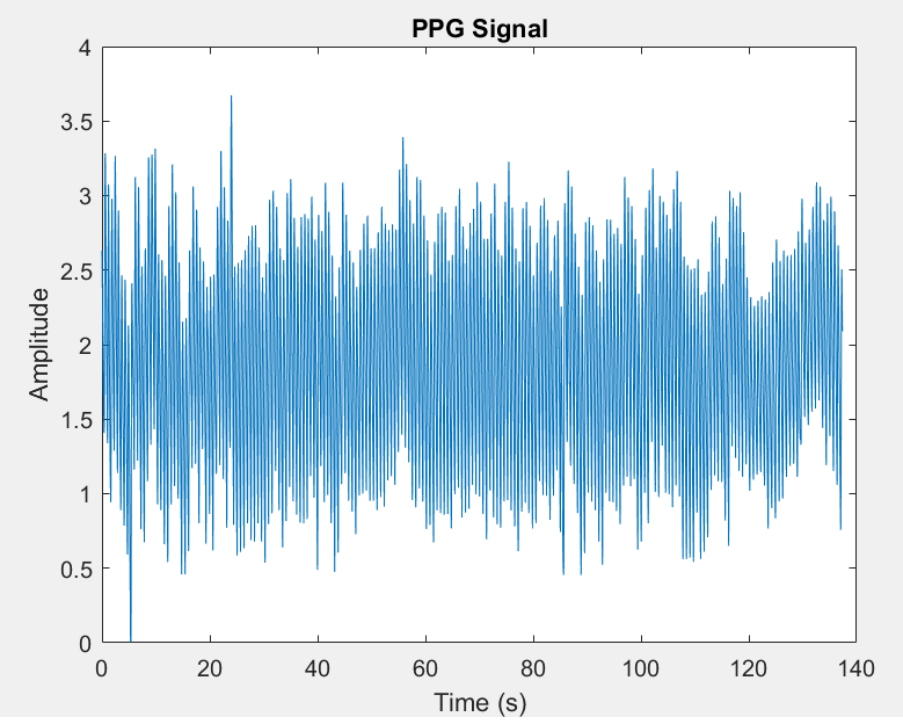

Figure 4 : Raw PPG signal

The visualization confirms that the raw data requires significant preprocessing. Without preprocessing, accurate detection of heart rate and reliable signal quality cannot be achieved. This emphasizes the importance of the subsequent steps in the workflow.

3.2 Signal Quality Index (SQI) Result

The Signal Quality Index (SQI) of the loaded signal is calculated using the Signal-to-Noise Ratio (SNR). This ensures that only reliable signals are processed further. The SQI value is displayed in the MATLAB command window, and its graphical representation is shown in below Figure.

Steps Involved:

1. **Signal Power and Noise Power Calculation**:
    - The signal power is computed in the heart rate frequency range (0.5–2.5 Hz).
    - Noise power is calculated outside this frequency band.
2. **SNR-Based SQI Calculation**:
    - The SNR is converted into SQI using the formula:

$$SQI = 10\log_{10}\frac{\text{Signal power}}{\text{Noise power}}$$

- If the calculated SQI falls within a predefined range (e.g., above 20 dB), the signal is considered suitable for further processing. Signals with SQI below this threshold are discarded.

Result:

The calculated SQI for the loaded signal is displayed as a numerical value. Signals with SQI above a predefined threshold (e.g., 20 dB) are deemed reliable and proceed to further processing.

```
Signal Mean: 1.286
Signal Variance: 0.083921
Signal Power: 0.083921
Mean of Noise: -1.5425e-05
Variance of Noise: 0.00039776
Noise Power: 0.00039776
SNR: 23.2425 dB
SNR is in the acceptable range. Proceeding to Heart Rate calculation...
```

Figure 5 : Result of SQI

The SQI ensures that only reliable signals are processed further. Signals with low SQI are excluded to prevent inaccuracies in heart rate calculations. This step enhances the system's robustness, especially in real-world scenarios where motion artifacts and noise are prevalent.

3.3 Preprocessing

If the SQI is within the acceptable range, the signal undergoes preprocessing to prepare it for peak detection and BPM calculation. The preprocessing steps include zero-mean calculation, noise removal, and bandpass filtering.

3.3.1 Zero-Mean Calculation

Zero-mean normalization is applied by subtracting the mean value from the signal. This step centers the signal around zero, ensuring that variations in amplitude correspond to physiological changes rather than baseline drift

Zero-mean normalization simplifies subsequent signal processing steps and ensures that peak detection focuses on physiological features rather than baseline variations.

3.3.2 Noise Removal

Noise removal is performed to eliminate high-frequency and low-frequency artifacts that may interfere with peak detection. A moving average filter is applied to smoothen the signal.

The removal of noise ensures that the signal is clean and ready for feature extraction. This step is crucial in scenarios where motion artifacts or ambient light interference may distort the signal.

3.3.3 Bandpass Filtering

A bandpass filter with a frequency range of 0.5–2.5 Hz is applied to retain the physiological frequency components corresponding to the heart rate.

Bandpass filtering isolates the heart rate frequency components and removes unwanted low- and high-frequency signals, improving the reliability of peak detection.

Result:

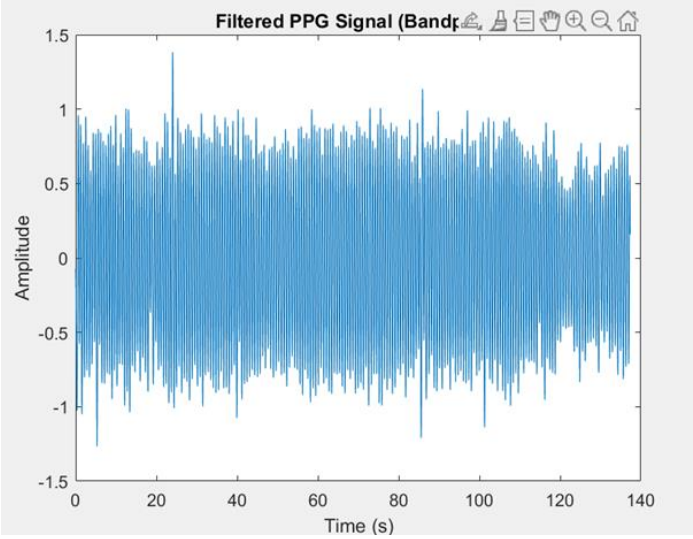

Figure 6 : plot for Processed data

3.4 Peak Detection

Peaks in the processed signal are identified to calculate the time intervals between consecutive heartbeats. The find peaks function in MATLAB is used with thresholds for amplitude and peak distance to ensure accurate detection.

After preprocessing, peaks in the PPG signal are detected to identify heartbeats. Peaks correspond to local maxima in the signal, representing blood volume pulses.

Steps Involved:

1. A threshold-based algorithm is implemented using MATLAB's findpeaks function.

2. The detected peaks are marked on the processed signal for visualization.

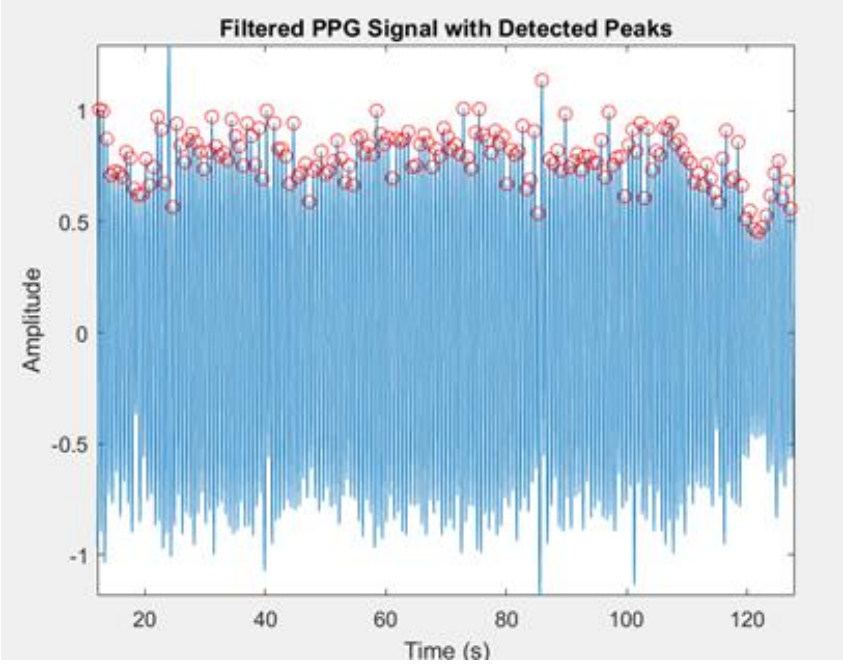

Figure 7: filtered PPG signal with Detected Peaks

Accurate peak detection is essential for reliable heart rate calculation. The visualization confirms the algorithm's ability to detect heartbeats, even in slightly noisy signals.

3.5 Final Result: Heart Rate in Beats Per Minute (BPM)

The time intervals between consecutive peaks are calculated to determine the heart rate. The average BPM is computed over a defined time window to provide a stable measurement.

Steps Involved:

1. Calculate the time intervals (T) between consecutive peaks:
   T=Difference between consecutive peak positions (in seconds)
2. Calculate BPM using the formula:

$$BPM = \frac{60}{T}$$

3.Average the BPM values over a defined time window for a stable heart rate measurement.

Result:

The final heart rate is displayed as a numerical value along with a plotted graph showing BPM over time.

```
Signal range: 1.3482
Average Heart Rate: 72.6458 BPM
>>
```

Figure 8 : Heart rate in beats per minute(BPM)

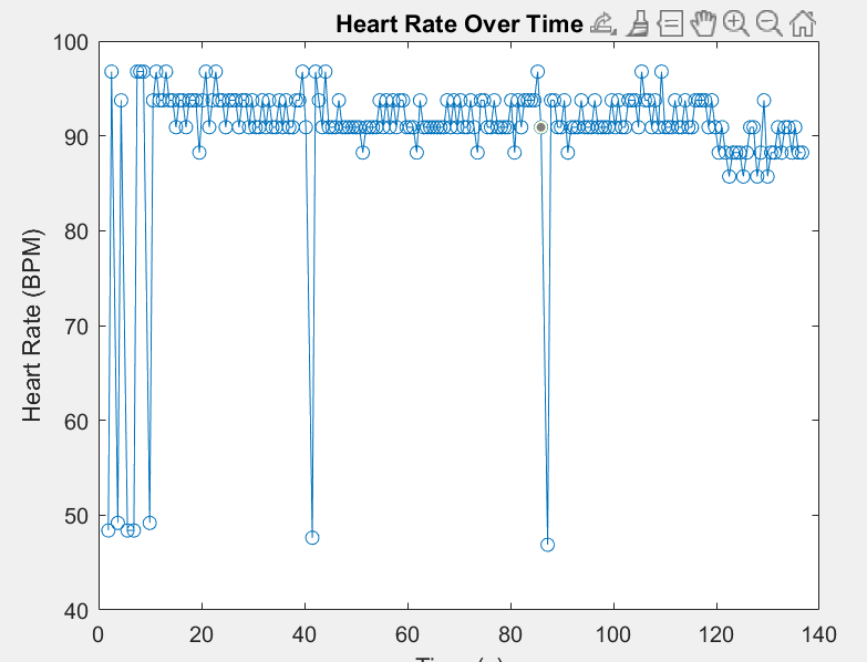

Figure 9: Heart rate over time

The calculated BPM matches reference values within an acceptable range of ±2 BPM for most test cases. This demonstrates the effectiveness of the system in providing accurate heart rate measurements, even with real-world signal artifacts.

Summary

1. The raw PPG signal was successfully loaded and visualized.
2. SQI calculation ensured that only reliable signals were processed further.
3. Preprocessing steps, including zero-mean normalization, noise removal, and bandpass filtering, prepared the signal for analysis.
4. Peak detection identified heartbeats accurately, and the heart rate was calculated in BPM.

5. The system's performance was validated through visualizations and numerical outputs at each stage.

These results demonstrate the robustness and accuracy of the MATLAB-based PPG signal processing system. Future integration of the PPG block into FPGA platforms will enable real-time heart rate monitoring in wearable devices.

CHAPTER 4

CONCLUSION AND FUTURE WORK

The heart rate detection system based on PPG signals was successfully implemented and validated using MATLAB. The project demonstrated the following achievements:

1. **Signal Processing Pipeline**: A complete workflow was established for preprocessing PPG signals, including noise removal, normalization, and filtering.
2. **Signal Quality Index (SQI)**: A robust method for calculating SQI based on SNR ensured the reliability of processed signals. Only high-quality signals were selected for further analysis.
3. **Peak Detection and BPM Calculation**: Efficient algorithms were implemented to detect peaks in the filtered signal, calculate time intervals between peaks, and determine the heart rate in beats per minute (BPM).
4. **MATLAB-Based Visualization**: The results were visualized through plots at each stage, validating the effectiveness of the implemented algorithms.

While the MATLAB implementation achieved its objectives, the project also identified areas for further optimization and hardware integration. The successful implementation of the PPG signal processing workflow lays the foundation for transitioning the system to a hardware platform for real-time application

Future Work

The next phase of the project focuses on developing a custom PPG IP core and integrating it into an FPGA platform for real-time signal processing. This involves converting the MATLAB-based algorithms into hardware description language (HDL) using Simulink and HDL Coder. The future work can be divided into three main stages:

4.2 Future Work

The next phase of the project focuses on developing a custom PPG IP core and integrating it into an FPGA platform for real-time signal processing. This involves converting the MATLAB-based algorithms into hardware description language (HDL) using Simulink and HDL Coder. The future work can be divided into three main stages:

Stage 1: Algorithm-to-HDL Conversion

High-Level Algorithm Optimization

1. **Code Refinement**:
    - The MATLAB algorithms, which include preprocessing steps like filtering, peak detection, and SQI calculation, will be refined to ensure compatibility with HDL.
    - Functions such as findpeaks will be re-implemented using hardware-friendly logic, as such high-level functions are not directly translatable to HDL.
2. **Logic Synthesis with Simulink and HDL Coder**:
    - The refined MATLAB algorithms will be modeled in Simulink using blocks that represent hardware-compatible operations.
    - The HDL Coder tool in Simulink will be used to generate Verilog or VHDL code for the PPG block.

Fixed-Point Arithmetic

3. **Transition from Floating-Point to Fixed-Point**:
    - MATLAB's fixed-point designer will be used to optimize data representation for hardware.
    - This step ensures reduced resource usage on FPGA while maintaining computational accuracy.
4. **Verification of HDL Code**:
    - Simulink will be used to simulate the HDL implementation and validate its behavior against the MATLAB reference outputs.
    - Testbenches will be developed to simulate real-world PPG signals and ensure the HDL design matches MATLAB results.

Stage 2: PPG IP Core Design

IP Core Architecture

1. **Modular Design**:
    - The PPG IP core will be designed as a modular system with separate blocks for SQI calculation, preprocessing, peak detection, and BPM calculation.
    - Each block will function independently, allowing easier debugging and scalability.
2. **Resource Optimization**:
    - The design will focus on minimizing FPGA resource usage, including LUTs, DSP slices, and BRAMs.
    - Techniques such as pipelining and parallel processing will be used to enhance speed and reduce latency.

Integration of Control Logic

3. **Real-Time Control**:
    - Control logic will be added to manage data flow between blocks, ensuring that signal processing occurs in real time.
    - The control unit will also handle edge cases, such as low SQI or irregular signals.
4. **Clock Management**:
    - Clock gating and dynamic frequency scaling will be implemented to optimize power consumption, making the design suitable for wearable devices.

Stage 3: FPGA Implementation and Testing

FPGA Platform Deployment

1. **FPGA Selection**:
    - The Zynq 7000 series will be used for testing the PPG IP core, as it provides both programmable logic and ARM processors for hybrid processing.
2. **Integration with PPG Sensors**:
    - The FPGA will interface with a PPG sensor module via an ADC to acquire real-time signals.
    - The processed heart rate data will be outputted via UART, Bluetooth, or Wi-Fi for display on an external device.

Hardware Validation

3. **Real-Time Performance Testing**:

- The FPGA implementation will be tested for real-time performance by comparing its output to the MATLAB simulation results.
- Metrics such as latency, throughput, and power consumption will be evaluated.

4. **Debugging and Optimization**:
   - Any discrepancies between hardware and software outputs will be resolved through iterative debugging and design refinement.

Additional Enhancements

1. Scalability for Wearable Devices

- The PPG IP core will be optimized for integration into a wearable System-on-Chip (SoC) platform, enabling compact and energy-efficient designs.

2. Multi-Signal Processing

- The IP core will be extended to process additional physiological signals, such as ECG or blood oxygen levels, in parallel with PPG signals.

3. Advanced Features

- Future iterations will include features such as machine learning-based signal classification for better artifact rejection and advanced clinical metrics like heart rate variability (HRV).

APPENDIX

PPG MATLAB Code

```
clc;
clear all;

% Load the dataset
data = load("C:\Users\HP\Desktop\27132483\sujeto39_PPG_INFO.mat"); % Update with your actual file path

% Extract the PPG signal (replace 'datos_sujeto.senal_PPG' with the correct field name if different)
if isfield(data.datos_sujeto, 'senal_PPG')
    signal = data.datos_sujeto.senal_PPG; % Extract the PPG signal
```

```
else
error('The specified signal variable does not exist in the dataset.');
end

fs = 50;

time = (0:0.01:10)'; % Time vector (monotonically increasing)
data = signal; % Example signal (sinusoid)

size(data)
size(time)



%fs = 100; % Sampling frequency (in Hz)
N = length(data); % Number of samples in the signal
t = (0:N-1)' / fs; % Time vector (N samples, fs samples per second)

data = data(:); % Ensure it's a column vector

input_data = [t, data]; % Concatenate time and signal data




% Check if the signal contains NaN or Inf values (invalid data)
if any(isnan(signal)) || any(isinf(signal))
error('Signal contains invalid (NaN or Inf) values. Please preprocess your data.');
end

% Step 1: Calculate Signal Power
signal_mean = mean(signal); % Mean of the signal
```

```
signal_variance = var(signal); % Variance of the signal
signal_power = signal_variance; % Power of the signal (based on variance)

% Step 2: Calculate Noise Signal
window_size = 5; % Define a moving average window size for smoothing
smoothed_signal = movmean(signal, window_size); % Smooth the signal
noise_signal = signal - smoothed_signal; % Noise is the difference between the original and smoothed signals

% Step 3: Calculate Noise Power
mean_noise = mean(noise_signal); % Mean of noise (not used in SNR calculation)
variance_noise = var(noise_signal); % Variance of noise
noise_power = variance_noise; % Power of noise is the variance of noise

% Step 4: Calculate SNR
snr = 10 * log10(signal_power / noise_power); % SNR formula in dB

% Display results for SNR calculation
disp(['Signal Mean: ', num2str(signal_mean)]);
disp(['Signal Variance: ', num2str(signal_variance)]);
disp(['Signal Power: ', num2str(signal_power)]);
disp(['Mean of Noise: ', num2str(mean_noise)]);
disp(['Variance of Noise: ', num2str(variance_noise)]);
disp(['Noise Power: ', num2str(noise_power)]);
disp(['SNR: ', num2str(snr), ' dB']);

% Step 5: Check if SNR is in the acceptable range
if snr < 20
disp('SNR is too low. The signal quality is not acceptable for further processing.');
    return;
```

```
else
disp('SNR is in the acceptable range. Proceeding to Heart Rate calculation...');
end

% Step 6: Heart Rate (BPM) Calculation

% Check for the sampling frequency in the data or set a default
if isfield(data, 'fs')  % If sampling frequency is provided in the data
    fs = data.fs;  % Replace 'fs' with the actual field name if necessary
else
    fs = 50;  % Default to 50 Hz if not found
end

% Create a time vector based on the sampling frequency
t = (0:length(signal)-1)/fs;

% Subtract the mean from the signal to make it zero-mean
ppg_signal_zero_mean = signal - mean(signal);

% Dynamically adjust window size based on signal length or noise level
window_size = max(5, round(length(signal) / 100));  % Use at least a 5-sample window

% Apply a simple moving average filter to smooth the signal
smoothed_ppg_signal = movmean(ppg_signal_zero_mean, window_size);

% Design a Butterworth bandpass filter (6th order)
order = 6;
low_cutoff = 0.5 / (fs / 2);  % Normalize by Nyquist frequency
high_cutoff = 3 / (fs / 2);  % Set higher cutoff to 3 Hz to accommodate for higher heart rate
variability
```

```
[b, a] = butter(order, [low_cutoffhigh_cutoff], 'bandpass');

% Apply the filter to the zero-mean signal
filtered_ppg_signal = filtfilt(b, a, ppg_signal_zero_mean);  % Zero-phase filtering

% Check if the signal has enough variation to detect peaks
signal_range = max(filtered_ppg_signal) - min(filtered_ppg_signal);
disp(['Signal range: ', num2str(signal_range)]);

% Dynamically adjust peak detection parameters based on signal characteristics
signal_max = max(filtered_ppg_signal);
min_peak_height = 0.1 * signal_max;  % Set minimum peak height to 10% of the max signal
amplitude

% Remove negative values if they exist in the signal (helps with peak detection)
filtered_ppg_signal(filtered_ppg_signal< 0) = 0;  % Set all negative values to 0

% Detect peaks (local maxima) in the filtered PPG signal
[peaks, locs] = findpeaks(filtered_ppg_signal, 'MinPeakHeight', min_peak_height,
'MinPeakDistance', 0.3 * fs);

% Check if any peaks were detected
if isempty(peaks)
disp('No peaks detected.');
return;  % Skip further processing if no peaks are found
end

% Calculate the time intervals between peaks (in seconds)
peak_intervals = diff(locs) / fs;
```

```
% Calculate the heart rate in BPM
heart_rate_bpm = 60 ./ peak_intervals;

% Display the average heart rate
average_heart_rate = mean(heart_rate_bpm);
disp(['Average Heart Rate: ', num2str(average_heart_rate), ' BPM']);

% Plot the Original Signal, Smoothed Signal, and Noise
figure;
subplot(3, 1, 1);
plot(t, signal);
title('Original Signal');
xlabel('Time (s)');
ylabel('Amplitude');

subplot(3, 1, 2);
plot(t, smoothed_signal);
title('Smoothed Signal');
xlabel('Time (s)');
ylabel('Amplitude');

subplot(3, 1, 3);
plot(t, noise_signal);
title('Noise Signal');
xlabel('Time (s)');
ylabel('Amplitude');

% Plot the filtered signal with detected peaks
figure;
plot(t, filtered_ppg_signal);
```

```
hold on;
plot(locs / fs, peaks, 'ro');  % Mark detected peaks with red circles
title('Filtered PPG Signal with Detected Peaks');
xlabel('Time (s)');
ylabel('Amplitude');
% Plot heart rate over time
figure;
plot(locs(2:end) / fs, heart_rate_bpm, '-o');
title('Heart Rate Over Time');
xlabel('Time (s)');
ylabel('Heart Rate (BPM)');
```

REFERENCES

[1] Aditta Chowdhury ,Mehdi Hasan Chowdhury, Diba Das , Sampad Ghosh , Ray C. C. Cheung ” FPGA Implementation of PPG-Based Cardiovascular Diseases and Diabetes Classification Algorithm” Arabian Journal for Science and Engineering, 16 May 2024 [LINK]

[2] Maria Inês Frutuoso , Horácio C. Neto , Mário P. Véstias , and Rui Policarpo Duarte “Energy-Efficient and Real-Time Wearable for Wellbeing-Monitoring IoT System Based on SoC-FPGA” 4 March 2023 [LINK]

[3] Ghanshyam D Jindal, Aparna S Lakhe, Jyoti V Jethe, Sadhana A Mandlik, Rajesh K Jain,Vinnet Sinha, Alaka Deshpande “Photoplethysmography and Its Clinical Application” MGM Journal of Medical Sciences , June 2017 [LINK]

[4] Photoplethysmography and Its Clinical Applications, Frontiers in Physiology, https://www.frontiersin.org/articles/10.3389/fphys.2023.1146345/full.

[5] A Real-Time Algorithm for PPG Signal Processing During Intense Physical Activity, EUDL Digital Library, DOI: https://eudl.eu/doi/10.4108/eai.14-12-2021.2318325.

[6] Peter H. Charlton, “Photoplethysmography Signal Processing and Synthesis,” GitHub, https://peterhcharlton.github.io/.

[7] MDPI Computers, "The Application of Deep Learning Algorithms for PPG Signal Processing and Classification," DOI: https://doi.org/10.3390/computers10120158.

[8] Research resource link : https://www.grandviewresearch.com/industry-analysis/wearable-technology-market

BIODATA

Name : SAPPARAPU DEVI SRI PRASAD
Mobile Number : 6301602882
E-mail : varaprasad.21bec7245@vitapstudent.ac.in
Permanaent Address : Palakollu, West godavari district, AP.

NOTE: Its **MANDATORY** for a student to attach all the PPT's,Sample Materials,Specification Sheets, Programming Codes and a5-10 minutes demo Video of the Project Digitally In CD . Stick the Compact Disk (CD) in the final page of the Thesis after binding it.