Real-time Heart Monitoring and ECG Signal Processing

Fatima Bamarouf, Claire Crandell, and Shannon Tsuyuki

Advisors: Drs. Yufeng Lu and Jose Sanchez

Department of Electrical and Computer Engineering
Bradley University
February 29, 2016
Contents

• Introduction and Overview
• Progress
• Summary and Conclusions
Introduction and Overview

• Problem Description
• Project Objectives
• System Block Diagram
• Division of Labor
Problem Description

• Arrhythmias
  • Are irregular heartbeats caused by defective electrical signals in the heart [1]
  • Include premature ventricular contractions (PVCs)
  • PVCs may lead to ventricular tachycardia (VT)

Figure 1. Electrocardiogram with “V” labels for PVCs [2]
Problem Description

- An electrocardiogram (ECG) describes the heart’s electrical activity
- An ECG can be recorded using a Holter monitor or event monitor

Figure 2. Features of a normal ECG [3]
Problem Description

• Holter and event monitors are limited in functionality
  • Utilize some in-platform signal processing for diagnostic assistance
  • Must perform some signal processing offline
  • Are unable to address medical issues in real time
Figure 3. Overall heart monitoring system diagram
Project Objectives

• Develop a low-power, stand-alone embedded system for continuous heart monitoring that will
  • Process ECG data in real time
  • Detect PVCs accurately and consistently
  • Alert the patient’s doctor wirelessly of ventricular tachycardia
Project Objectives

• Real-time ECG signal processing
• On-board signal processing computations
• Battery-powered functionality
Division of Labor

• MATLAB Simulation (PVC detection)
  • Shannon/Fatima

• C Programming (PVC detection)
  • Claire/Shannon

• Wi-Fi Communication
  • Fatima/Claire/Shannon
Contents

• Introduction and Overview

• Progress
  • Fatima
  • Shannon
  • Claire

• Summary and Conclusions
Problem Approach

Figure 4. High-level flowchart for algorithms

1. Raw Data
2. Pan_Tompkins Algorithm (peaks detection)
3. QRS Data
4. Template Matching Algorithm (PVC detection)
5. PVC Data
6. Three consecutive PVCs (Tachycardia positive)
7. Send Wireless Message
Template Generation Algorithm [4]

QRS Data (30s)

Beat Amplitude Series
  Remove Extremities ($\sigma$)
  Sort In Ascending Order
  Divide Into 5 Subsets With Equal Amplitude Range
  Keep The Set With Max Of Data Points
  T1 & T2 Templates

Beat To Beat Time Series
  Remove Extremities ($\sigma$)
  Sorted In Ascending Order
  Divide Into 5 Subsets With Equal Time Duration
  Keep The Set With Max Of Data Points

Figure 5. Flowchart for T1 and T2 generation
Beat Amplitude Series

Figure 6. MATLAB plot of QRS detection results. The marked peaks were placed into an array.
Template Generation Algorithm

QRS Data (30s)

Beat To Beat Time Series

Beat Amplitude Series

Remove Extremities (3Ϭ)

Sort In Ascending Order

Divide Into 5 Subsets With Equal Amplitude Range

Keep The Set With Max Of Data Points

T1 & T2 Templates

Beat To Beat Time Series

Remove Extremities (3Ϭ)

Sorted In Ascending Order

Divide Into 5 Subsets With Equal Time Duration

Keep The Set With Max Of Data Points

Figure 7. Flowchart for T1 and T2 generation
Figure 8. Flowchart for T1 and T2 generation
Figure 9. Flowchart for T1 and T2 generation
Template Generation Algorithm

Figure 10. Flowchart for T1 and T2 generation
Figure 11. Flowchart for T1 and T2 generation
Figure 12. MATLAB plot of QRS detection results. The marked RR-intervals were placed into an array.
Template Generation Algorithm

Figure 13. Flowchart for T1 and T2 generation
Templates T1 and T2

Figure 14. T1 and T2 generation
Template-Matching Algorithm

Figure 15. Flowchart for template-matching algorithm
Template-Matching Algorithm

Figure 16. Flowchart for template-matching algorithm
Normalized Correlation Coefficient

\[ x_k = \frac{\sum_{n=0}^{L-1} [b_k(n) - \bar{b}_k][N(n) - \bar{N}]}{\sqrt{\sum_{n=0}^{L-1} [b_k(n) - \bar{b}_k]^2 [N(n) - \bar{N}]^2}} \]

where \( b_k(n) \) is the QRS complex in the \( k^{th} \) beat and \( L \) is the length of the predetermined T1 (N(n)).
Template-Matching Algorithm

QRS Data (30min)

k-th Beat Aligned to T1

Correlation of Aligned k-th Beat to Template T1

Stretch/Compress k-th to k+1 Data Vector to Aligned with T2

Correlation of Data Between Two Fiducial Points to T2

Combine Both Correlation Using Exponential Function Correlation

Compare the Correlation result to the k-th Beat Aligned to T1 Threshold

PVC or not PVC

Figure 17. Flowchart for template-matching algorithm
Exponential Function Correlation

\[
z_k = f(x_k, y_k) = \frac{(e^{x_k r} + e^{y_k r})}{2e}
\]

where \( r \) determines the increasing rate of the slope and \( z_k \geq z_{thre} \).
CC3200 WiFi Setup

• The CC3200 was set up in station mode to access the internet

Figure 18. CC3200 set up in station mode [5]
Wireless Research

- Wireless platforms tested
  - Temboo
  - PubNub
Temboo

- Middleware that allows different devices (such as the LaunchPad) to access web-based services

Figure 19. Temboo integration with an embedded device [6]
Energia and Code Composer Studio

• Wireless platforms (such as Temboo) use Energia sketches for C++ functionality
• Energia is a rapid prototyping platform for the Texas Instruments MCU Launchpad (CC3200)
Energia and Code Composer Studio

• To be able to load the Energia sketch code from CCS, the libraries were changed to be compatible with Energia’s GNU compiler
  • Hardware libraries
  • DSP library
Code Optimization

• CC3200 has only 256kb of RAM
• The template matching algorithm requires storing 20 seconds of heart data on-board
• Instead, we obtained used a simple average to find a suitable template:
  \[ 0.9 \times \text{average} \leq \text{template} \leq 1.1 \times \text{average} \]
**TABLE I. PERFORMANCE OF TEMPLATE-MATCHING MATLAB SIMULATION**

<table>
<thead>
<tr>
<th>Record</th>
<th>PVC Sensitivity</th>
<th>PVC Positive Predictivity</th>
</tr>
</thead>
<tbody>
<tr>
<td>106</td>
<td>0.729</td>
<td>1.000</td>
</tr>
<tr>
<td>116</td>
<td>1.000</td>
<td>0.973</td>
</tr>
<tr>
<td>119</td>
<td>1.000</td>
<td>0.998</td>
</tr>
<tr>
<td>201</td>
<td>1.000</td>
<td>0.789</td>
</tr>
<tr>
<td>203</td>
<td>1.000</td>
<td>0.651</td>
</tr>
<tr>
<td>208</td>
<td>1.000</td>
<td>0.892</td>
</tr>
</tbody>
</table>
Past Schedule

Figure 20. Gantt chart for fall semester
Figure 21. Gantt chart through the end of the project
Contents

• Introduction and Overview

• Progress
  • Fatima
  • Shannon
  • Claire

• Summary and Conclusions
Template-Matching Algorithm

Figure 22. Flowchart for template-matching algorithm
Figure 23. T2 (interval between R peaks) and RR-interval signal for PVC beat
Interpolation and Extrapolation

• Let $b_k$ be the data between $k^{th}$ beat and $k + 1$ beat.

• If $\text{length}(b_k) \leq \text{Length}(T2)$, we do extrapolation based on

\[
\text{new } b_k = b_k [1 + \alpha (n - 1)] \quad (n = 1, 2, \ldots, L) 
\]

\[
\alpha = \frac{\text{length}(b_k)}{\text{length}(T_2)}
\]
Example (Extrapolation)

\[ L_k = 10, \quad L = 21, \quad \alpha_k = \frac{10}{21} \approx 0.476 \]

**TABLE II. INDICES AFTER EXTRAPOLATION**

<table>
<thead>
<tr>
<th>( n ) value</th>
<th>Calculated Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1.48</td>
</tr>
<tr>
<td>3</td>
<td>1.95</td>
</tr>
<tr>
<td>4</td>
<td>2.43</td>
</tr>
<tr>
<td>5</td>
<td>2.90</td>
</tr>
</tbody>
</table>
Alternative QRS Detection [7]

- Uses the cumulative sum (CUSUM) test
  - Improved QRS onset/offset detection through statistics
  - Floats above noise level
Alternative QRS Detection

\[
ECG_e(k) = \frac{ECG(k)}{\sqrt{ECG^2(k) + ECG_H^2(k)}}
\]

where \( ECG_H \) is the Hilbert transform of the ECG signal.
Alternative QRS Detection

\[ AS = 2(EGG'_e(k))^2 \]

where \( ECG'_e \) is the derivative of the envelope signal
Figure 24. MATLAB plot of the raw and auxiliary signals generated using the alternative QRS detection algorithm.
Alternative QRS Detection

- Use a search window and determine a threshold
- If calculated average exceeds threshold, mark sample as point of change
Alternative QRS Detection

Figure 25. MATLAB plot of the auxiliary signal and the cumulative average
Wireless - Plotly

• Code language: JavaScript Object Notation (JSON)
  • Language independent
  • Based on JavaScript, C/C++/C#, Python, Perl, etc.
Figure 26. Sample Plotly graph of 1000 samples of heart data
Past Schedule

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm assessment (accuracy, sensitivity,</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>specificity)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code the PVC algorithm in C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alternative T1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>construction method</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 27. Gantt chart for fall semester
Current/Future Schedule

Figure 28. Gantt chart through the end of the project
Contents

• Introduction and Overview

• Progress
  • Fatima
  • Shannon
  • Claire

• Summary and Conclusions
Figure 29. Flowchart for T1 and T2 generation
Testing of QRS Detection Phase

- Pan-Tompkins algorithm implemented on CC3200

Figure 30. CC3200 Launchpad [8]
Figure 31. UART data flow between the PC and the CC3200
Figure 32. Pan-Tompkins block diagram
Pan-Tompkins Algorithm

- Raw ECG Data
- Low-Pass Filter
- High-Pass Filter
- Derivative
- Squaring Function
- Moving-Window Integration
- QRS Complex Data

Figure 33. Flowchart for Pan-Tompkins algorithm
## Results

### TABLE III. PERFORMANCE OF PAN-TOMPKINS C IMPLEMENTATION

<table>
<thead>
<tr>
<th>Record</th>
<th>QRS Sensitivity (%)</th>
<th>QRS Positive Predictivity (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>97.83</td>
<td>100.00</td>
</tr>
<tr>
<td>102</td>
<td>96.69</td>
<td>100.00</td>
</tr>
<tr>
<td>103</td>
<td>99.43</td>
<td>100.00</td>
</tr>
<tr>
<td>105</td>
<td>99.76</td>
<td>99.76</td>
</tr>
<tr>
<td>106</td>
<td>96.67</td>
<td>99.69</td>
</tr>
</tbody>
</table>
C Implementation Results

Figure 34. T1 (QRS complex) template generated on the CC3200
C Implementation Results

Figure 35. T2 (RR interval) template generated on the CC3200
Wavelet Transform Algorithm [9]

• Alternative QRS detection method
• Zero crossings of wavelet transform used to find QRS onset and offset
Wavelet Transform Algorithm

• First-order wavelet transform:

\[ y(n) = \left( \frac{1}{1.5} \right) \ast (-2 \ast x(n) + 2 \ast x(n - 1)) \]
Wavelet Transform Algorithm

Figure 35. QRS onsets and offsets detected using the wavelet transform
Temboo/Twilio

- Sending an SMS message using the CC3200 LaunchPad
- Message includes text and image file
Figure 36. Transmitting an SMS message using the LaunchPad.
System Integration

• Combined project uses Energia sketch in Code Composer Studio
• GNU compiler replaced ARM compiler
• Mixed C/C++ code
System Integration

- Issues addressed
  - Memory configuration
  - Synchronization with PC

Figure 37. Improved UART testing system
Past Schedule

Figure 39. Gantt chart for fall semester
Current/Future Schedule

Figure 40. Gantt chart through the end of the project
Contents

• Introduction and Overview
• Progress
  • Fatima
  • Shannon
  • Claire
• Summary and Conclusions
Summary and Conclusions

• PVCs are irregular heartbeats that may lead to VT

• An embedded device is proposed that will detect PVCs in real time and wirelessly alert the patient’s doctor of VT
Summary and Conclusions

• Design must include real-time ECG signal processing, on-board signal processing computations, and battery-powered functionality
Summary and Conclusions

• Current/Future Tasks
  • Testing the integrated algorithm/wireless code
  • Adding additional improvements to the MATLAB simulation if necessary
Real-time Heart Monitoring and ECG Signal Processing

Fatima Bamarouf, Claire Crandell, and Shannon Tsuyuki

Advisors: Drs. Yufeng Lu and Jose Sanchez

Department of Electrical and Computer Engineering
Bradley University
February 29, 2016
References


Removing Extremities (3σ)

• An amplitude \( A \) is kept if it fulfills the criterion

\[
\bar{A} - 3\sigma \leq A \leq \bar{A} + 3\sigma
\]

where \( \bar{A} \) is the mean of the amplitude range
T1 Alignment

Figure 16. T1 template and PVC QRS complex aligned to T1’s R peak index
Equal Sets

- sort_series=[0 1 1.5 1.7 2 3 5 5 7 9 10]
- Step=(Max-Min)/5=2
- Five equal sets: [0 1 1.5 1.7 2] [3] [5 5] [7] [9 10]
- Set one has maximum data points
Converting MATLAB to C

- Template-Matching Functions
  - normIntRange.m
  - normal_amplt.m
  - Corr_Coeff.m
  - Z_Corr.m
Conversion of MATLAB Code

• Conversion of template-matching MATLAB code to C code
  • Normal_amplt.m
  • Integrated code project
State Diagram

Start

Store heart data into memory

Transmit a message to the doctor (for VT)

Determine if VT is present

Perform preprocessing

Classify each beat as PVC or non-PVC

Figure 43. State diagram for heart monitoring system
Detailed Gantt Chart (1)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Classification function z(k)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Determination of threshold</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>value z(thre)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Algorithm assessment</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(accuracy, sensitivity,</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>specificity)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance (time to run the</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>code, estimation in MATLAB)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 44. Gantt chart for the MATLAB simulation (PVC algorithm) phase of the project
Detailed Gantt Chart (2)

Figure 45. Gantt chart for the C implementation (PVC algorithm) phase of the project
Figure 46. Gantt chart for the wireless development phase of the project
Specificity and Sensitivity [8]

\[
SP = \frac{TN}{TN + FP} \quad SE = \frac{TP}{TP + FN}
\]

- TP (True Positive): detected QRS complex that is present in the signal
- TN (True Negative): data point between QRS complexes that does not contain a QRS peak
- FP (False Positive): incorrect identification of QRS peak
- FN (False Negative): QRS peak that was not detected by the algorithm
Memory Requirements

• Sampling rate for ECG signal (MIT-BIH arrhythmia database): 360 Hz
• Number of samples required for 30 seconds of ECG data: 10,800
• Amount of memory required: 21 kB
Nonfunctional Requirements: Metrics

Objective: The device should be compatible with all patient data in the MIT-BIH database. [3]

Metric:
- Highly compatible: 10 points
- Very compatible: 7.5 points
- Compatible: 5.0 points
- Somewhat compatible: 2.5 points
- Not compatible: 0 points
Nonfunctional Requirements: Metrics

Objective: The device should be portable.

Metric:
- Very easy to carry around: 10 points
- Easy to carry around: 7.5 points
- Portable: 5.0 points
- Uncomfortable to carry around: 2.5 points
- Difficult to carry around: 0 points
Nonfunctional Requirements: Metrics

TABLE VI. QUANTITATIVE PERFORMANCE LEVELS FOR REAL-TIME HEART MONITORING [8,9]

<table>
<thead>
<tr>
<th>Power Consumption in 24 Hours of Continuous Use (W)</th>
<th>Price ($)</th>
<th>Value Scaled</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.50</td>
<td>500</td>
<td>10</td>
</tr>
<tr>
<td>2.50</td>
<td>600</td>
<td>7.5</td>
</tr>
<tr>
<td>3.25</td>
<td>700</td>
<td>5</td>
</tr>
<tr>
<td>4.00</td>
<td>800</td>
<td>2.5</td>
</tr>
<tr>
<td>4.75</td>
<td>900</td>
<td>0</td>
</tr>
</tbody>
</table>
TABLE V. MORPHOLOGICAL CHART FOR HEART MONITORING SYSTEM [10,11,12,13,14,15,16]

<table>
<thead>
<tr>
<th>Functions</th>
<th>Means</th>
<th>Means</th>
<th>Means</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storing heart data</td>
<td>Flash memory</td>
<td>RAM</td>
<td></td>
</tr>
<tr>
<td>Preprocessing (Filtering/QRS detection)</td>
<td>Pan-Tompkins</td>
<td>Wavelet transform</td>
<td>Wavelet transform and Pan-Tompkins</td>
</tr>
<tr>
<td>PVC detection</td>
<td>Wavelet transform</td>
<td>Template matching</td>
<td>RR-interval</td>
</tr>
<tr>
<td>Ventricular tachycardia detection</td>
<td>Three or more consecutive PVCs</td>
<td>Three or more consecutive PVCs, heart rate greater than 100 beats per minute</td>
<td>Statistical analysis</td>
</tr>
<tr>
<td>Wireless functionality</td>
<td>eZ430-RF2500</td>
<td>CC2540 (Bluetooth)</td>
<td>CC3200</td>
</tr>
</tbody>
</table>
Design Evaluation: Design Alternatives

• Total design space: 162 designs
• Two designs analyzed in detail
## TABLE IV. PROJECT SCHEDULE

<table>
<thead>
<tr>
<th>Task</th>
<th>Duration (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PVC Algorithm (MATLAB)</td>
<td>65</td>
</tr>
<tr>
<td>PVC Algorithm (C)</td>
<td>100</td>
</tr>
<tr>
<td>Wi-Fi Communication</td>
<td>150</td>
</tr>
<tr>
<td>Progress Report I</td>
<td>80</td>
</tr>
<tr>
<td>Progress Report II</td>
<td>80</td>
</tr>
<tr>
<td>Final Presentation</td>
<td>80</td>
</tr>
<tr>
<td>Final Report</td>
<td>80</td>
</tr>
</tbody>
</table>
Schedule

Figure 47. Gantt chart for the fall semester
Figure 48. Gantt chart for the spring semester
Alternative Solution: Hardware

- eZ430-RF2500 (Texas Instruments)
  - MSP430F2274 MCU
  - CC2500 wireless transceiver
  - 32 kB flash memory

Figure 18. eZ430-RF2500 Development Kit [12]
Alternative Solution: Software

• PVC detection
  • Wavelet transform algorithm [13]
  • RR-interval algorithm [14]
Pan-Tompkins Algorithm [4]

1. Low-pass Filter
   - 11 Hz cut-off frequency
   - 5-sample delay
   - Gain of 36
   \[ y(n) = 2y(n - 1) - y(n - 2) + x(n) - 2x(n - 6) + x(n - 12) \]

2. High-pass Filter
   - 5 Hz cut-off frequency
   - 29-sample delay
   - Gain of 1
   \[ y(n) = y(n - 1) - \frac{1}{32} x(n) + x(n - 16) - x(n - 17) + \frac{1}{32} x(n - 32) \]
Pan-Tompkins Algorithm

3. Derivative
   - Provides information about QRS slope
   - Approximates derivative from 0-30 Hz
   - Has a 4-sample delay
     \[ y(n) = \frac{1}{8} [2x(n) + x(n - 1) - x(n - 3) - 2x(n - 4)] \]

4. Squaring Function
   - Emphasizes higher frequencies of the ECG (caused by QRS complexes)
     \[ y(n) = x^2(n) \]
Pan-Tompkins Algorithm

5. Moving-Window Integration
   • Detects long-duration and large-amplitude QRS complexes
     \[ y(nT) = \frac{1}{N} \left[ x(nT - (N - 1)T) + x(nT - (N - 2)T + \cdots + x(nT) \right] \]
Algorithm Efficacy, 100s

Figure 23. Performance of template-matching algorithm MATLAB simulation
Figure 49. Performance of template-matching algorithm MATLAB simulation
WFDB Library (PC side)

- `isigopen()`: open a specific WFDB record
- `getvec()`: get the next sample in the record
WFDB Library (PC side)

- `isigopen()`: open a specific WFDB record
- `getvec()` : get the next sample in the record
WFDB Toolbox (MATLAB)

• `rdsamp()`: place samples from a WFDB record into a vector
• `rdann()`: place annotations (characters) from a WFDB record into a vector
WFDB Toolbox (MATLAB)

- `wrsann()`: write experimental annotations into a vector
- `bxb()`: generate a report (with accuracy and positive predictivity data) using experimental annotations
Sample BXB Report

Beats-by-beat comparison results for record mitdb/100

Reference annotator: atr
Test annotator: test

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>n</th>
<th>s</th>
<th>v</th>
<th>f</th>
<th>q</th>
<th>o</th>
<th>x</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>356</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>S</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>V</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>F</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Q</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>O</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>X</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

QRS sensitivity: 97.83% (360/368)
QRS positive predictivity: 100.00% (360/360)
VEB sensitivity: - (0/0)
VEB positive predictivity: - (0/0)
SVEB sensitivity: 0.00% (0/4)
SVEB positive predictivity: - (0/0)
RMS RR interval error: 168.87 ms

Figure. Text file generated using the `bxb()` function in the WFDB toolbox
DSP library for CC3200

• Corr_Coeff.c
  • arm_sub_f32()
  • arm_mult_f32()

• Normal_amplt.c, normIntRange.c
  • arm_mean_f32()
  • arm_std_f32()
Sorting the Amplitude Series

• Let $S_{amp} = [2 \ 1 \ 10 \ 5 \ 9 \ 1.5 \ 3 \ 0 \ 1.7 \ 5 \ 7]$

• After sorting,

  sort_series = [0 \ 1 \ 1.5 \ 1.7 \ 2 \ 3 \ 5 \ 5 \ 7 \ 9 \ 10]
QRS Detection

Figure 50. MATLAB plot of QRS detection results
UART Data Transfer

- Obtain the next sample from the MIT-BIH database
- Convert the sample into a sequence of characters
- Add the characters to the output buffer
- 198 samples?
  - No (N)
  - Yes (Y)
- Pad the output buffer to a length of 990 characters
- Send out the buffer using UART, reset number of samples

Figure 30. UART data transfer flowchart (PC)
UART Data Transfer

UDMA Controller

Data available?

N

Y

Place buffer contents in memory

Trigger UART interrupt

UART Interrupt

Set bRxDone

Clear UART interrupt

Main Function

Is bRxDone set?

N

Y

Convert character buffer to integer buffer

Use the Pan-Tompkins algorithm to find peaks

Clear bRxDone

Figure 31. UART data transfer flowcharts (CC3200)
Alternative QRS Detection

Figure 24. Flowchart for alternative QRS detection method
Alternative QRS Detection

- Uses statistics to accurately locate QRS onset and offset
- Can be used to determine abnormal QRS complexes

Figure 24. Flowchart for alternative QRS detection method
Alternative QRS Detection

- Use the Hilbert transform to obtain the envelope

Figure 25. Flowchart for alternative QRS detection method
Alternative QRS Detection

- Estimate the derivative using a parabola:

\[ h'(k) = \frac{1}{10} \left(2(h(k + 2r) - h(k - 2r)) + h(k + r) - h(k - r)\right) \]

Figure 26. Flowchart for alternative QRS detection method
Alternative QRS Detection

• Calculate a cumulative mean for the QRS onset and offset windows
• Determine the probability density functions

Figure 27. Flowchart for alternative QRS detection method
Envelop Signal

Figure 41. Envelope of the initial ECG signal
Auxiliary Signal

Figure 42. Auxiliary signal of the envelope