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

- 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)

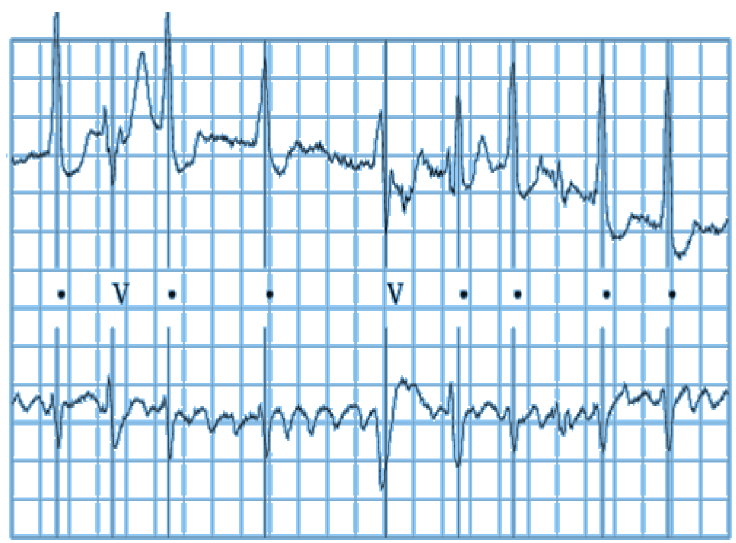


Figure 1. Electrocardiogram with “V” labels for PVCs [2]

Problem Description

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

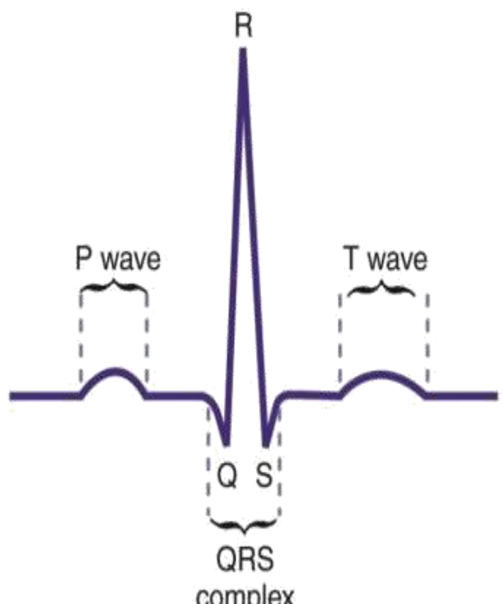


Figure 2. Features of a normal ECG [3]

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

System Block Diagram

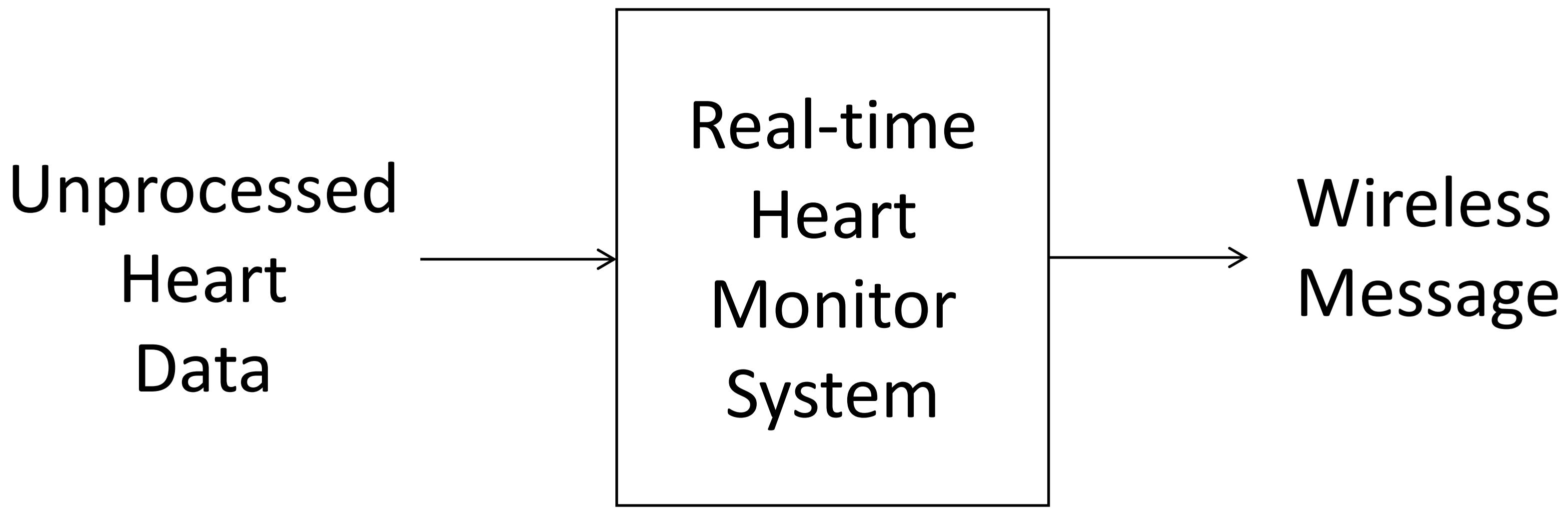


Figure 3. Overall heart monitoring system diagram

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

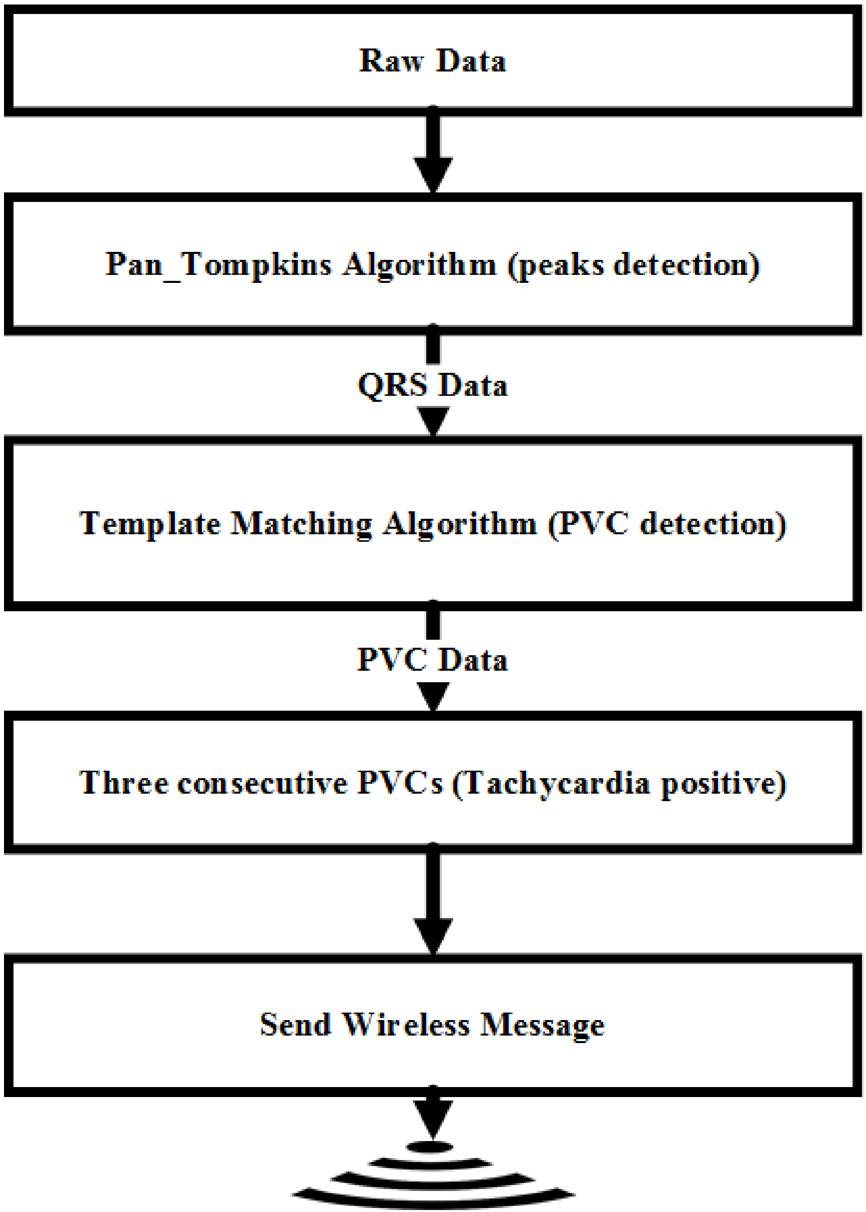


Figure 4. High-level flowchart for algorithms

Template Generation Algorithm [4]

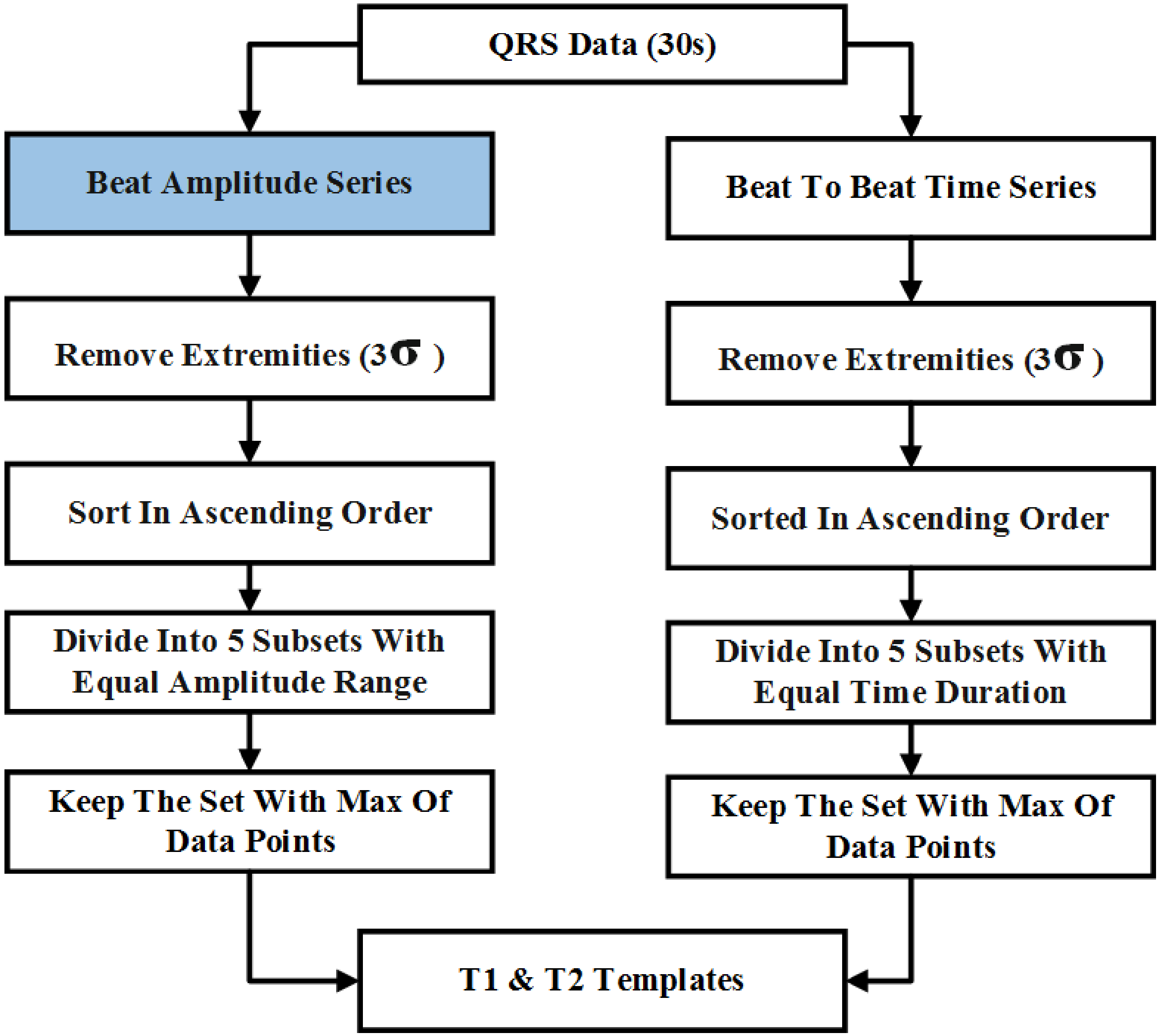


Figure 5. Flowchart for T1 and T2 generation

Beat Amplitude Series

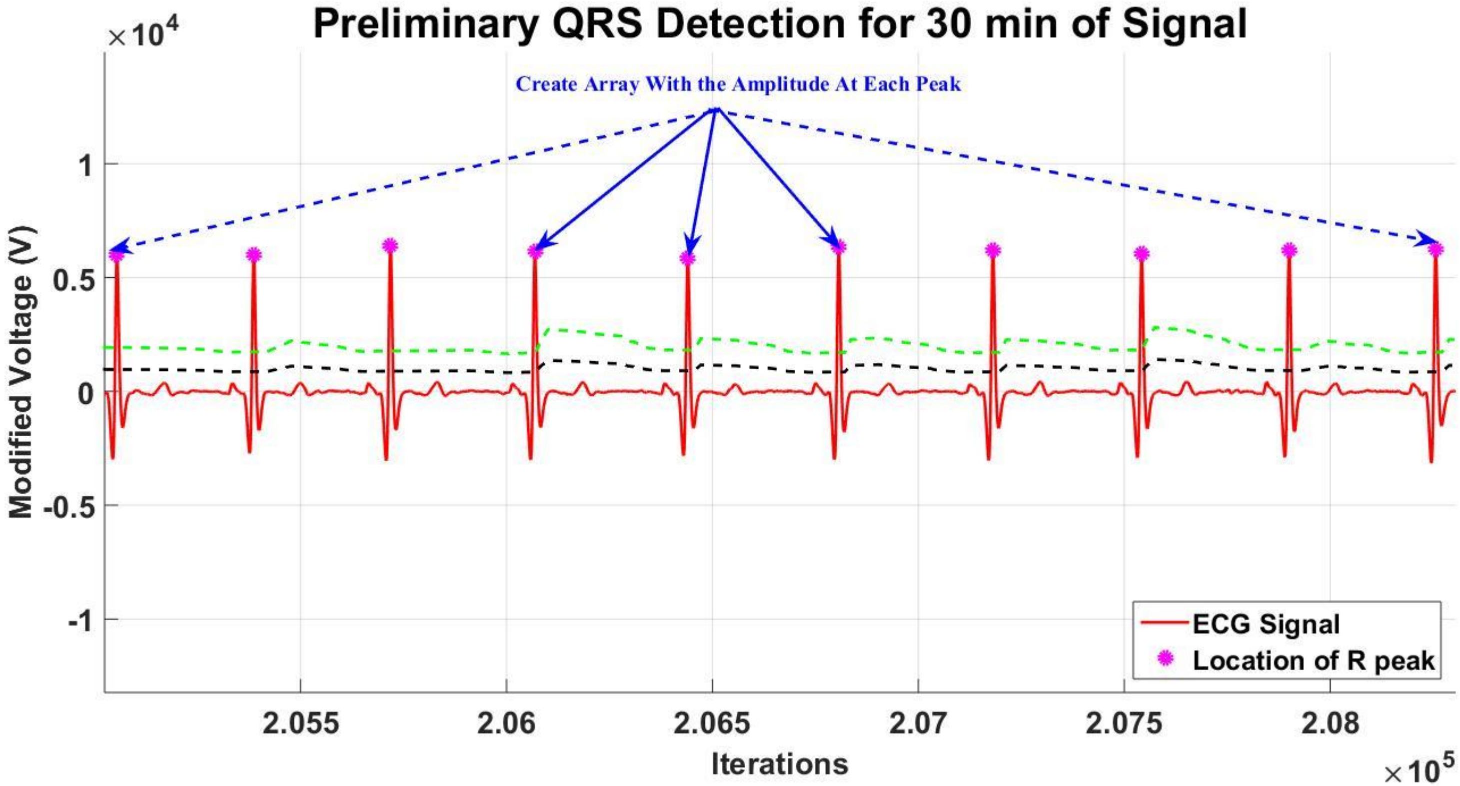


Figure 6. MATLAB plot of QRS detection results. The marked peaks were placed into an array.

Template Generation Algorithm

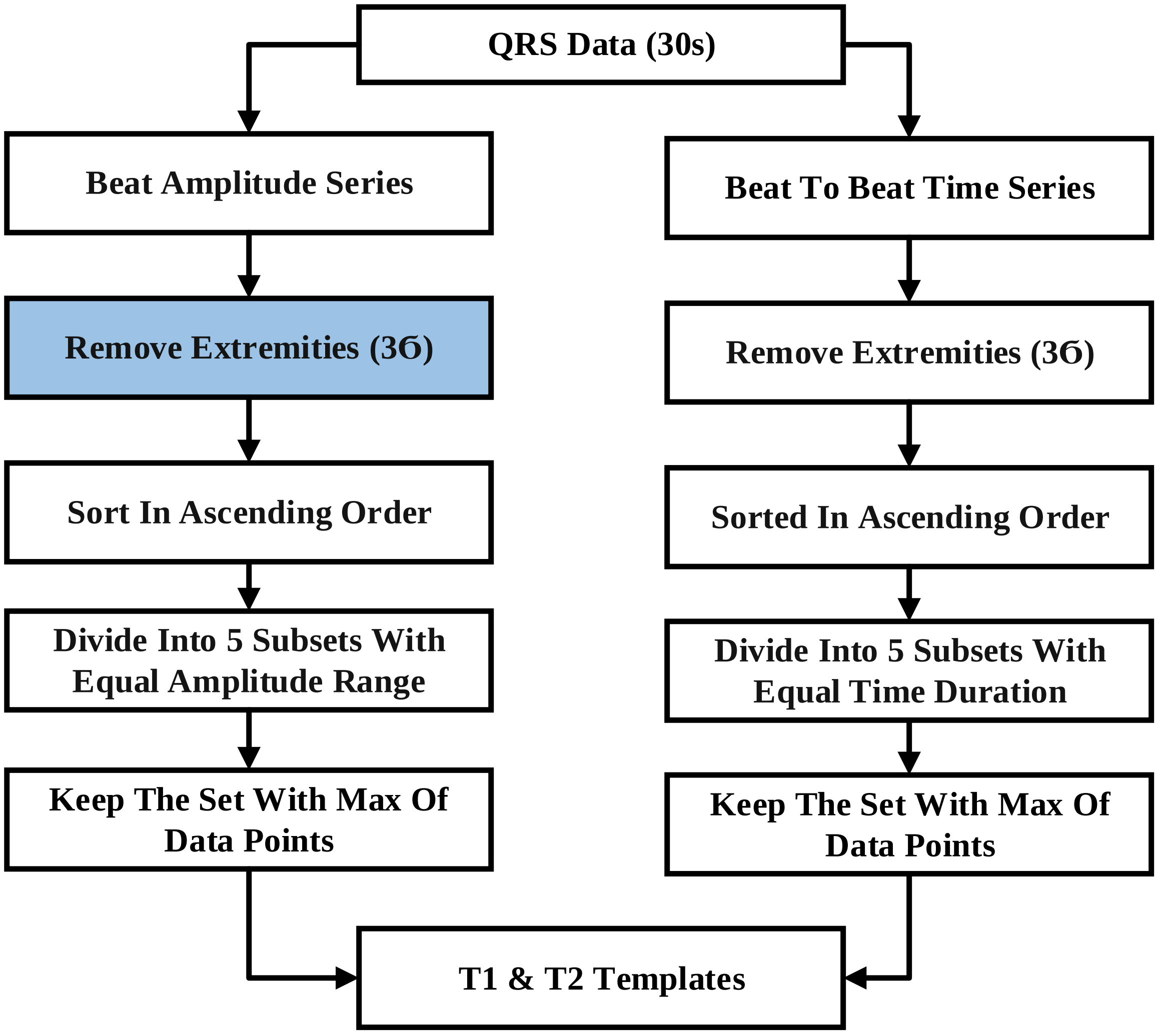


Figure 7. Flowchart for T1 and T2 generation

Template Generation Algorithm

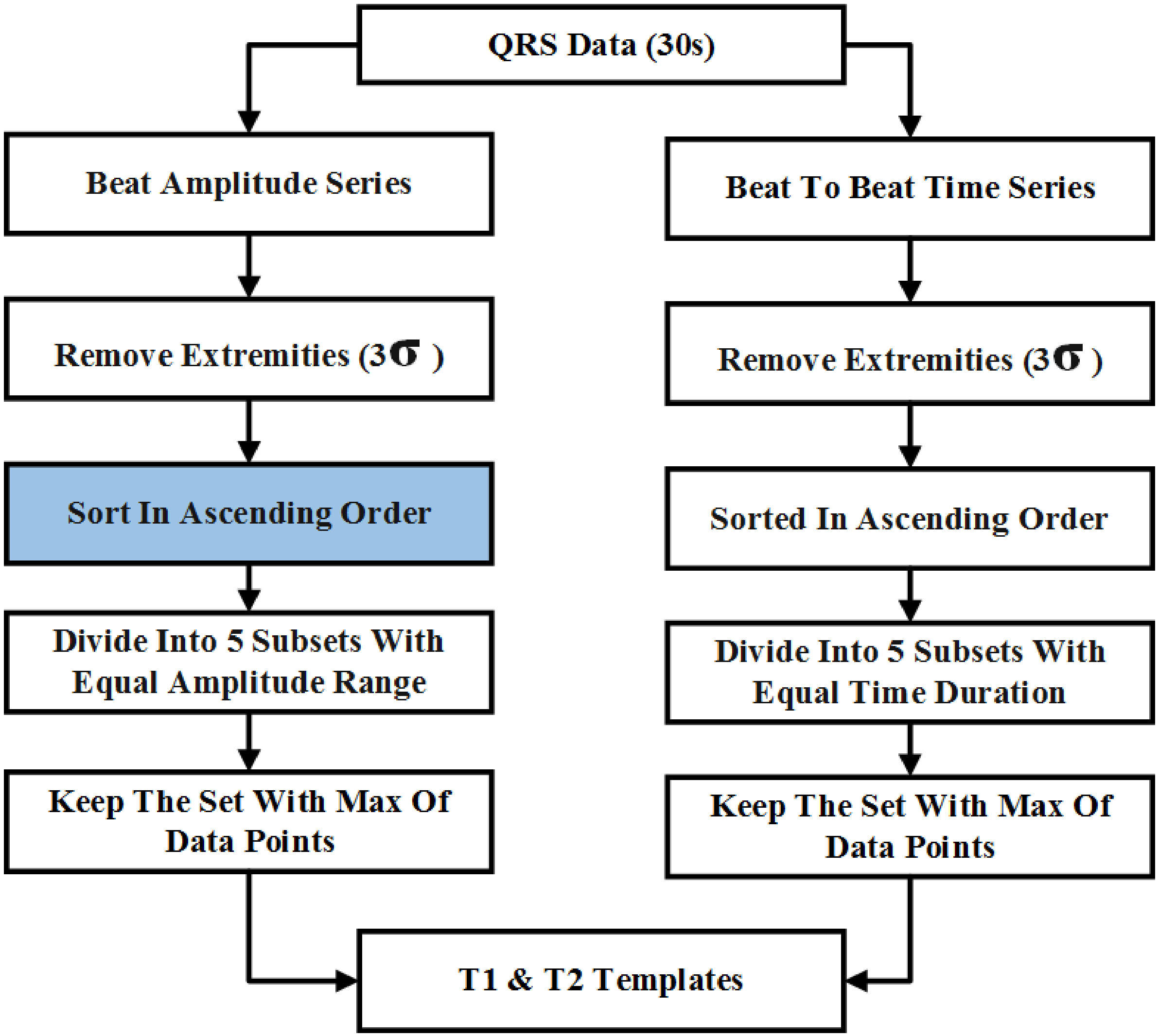


Figure 8. Flowchart for T1 and T2 generation

Template Generation Algorithm

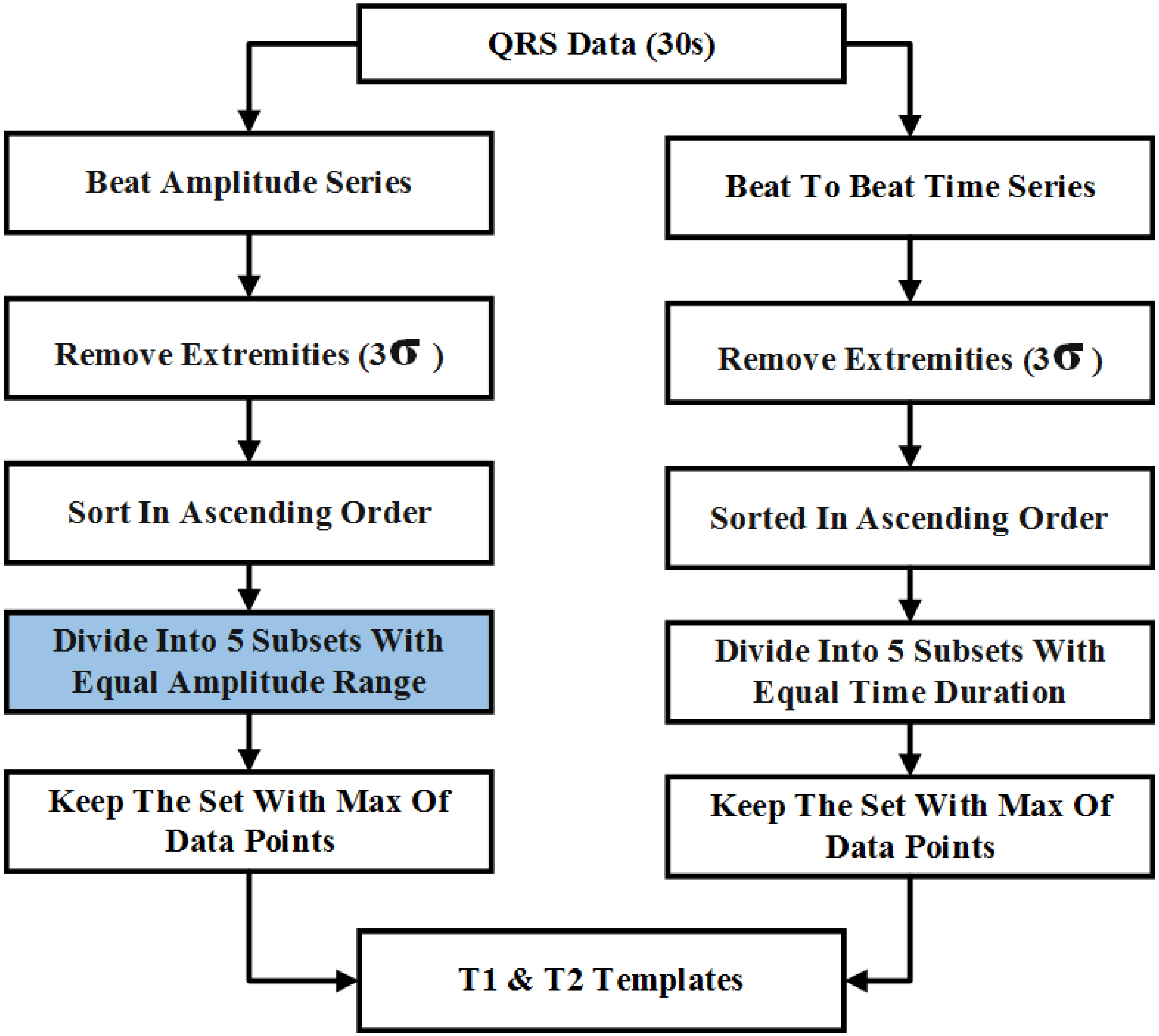


Figure 9. Flowchart for T1 and T2 generation

Template Generation Algorithm

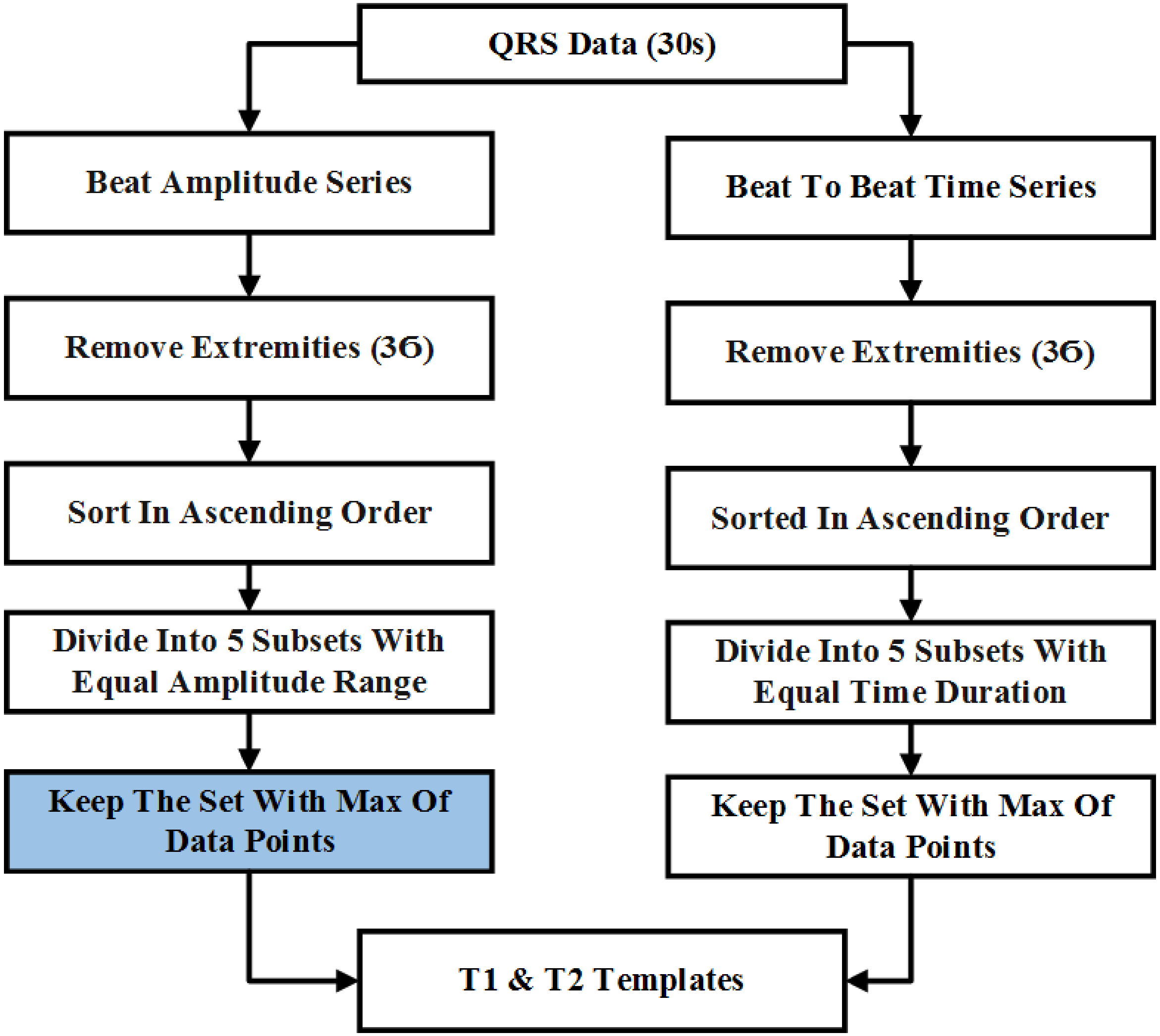


Figure 10. Flowchart for T1 and T2 generation

Template Generation Algorithm

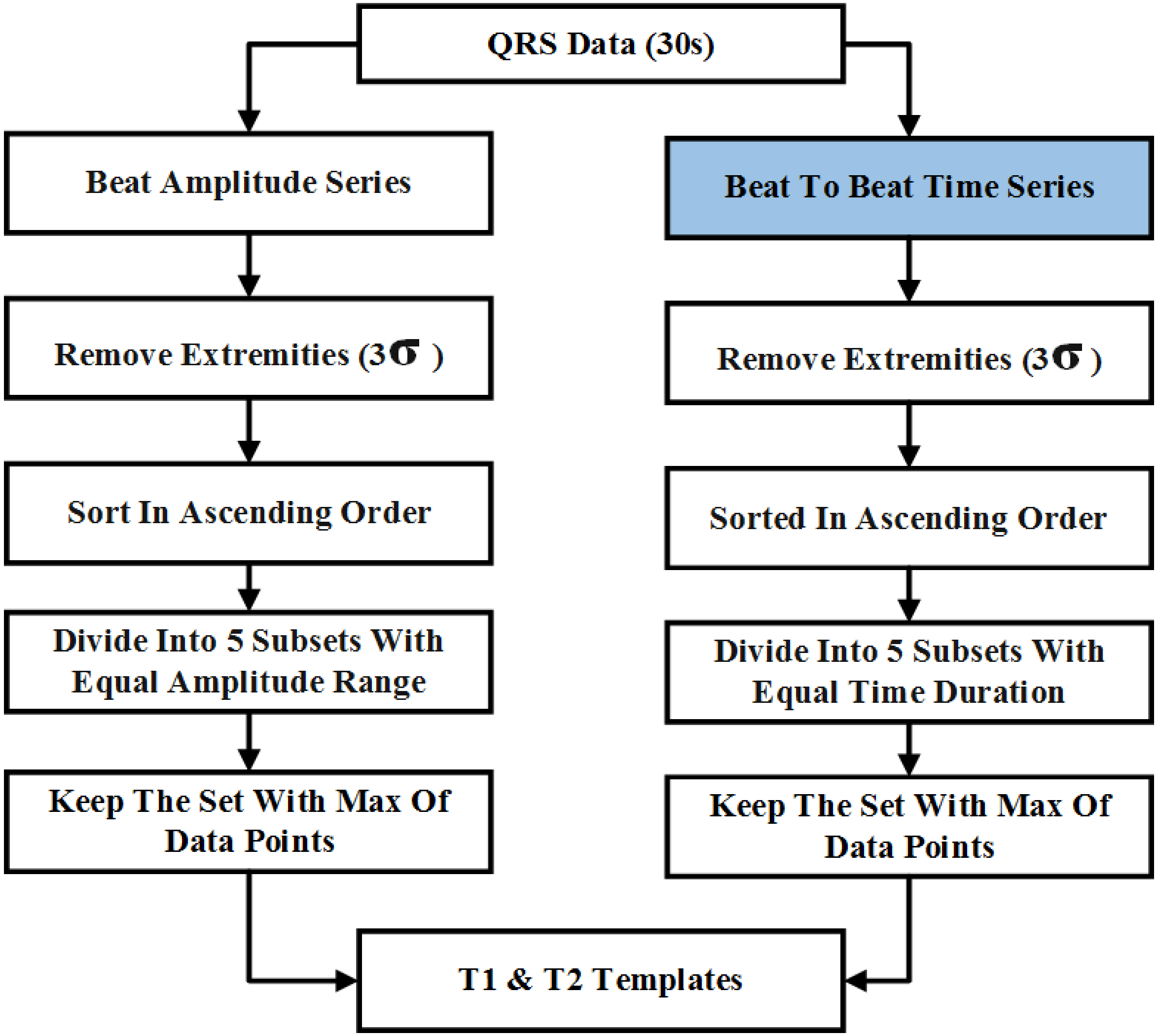


Figure 11. Flowchart for T1 and T2 generation

Data between Two Fiducial Points

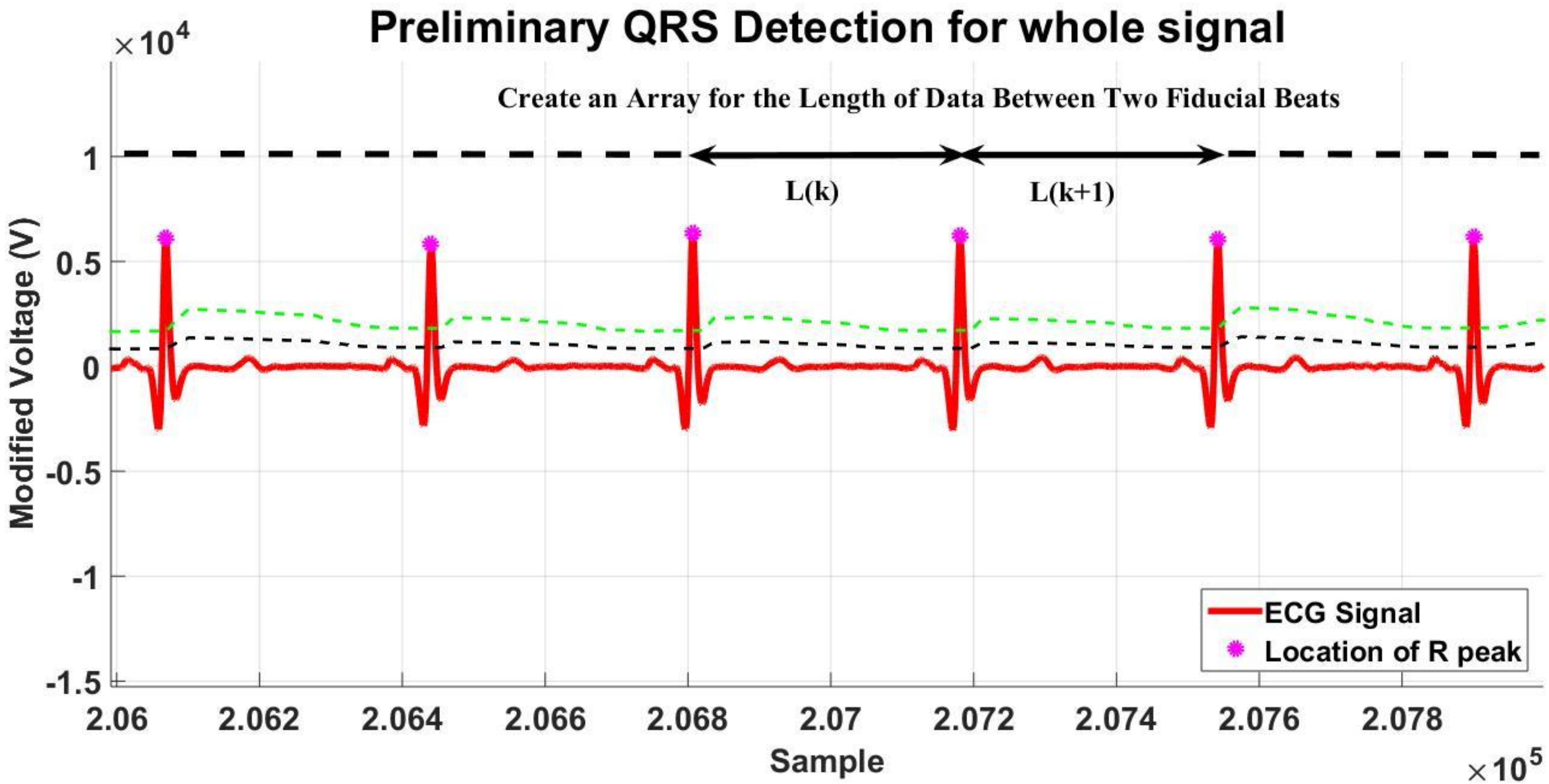


Figure 12. MATLAB plot of QRS detection results. The marked RR-intervals were placed into an array.

Template Generation Algorithm

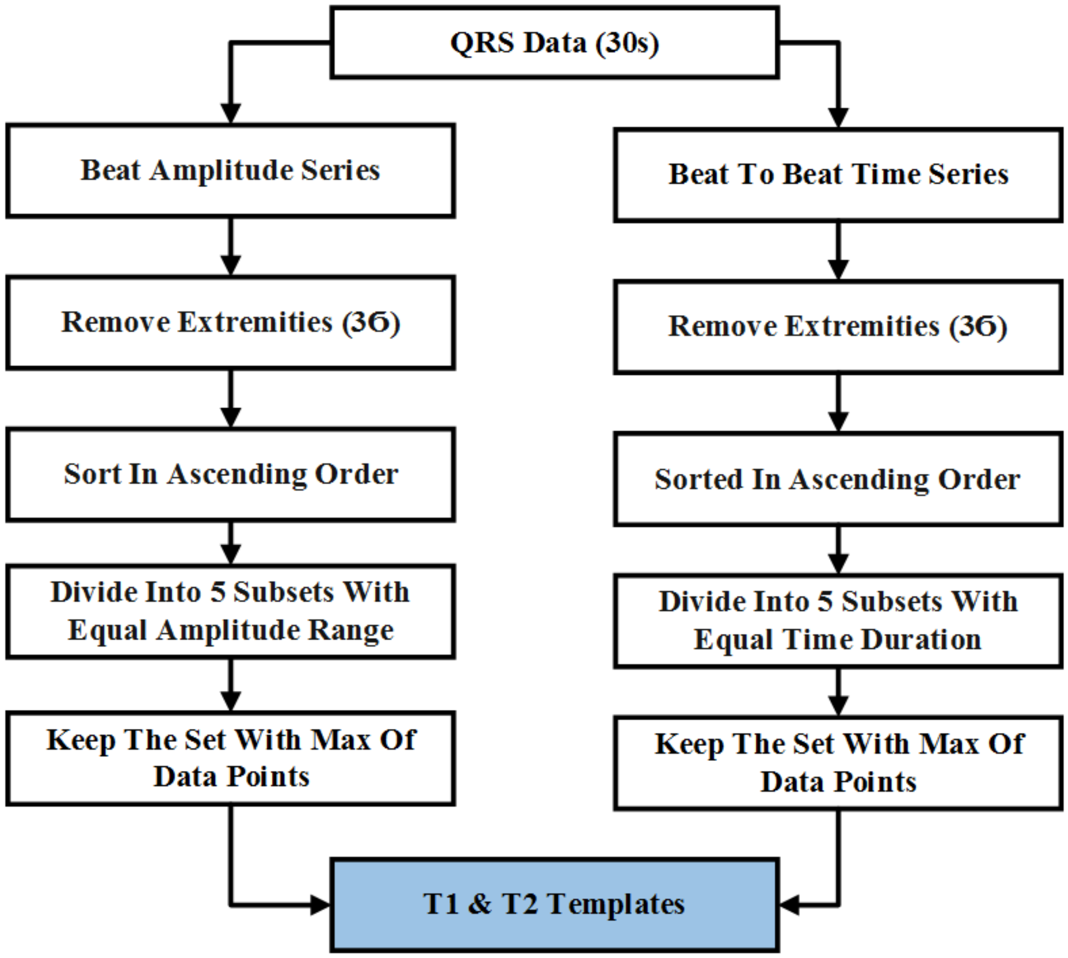


Figure 13. Flowchart for T1 and T2 generation

Templates T1 and T2

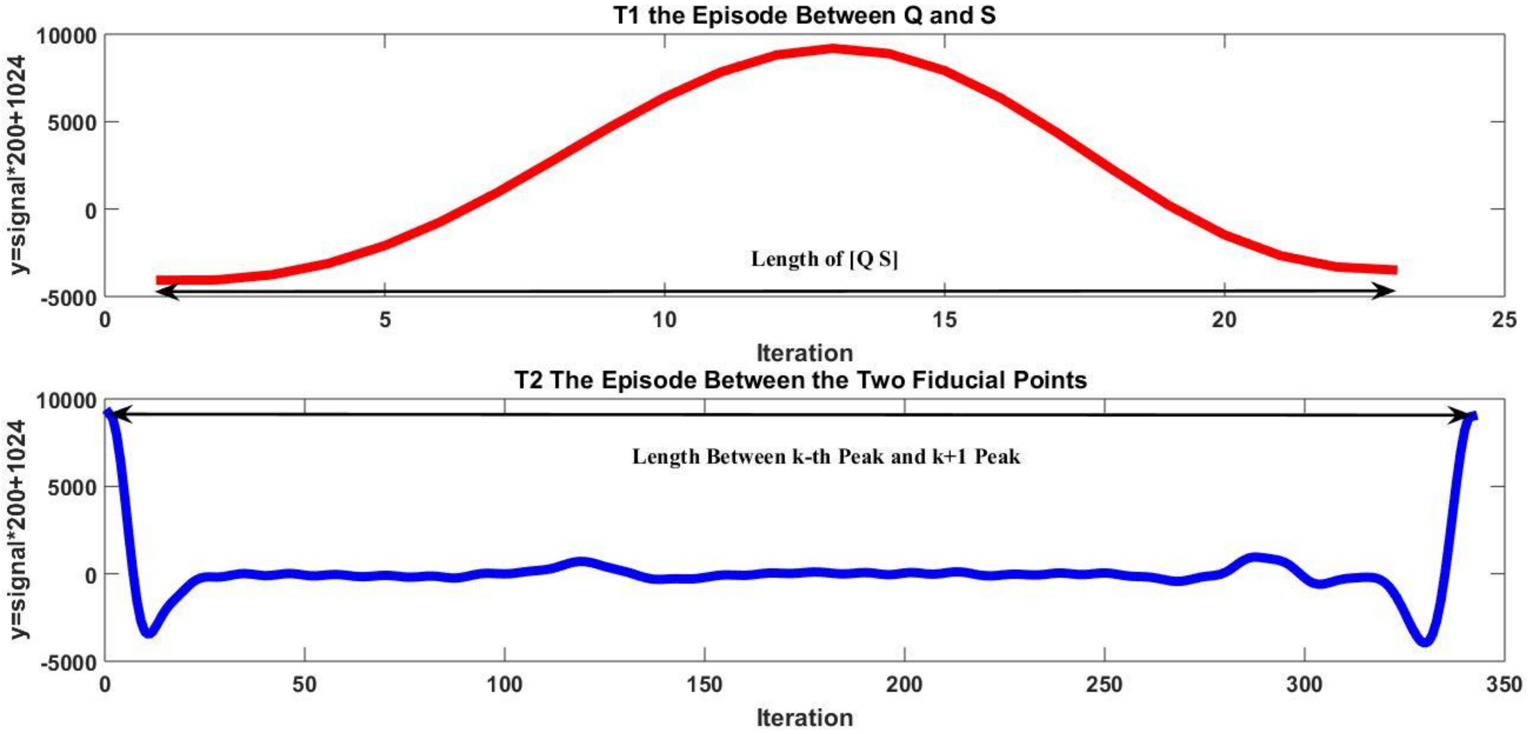


Figure 14. T1 and T2 generation

Template-Matching Algorithm

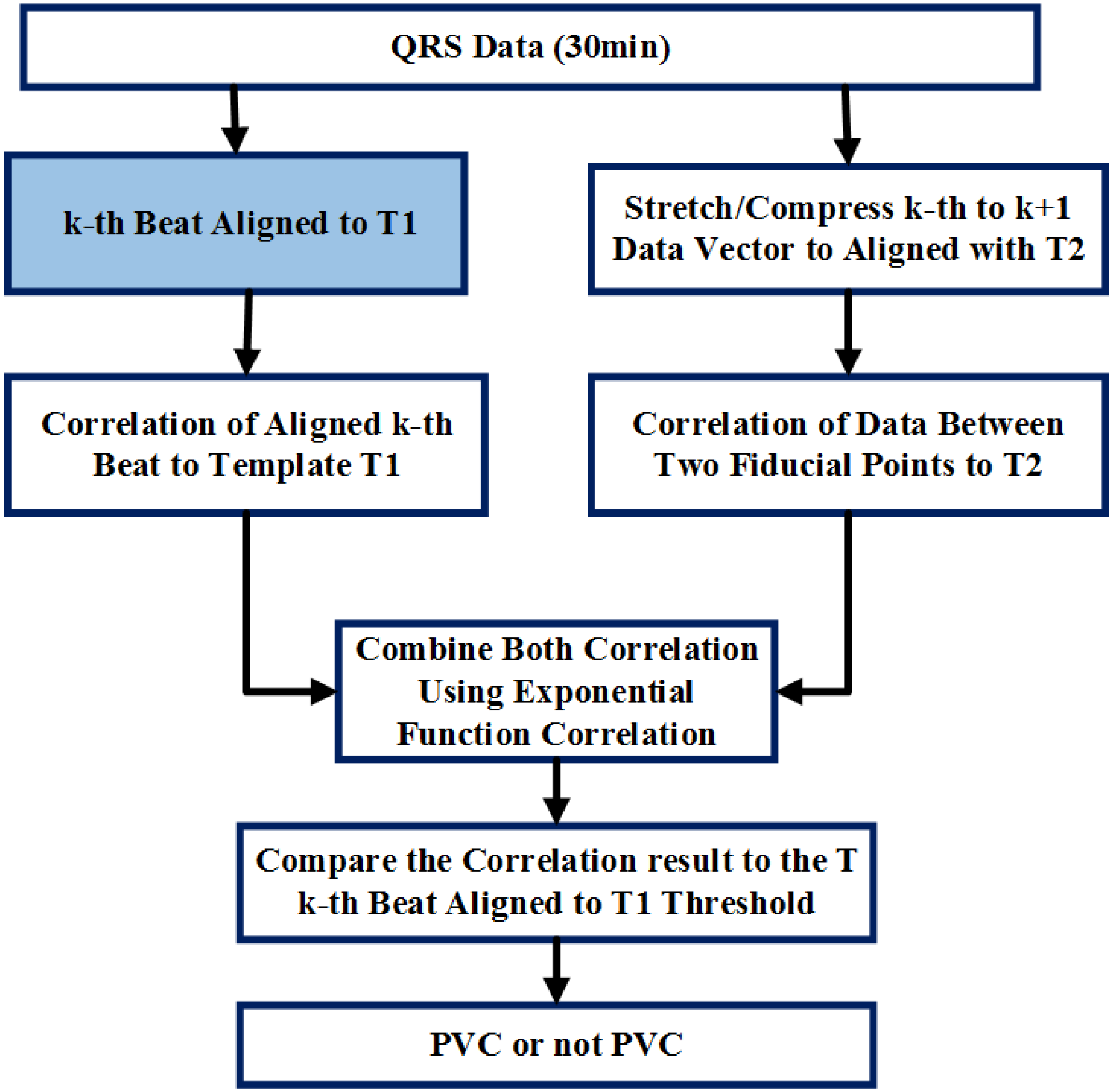


Figure 15. Flowchart for template-matching algorithm

Template-Matching Algorithm

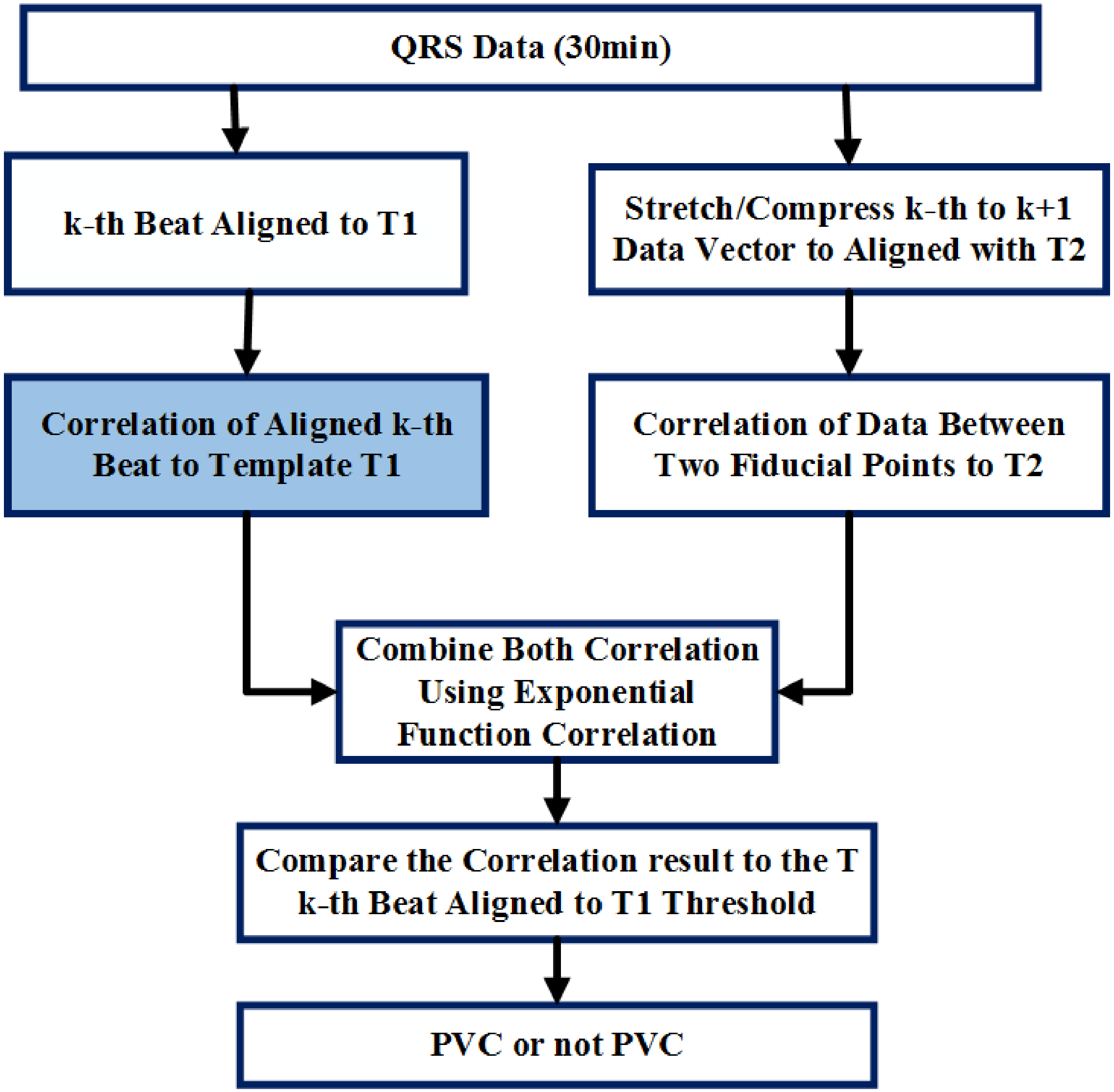


Figure 16. Flowchart for template-matching algorithm

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

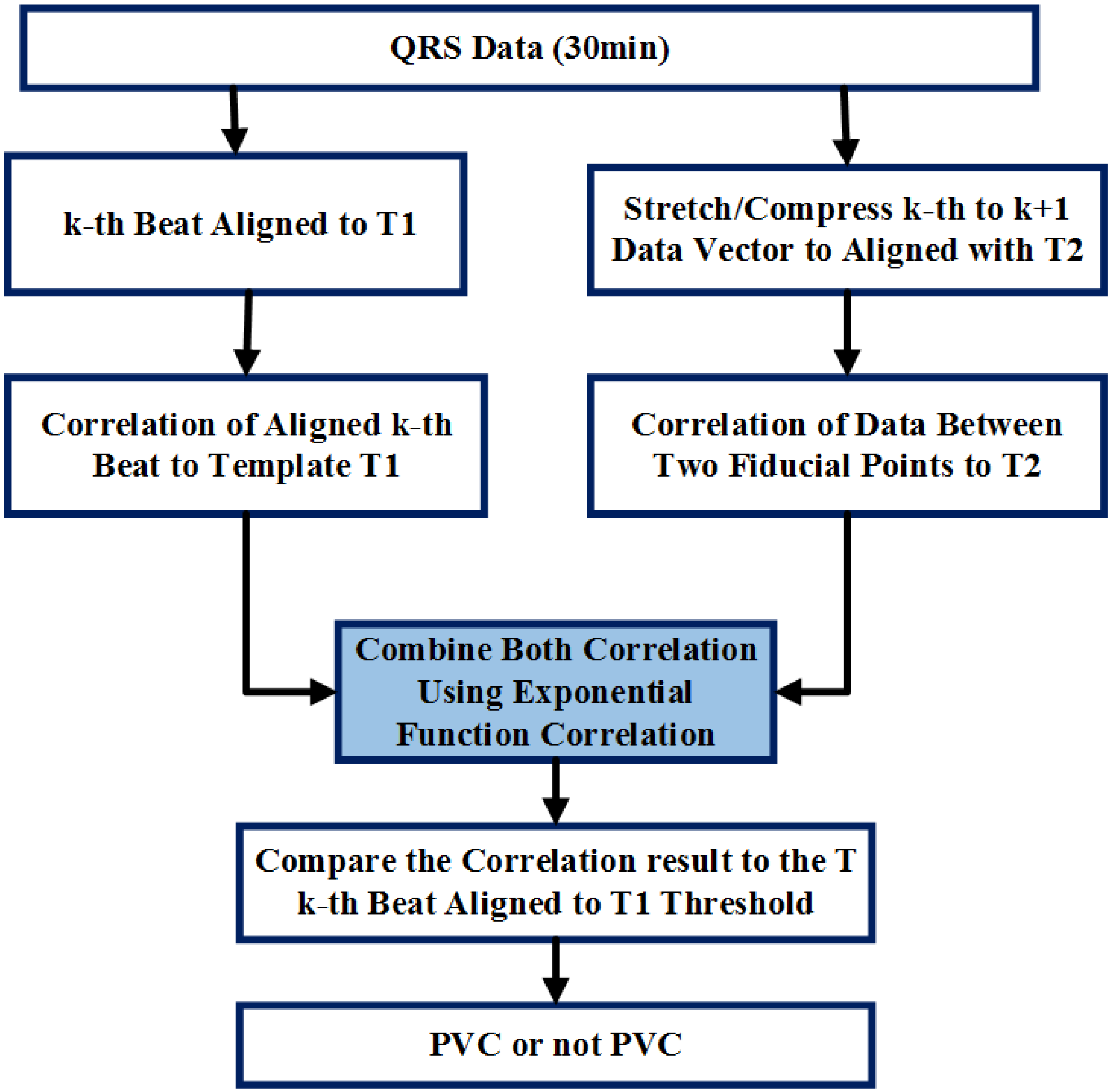


Figure 17. Flowchart for template-matching algorithm

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

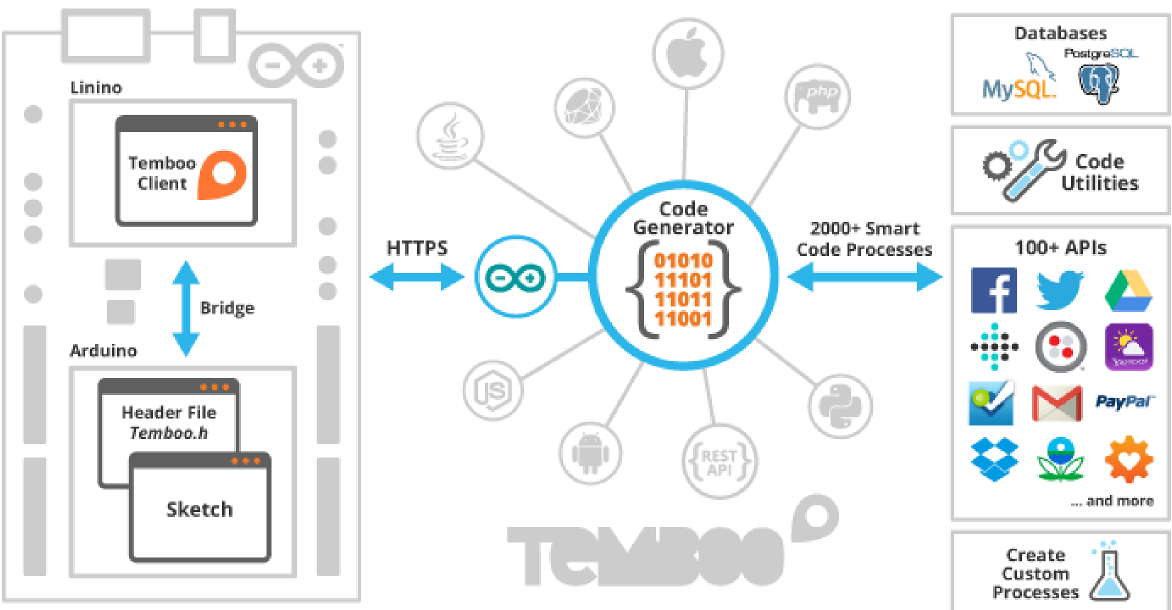


Figure 19. Temboo integration with an embedded device [6]

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:

$$0.9 * \text{average} \leq \text{template} \leq 1.1 * \text{average}$$

Template-Matching Testing

TABLE I. PERFORMANCE OF TEMPLATE-MATCHING MATLAB SIMULATION

Record	PVC Sensitivity	PVC Positive Predictivity
106	0.729	1.000
116	1.000	0.973
119	1.000	0.998
201	1.000	0.789
203	1.000	0.651
208	1.000	0.892

Past Schedule

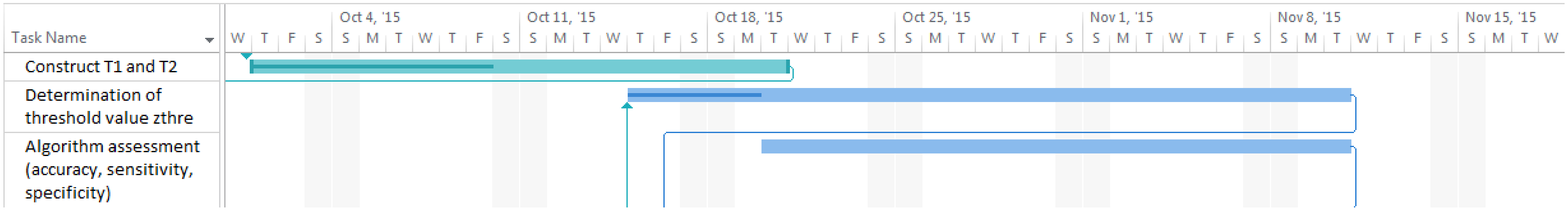
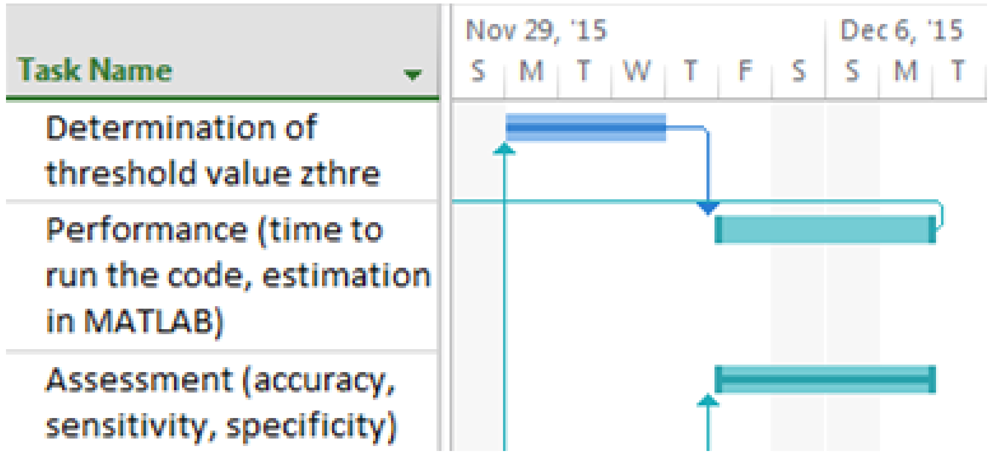


Figure 20. Gantt chart for fall semester

Current/Future Schedule



Task Name	Duration	Start	Finish
Alternative T1 construction method	16 days	Thu 1/21/16	Thu 2/11/16
Test the C code on the board using the testing data	26 days	Fri 2/12/16	Fri 3/18/16
Upload the data to a web server	4 days	Thu 1/21/16	Tue 1/26/16

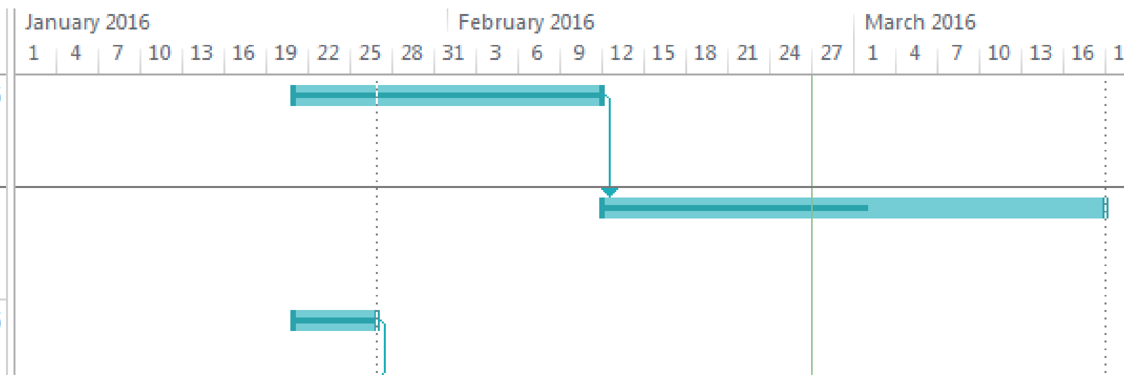


Figure 21. Gantt chart through the end of the project

Contents

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

Template-Matching Algorithm

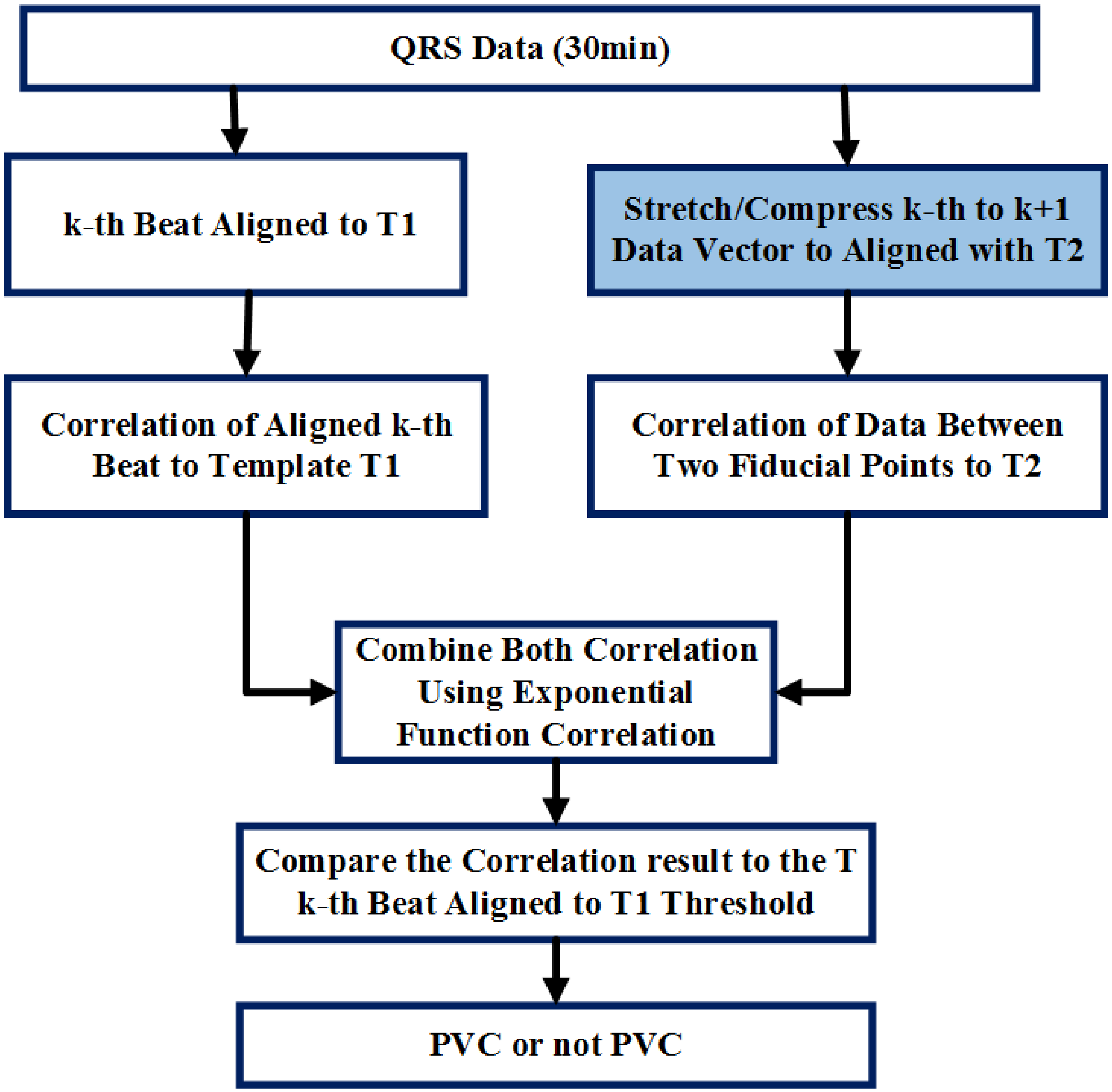


Figure 22. Flowchart for template-matching algorithm

Data between Two Fiducial Points

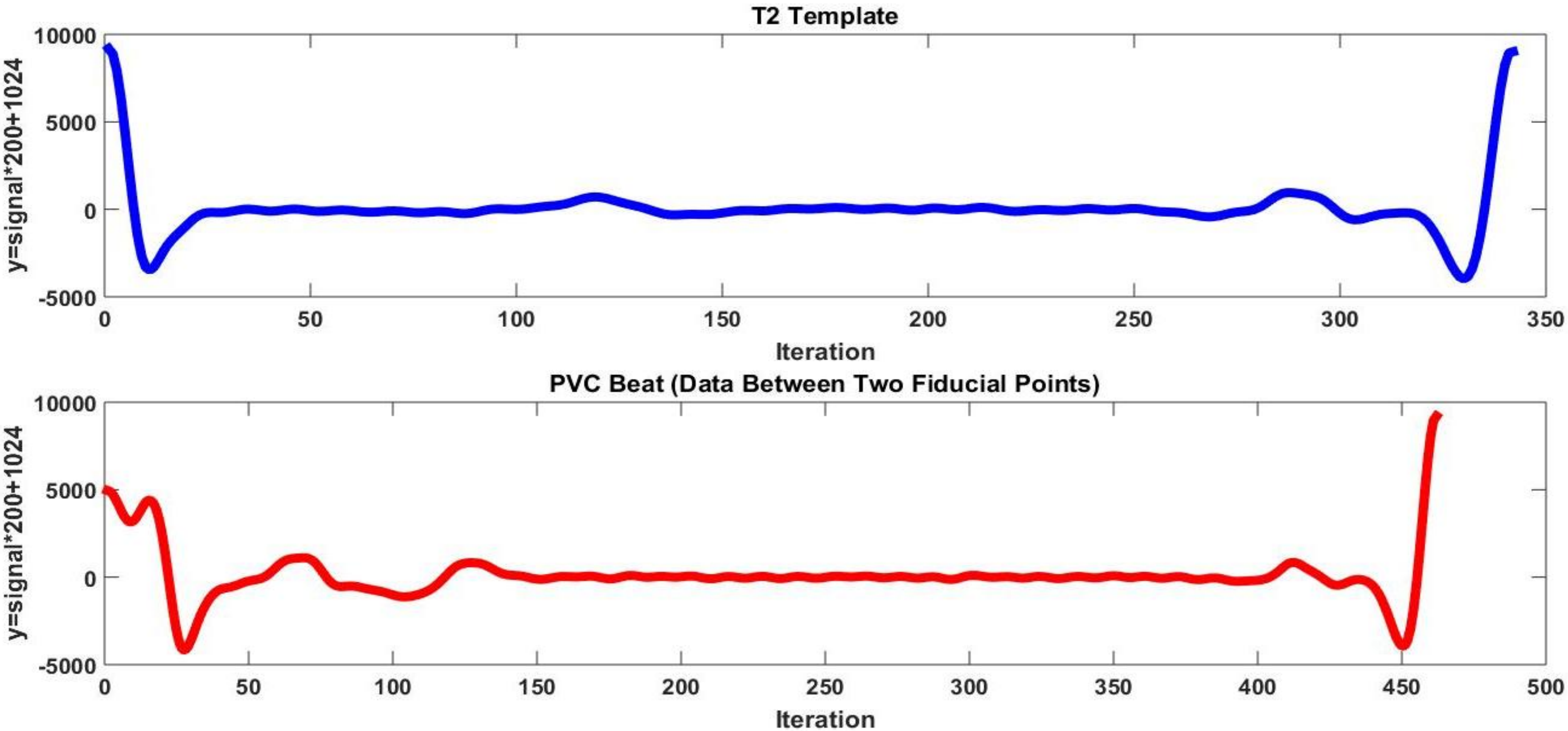


Figure 23. T2 (interval between R peaks) and RR-interval signal for PVC beat

Interpolation and Extrapolation

- Let b_k be the data between k^{th} beat and $k+1$ beat.
- If length(b_k)≤ Length($T2$), we do extrapolation based on

$$new\_b_k = b_k[1+\alpha(n-1)]\ (n = 1,2, \dots L)$$

$$\alpha = \frac{length(b_k)}{length(T_2)}$$

Example (Extrapolation)

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

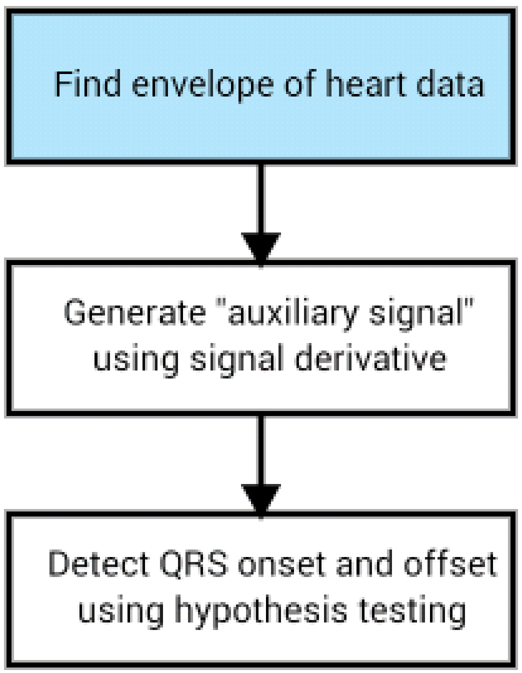
TABLE II. INDICES AFTER EXTRAPOLATION

n value	Calculated Index
1	1
2	1.48
3	1.95
4	2.43
5	2.90

Alternative QRS Detection [7]

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

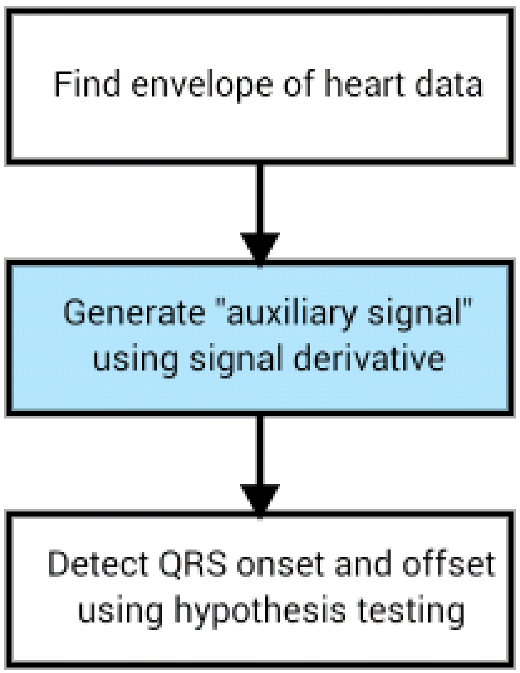
Alternative QRS Detection



$$ECG_e(k) = \sqrt{ECG^2(k) + ECG_H^2(k)}$$

where ECG_H is the Hilbert transform of the ECG signal

Alternative QRS Detection



$$AS = 2(ECG_e'(k))^2$$

where ECG_e' is the derivative of the envelope signal

Alternative QRS Detection

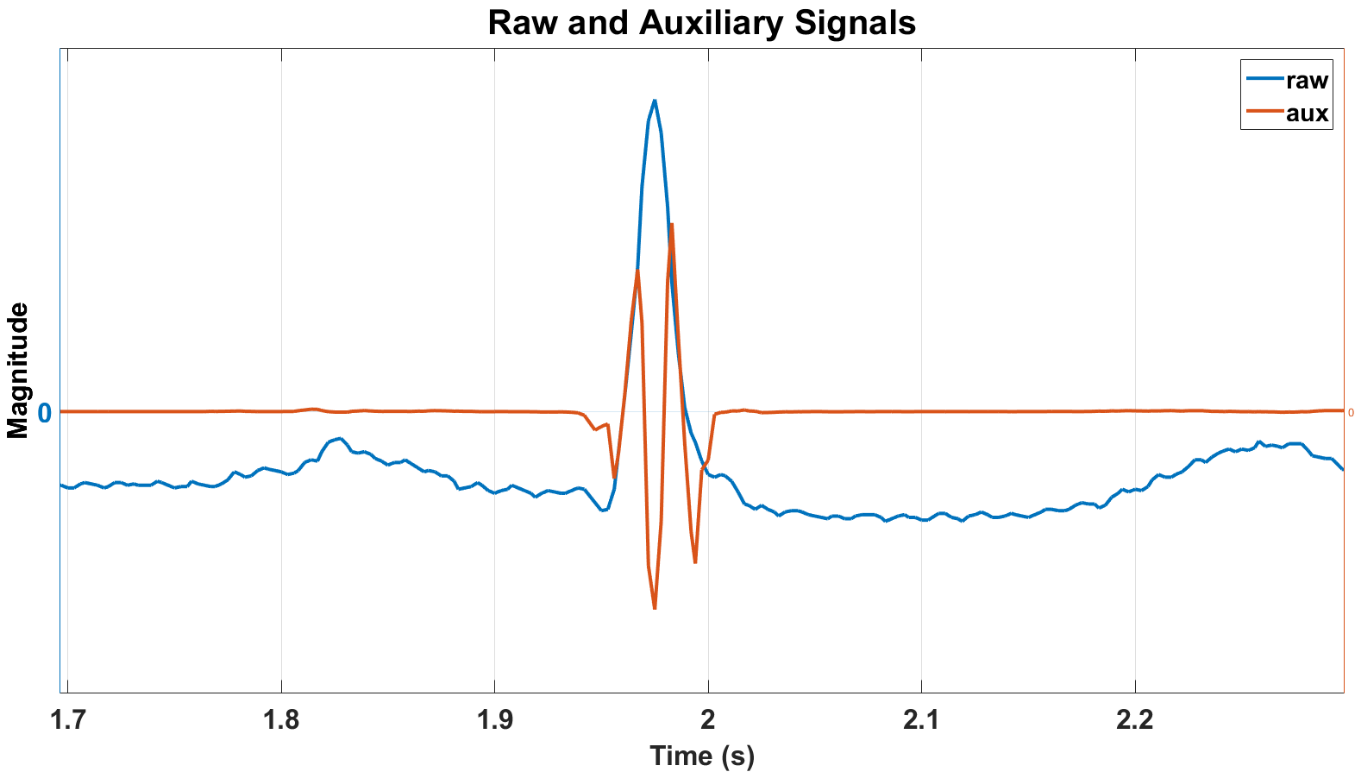
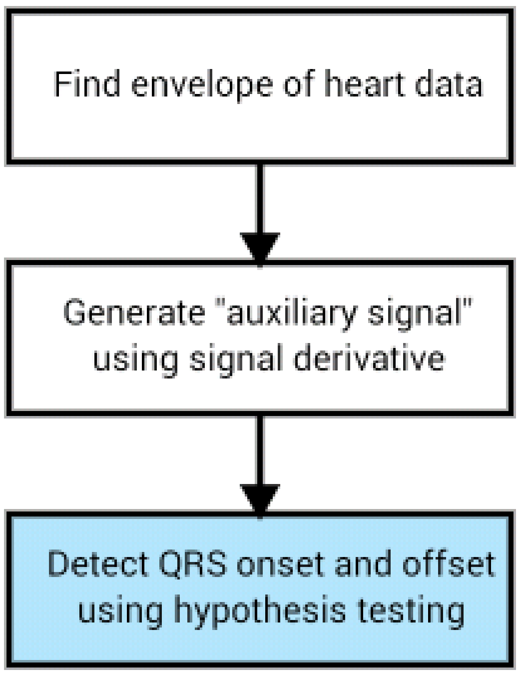


Figure 24. MATLAB plot of the raw and auxiliary signals generated using the alternative QRS detection algorithm

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

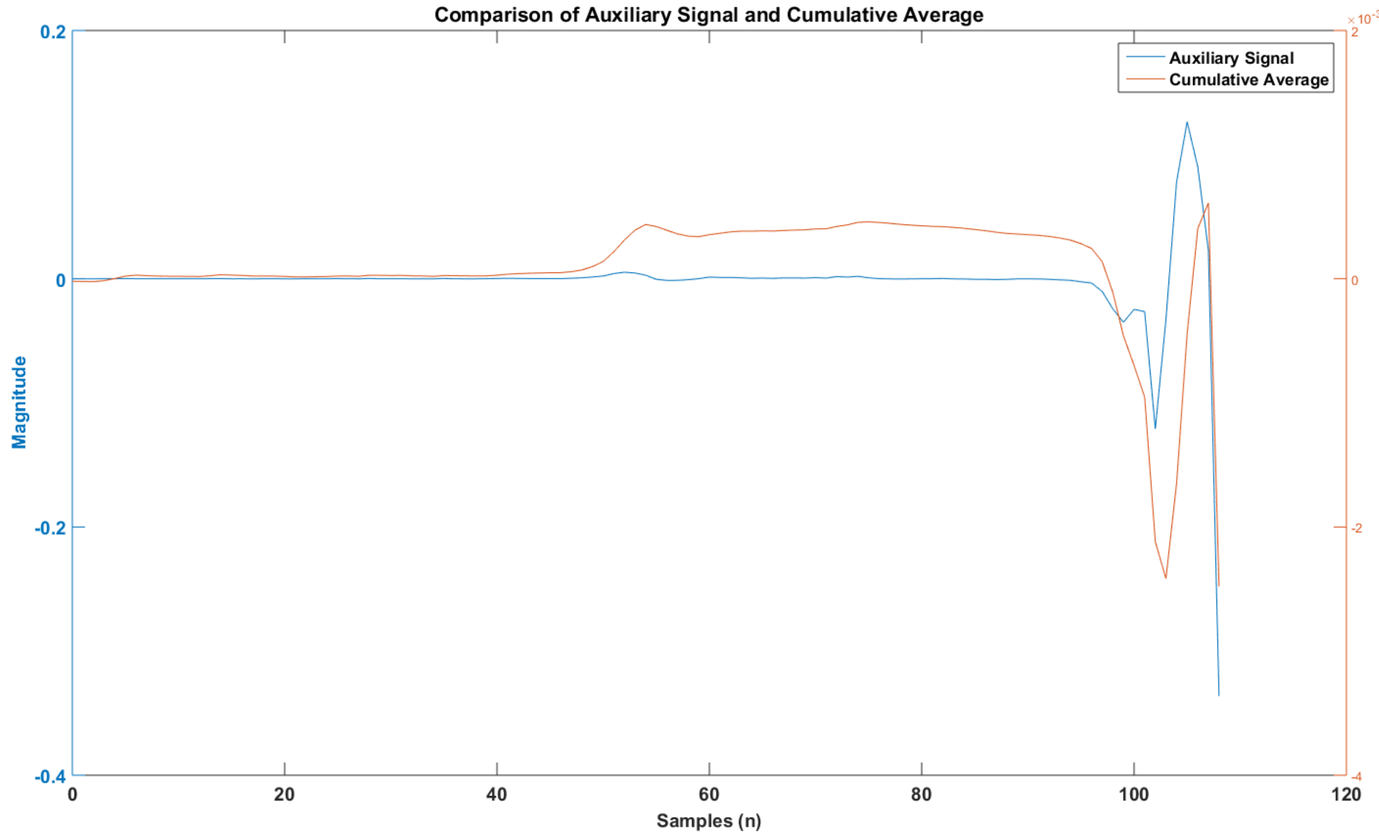


Figure 25. MATLAB plot of the auxiliary signal and the cumulative average

Wireless - Plotly

- Code language: JavaScript Object Notation (JSON)
  - Language independent
  - Based on JavaScript, C/C++/C#, Python, Perl, etc.

Wireless - Plotly

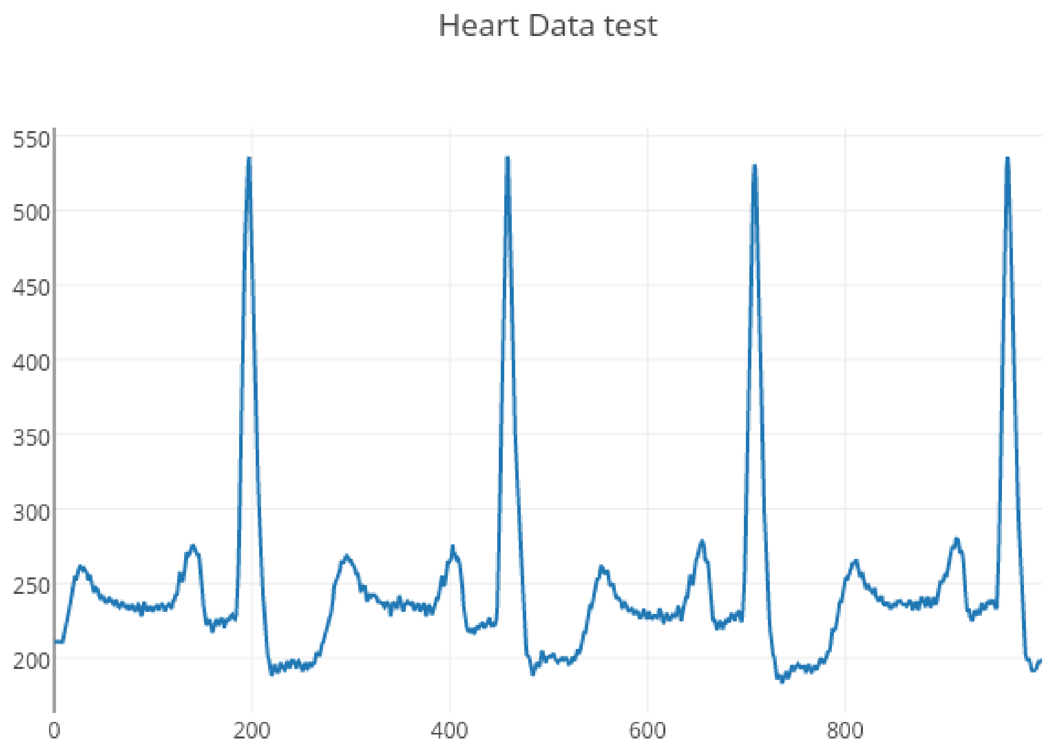


Figure 26. Sample Plotly graph of 1000 samples of heart data

Past Schedule

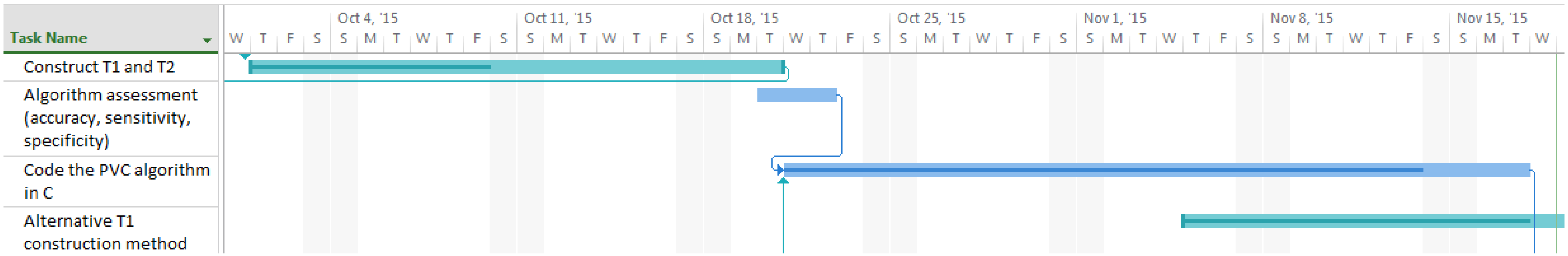


Figure 27. Gantt chart for fall semester

Current/Future Schedule

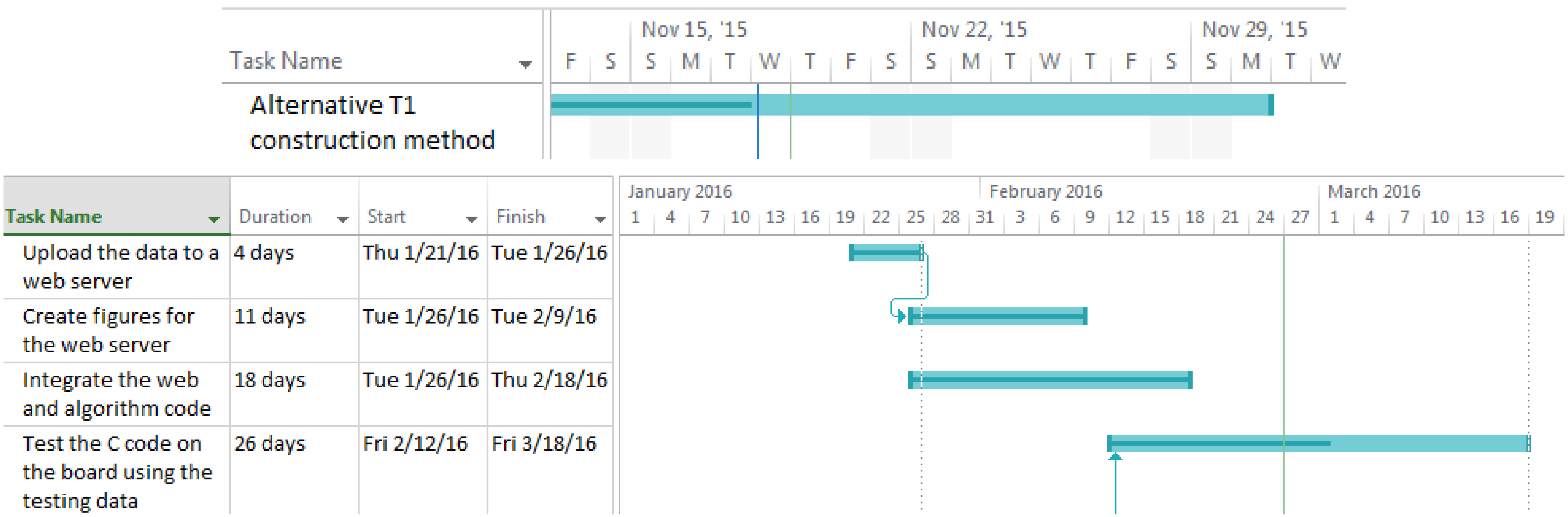


Figure 28. Gantt chart through the end of the project

Contents

Template Generation Algorithm

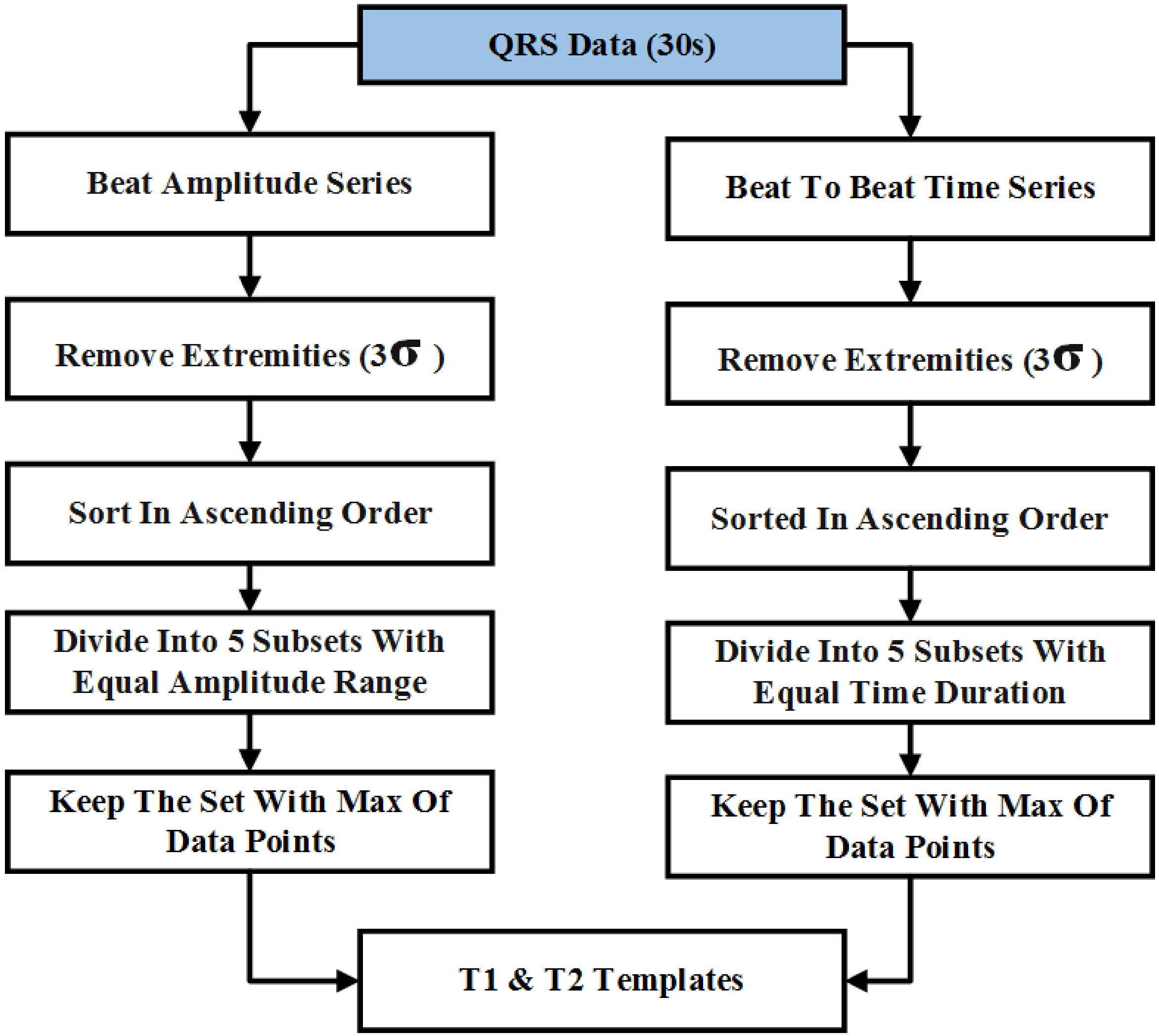


Figure 29. Flowchart for T1 and T2 generation

Testing of QRS Detection Phase

- Pan-Tompkins algorithm implemented on CC3200

Figure 30. CC3200 Launchpad [8]

UDMA Diagram

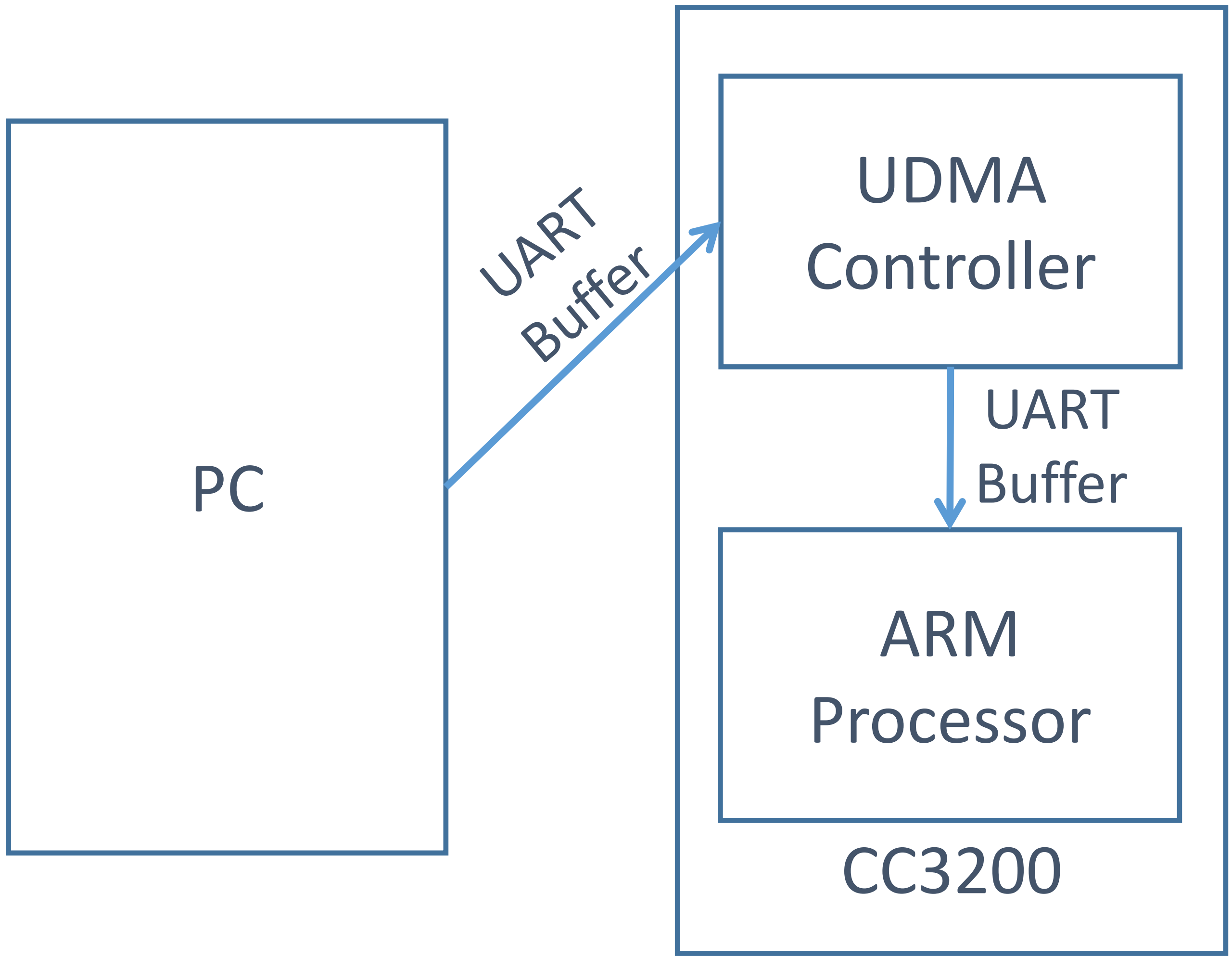


Figure 31. UART data flow between the PC and the CC3200

Pan-Tompkins Algorithm

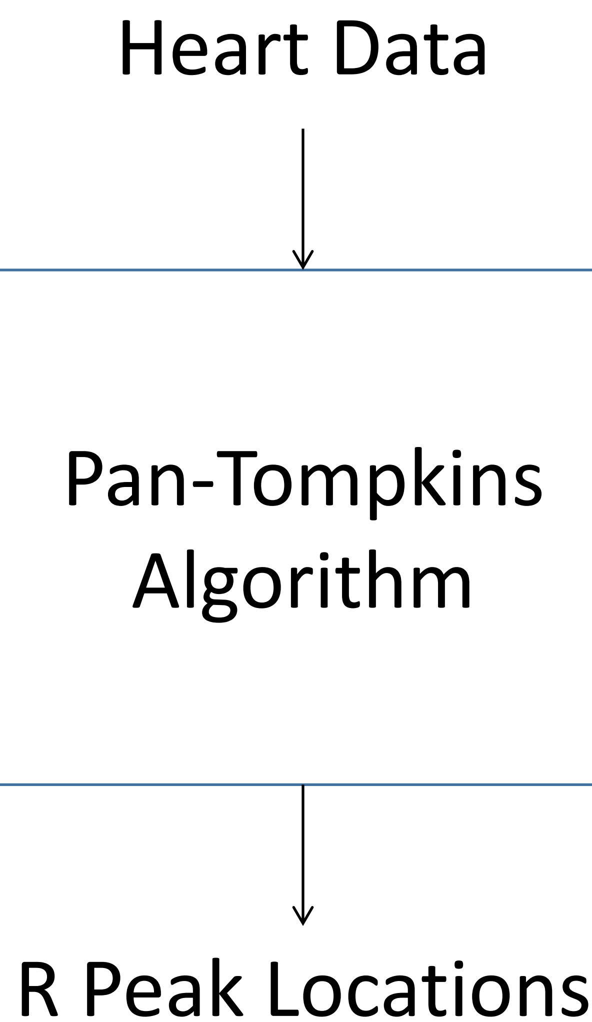


Figure 32. Pan-Tompkins block diagram

Pan-Tompkins Algorithm

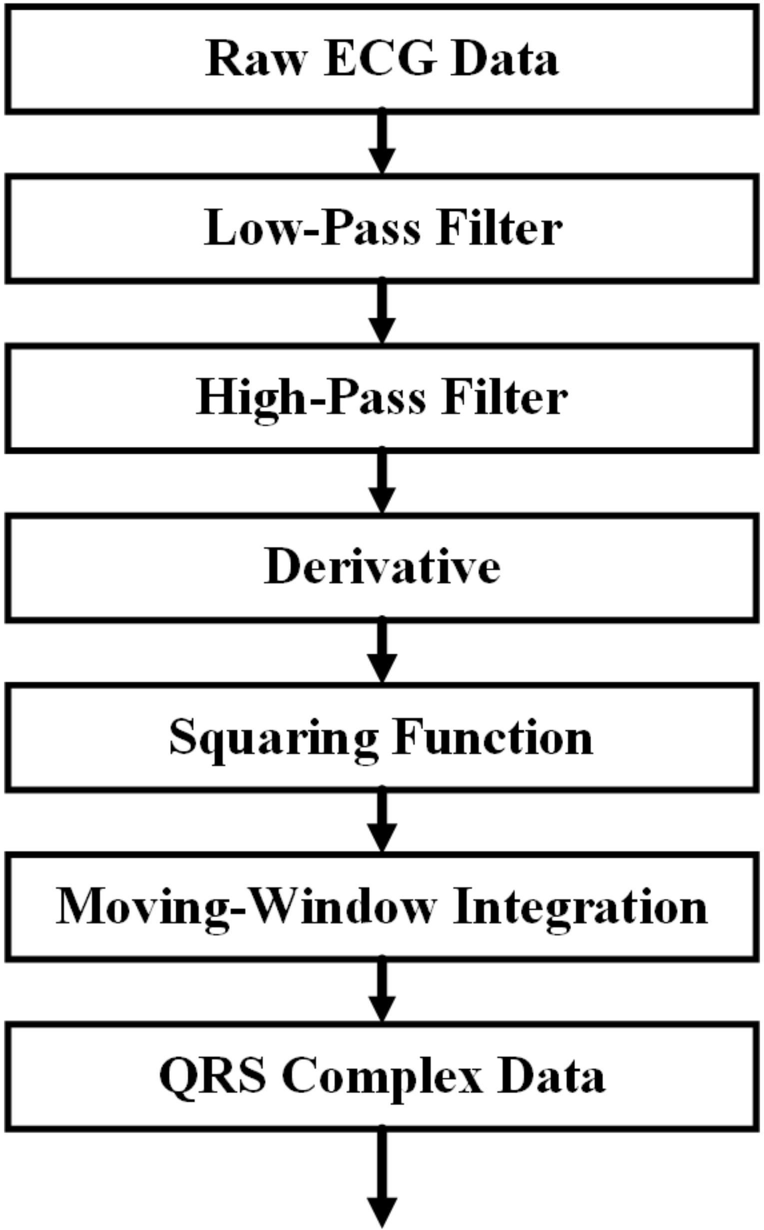


Figure 33. Flowchart for Pan-Tompkins algorithm

Results

TABLE III. PERFORMANCE OF PAN-TOMPKINS C IMPLEMENTATION

Record	QRS Sensitivity (%)	QRS Positive Predictivity (%)
100	97.83	100.00
102	96.69	100.00
103	99.43	100.00
105	99.76	99.76
106	96.67	99.69

C Implementation Results

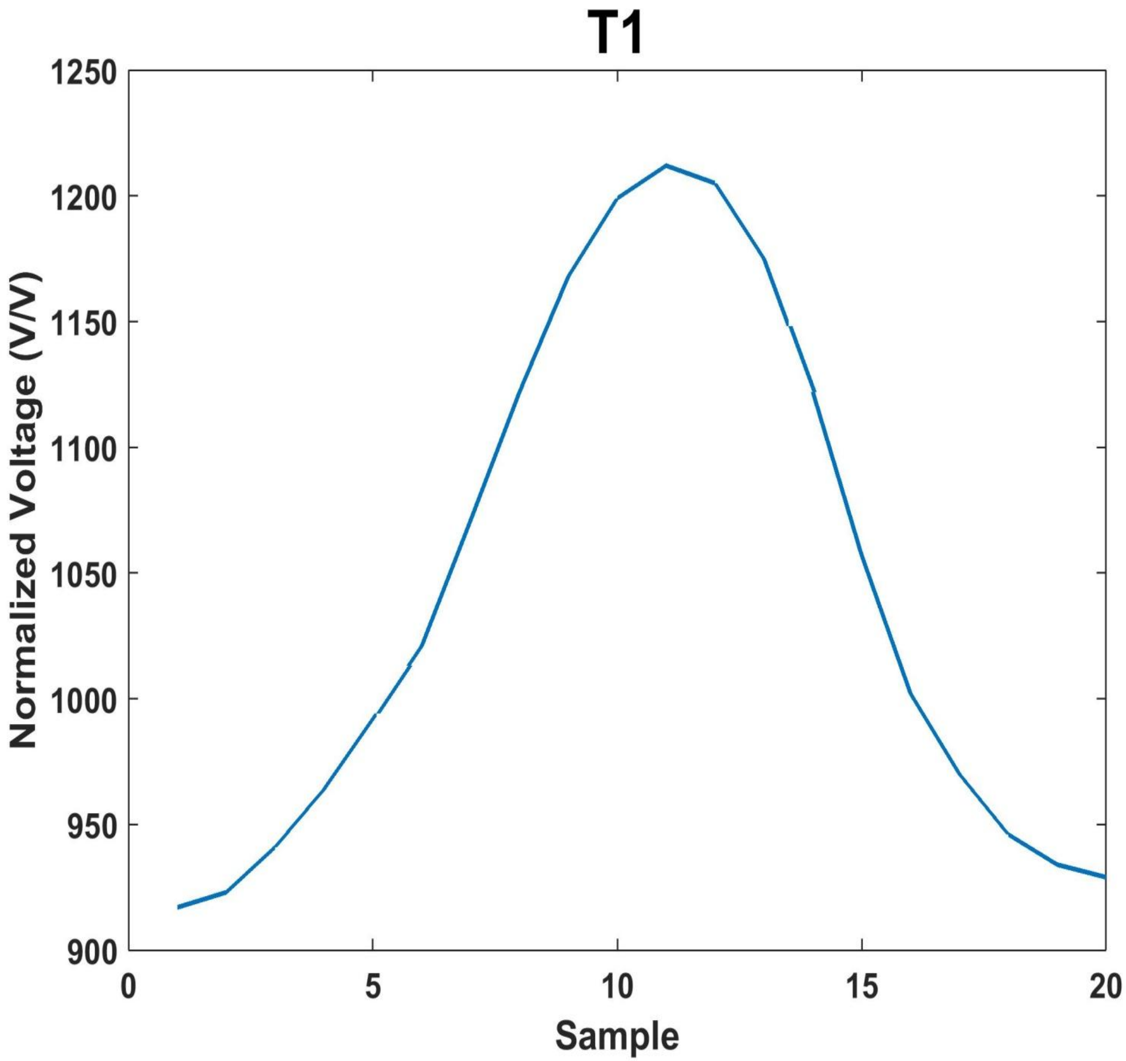


Figure 34. T1 (QRS complex) template generated on the CC3200

C Implementation Results

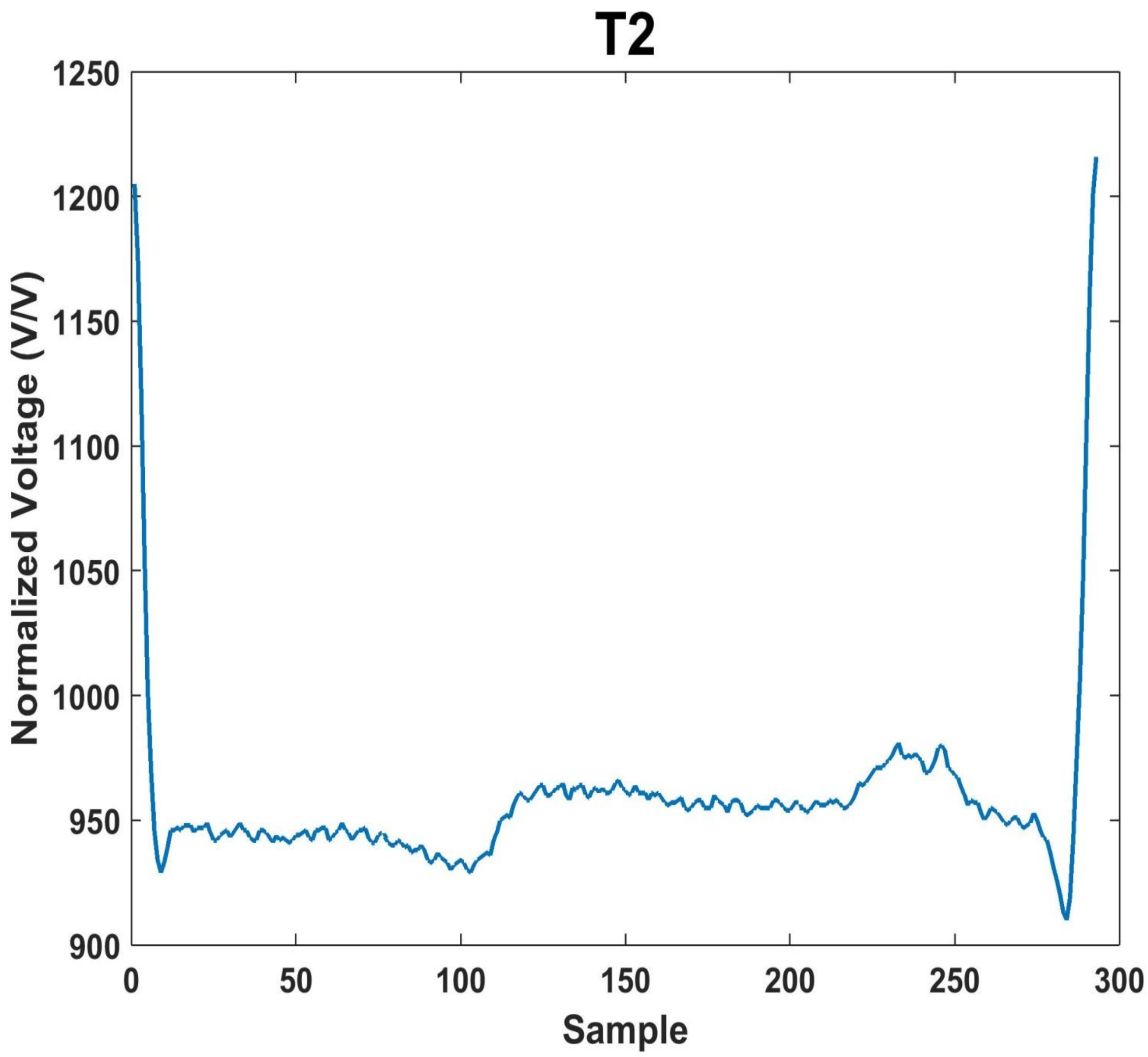


Figure 35. T2 (RR interval) template generated on the CC3200

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:

$$y(n) = \left(\frac{1}{1.5}\right) * (-2 * x(n) + 2 * x(n-1))$$

Wavelet Transform Algorithm

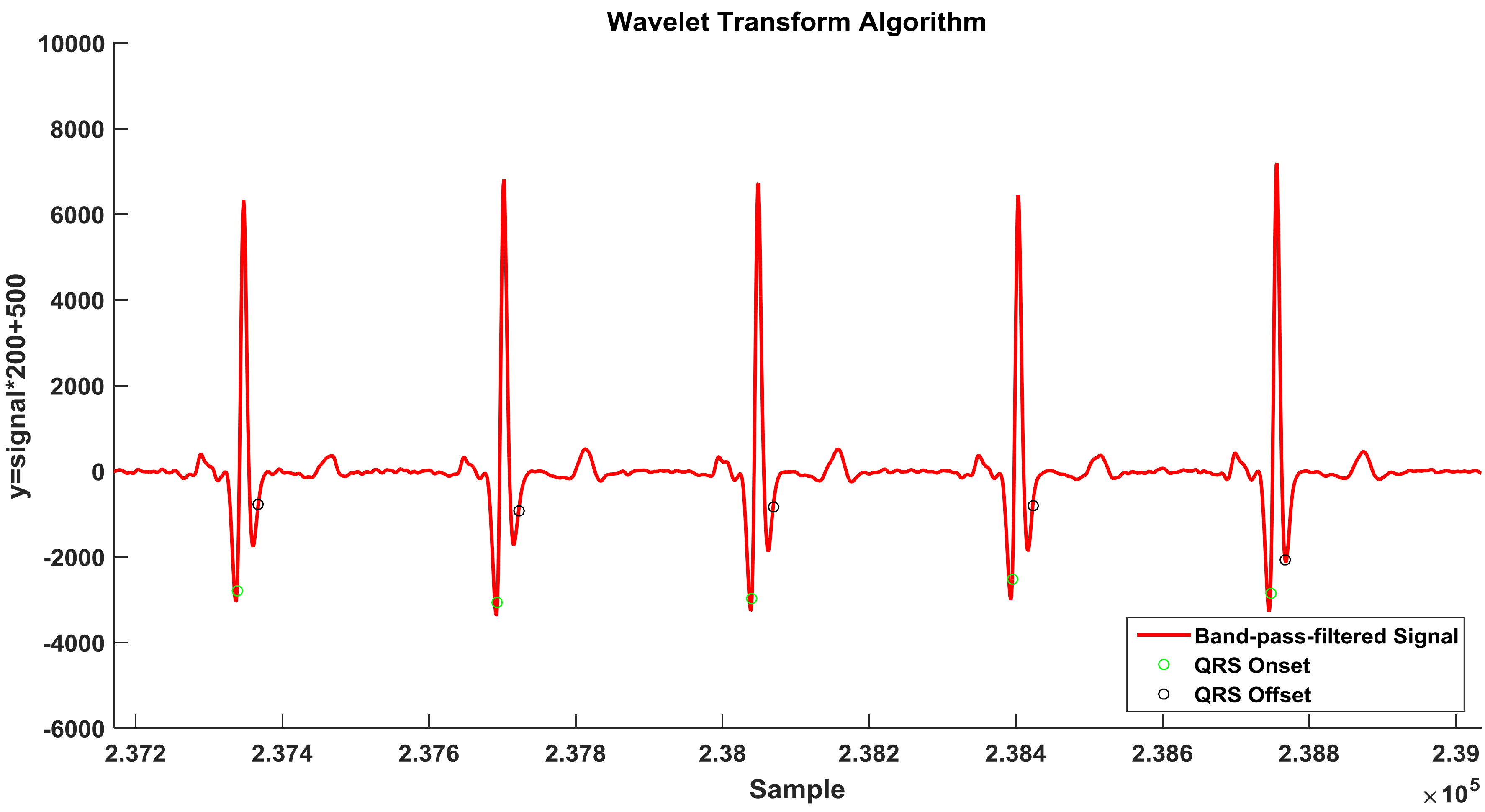


Figure 35. QRS onsets and offsets detected using the wavelet transform

Temboo/Twilio

- Sending an SMS message using the CC3200 LaunchPad
- Message includes text and image file

Temboo/Twilio

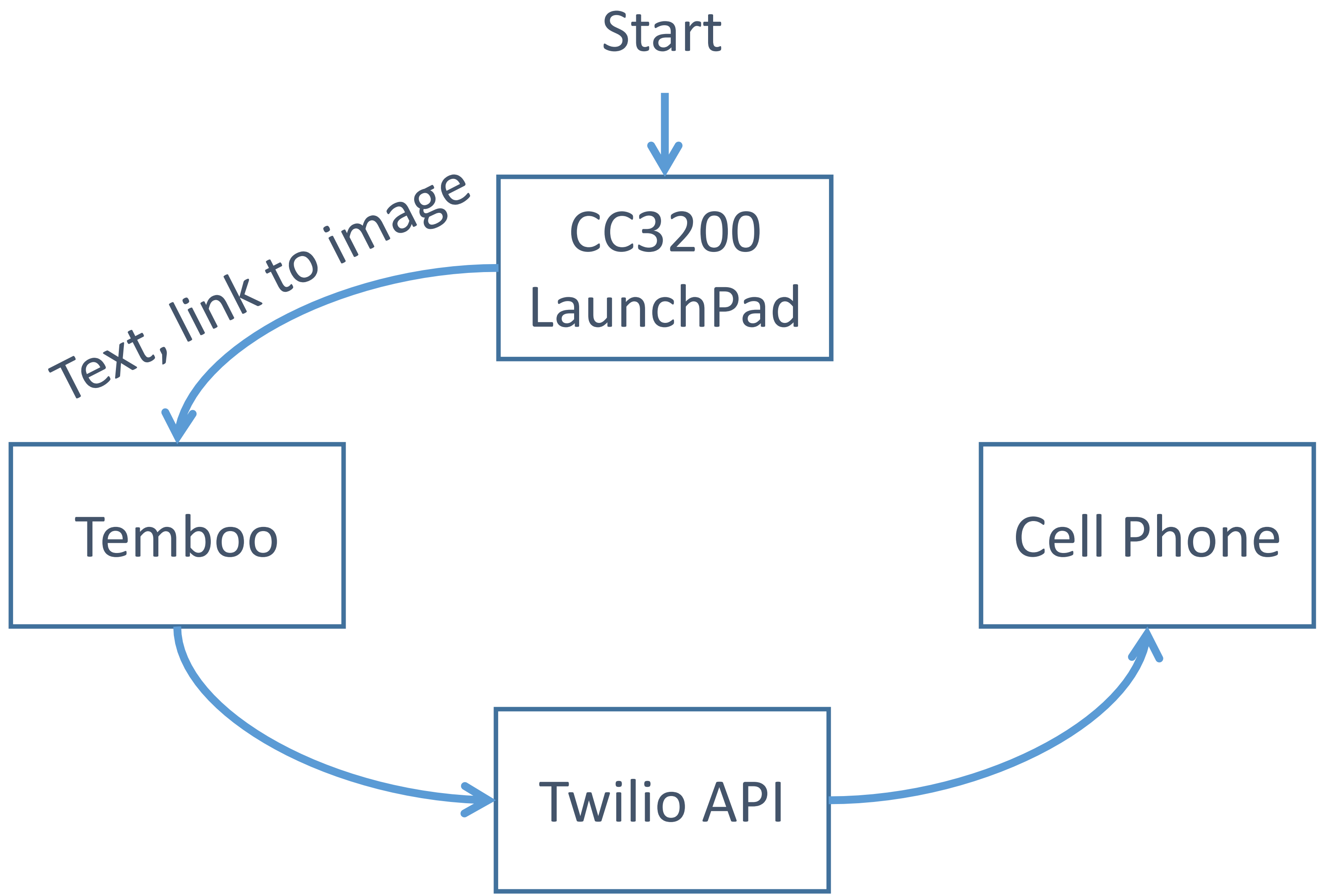


Figure 36. Transmitting an SMS message using the LaunchPad

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

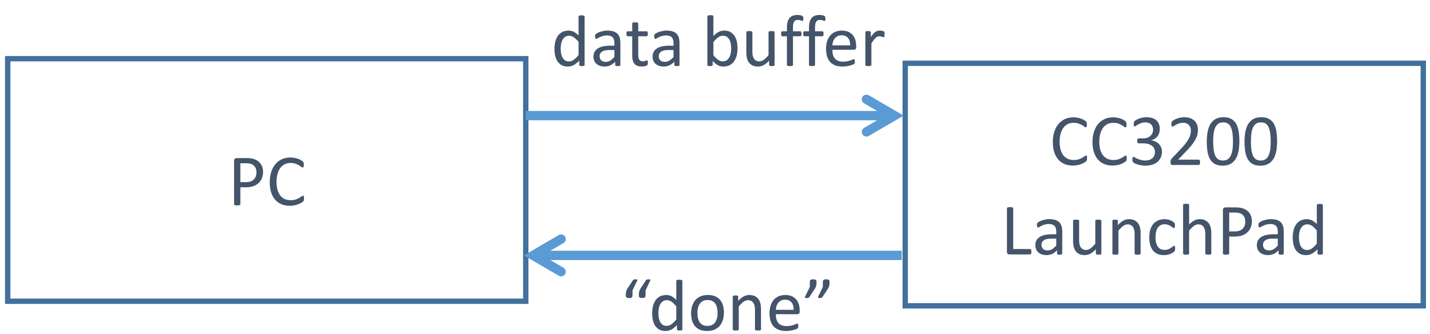


Figure 37. Improved UART testing system

Past Schedule

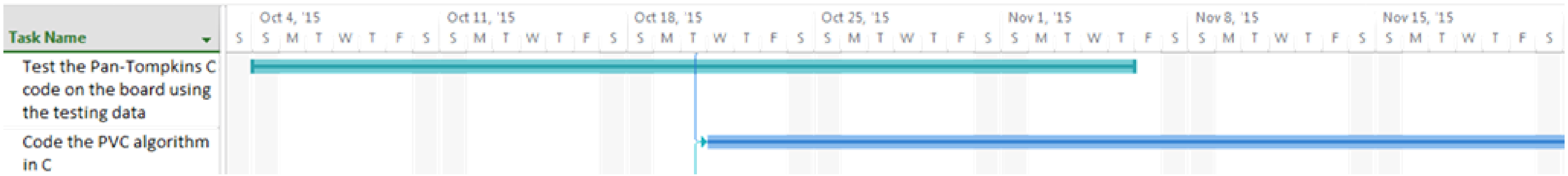
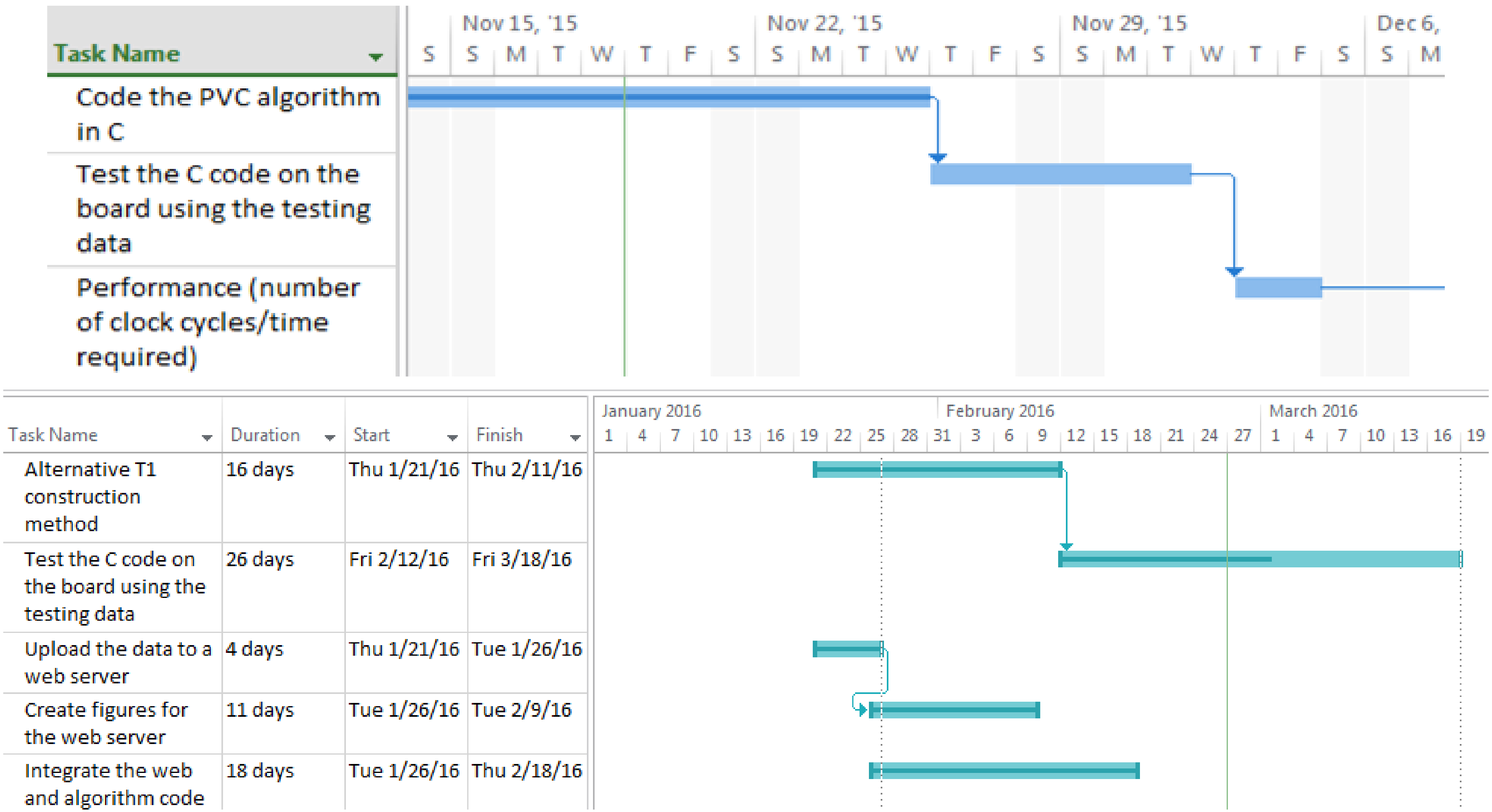


Figure 39. Gantt chart for fall semester

Current/Future Schedule



Task Name	Duration	Start	Finish
Alternative T1 construction method	16 days	Thu 1/21/16	Thu 2/11/16
Test the C code on the board using the testing data	26 days	Fri 2/12/16	Fri 3/18/16
Upload the data to a web server	4 days	Thu 1/21/16	Tue 1/26/16
Create figures for the web server	11 days	Tue 1/26/16	Tue 2/9/16
Integrate the web and algorithm code	18 days	Tue 1/26/16	Thu 2/18/16

Figure 40. Gantt chart through the end of the project

Contents

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

- [1] *Arrhythmias.* [Online] Available: http://watchlearnlive.heart.org/CVML_Player.php?moduleSelect=arrhyt
- [2] *MIT-BIH Arrhythmia Database.* [Online] Available: http://www.physionet.org/physiobank/database/mitdb/
- [3] *Cardiovascular System Assessments.* [Online] Available: http://media.lanecc.edu/users/driscolln/RT116/softchalk/Cardia_Assessment/Cardia_Assessment_print.html
- [4] P. Li, et al., "A low-complexity data-adaptive approach for premature ventricular contraction recognition," *Signal, Image and Video Processing*, vol. 8, no. 1, pp. 111-120, 2013. [Online] Available: http://link.springer.com/article/10.1007%2Fs11760-013-0478-6
- [5] CC3200 –Getting Started with WLAN Station. [Online] Available: http://processors.wiki.ti.com/index.php/CC32xx_Getting_Started_with_WLAN_Station
- [6] Arduino: Dynamic Living-Room Lights. [Online] Available: https://www.element14.com/community/groups/arduino/blog/2014/12/22/dynamic-living-room-lights-the-yun-review--when-the-penguin-met-the-arduino
- [7] A. I. Manriquez and Q. Zhang. "An algorithm for QRS onset and offset detection in single lead electrocardiogram records." [Online] Available: https://www.researchgate.net/profile/Alfredo_Illanes-Manriquez/publication/5845016_An_algorithm_for_QRS_onset_and_offset_detection_in_single_lead_electrocardiogram_records/links/5459ea610cf26d5090ad2c7e.pdf
- [8] *CC3200-LAUNCHXL.* [Online] Available: http://www.ti.com/ww/en/launchpad/launchpads-connected-cc3200-launchxl.html?DCMP=cc3100cc3200&HQS=cc3200launchpad-oob
- [9] C. Li, et al., "Detection of ECG Characteristic Points Using Wavelet Transforms." [Online] Available: http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=362922&url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel5%2F10%2F8315%2F00362922.pdf%3Farnumber%3D362922

Removing Extremities (3σ)

- An amplitude A is kept if it is 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

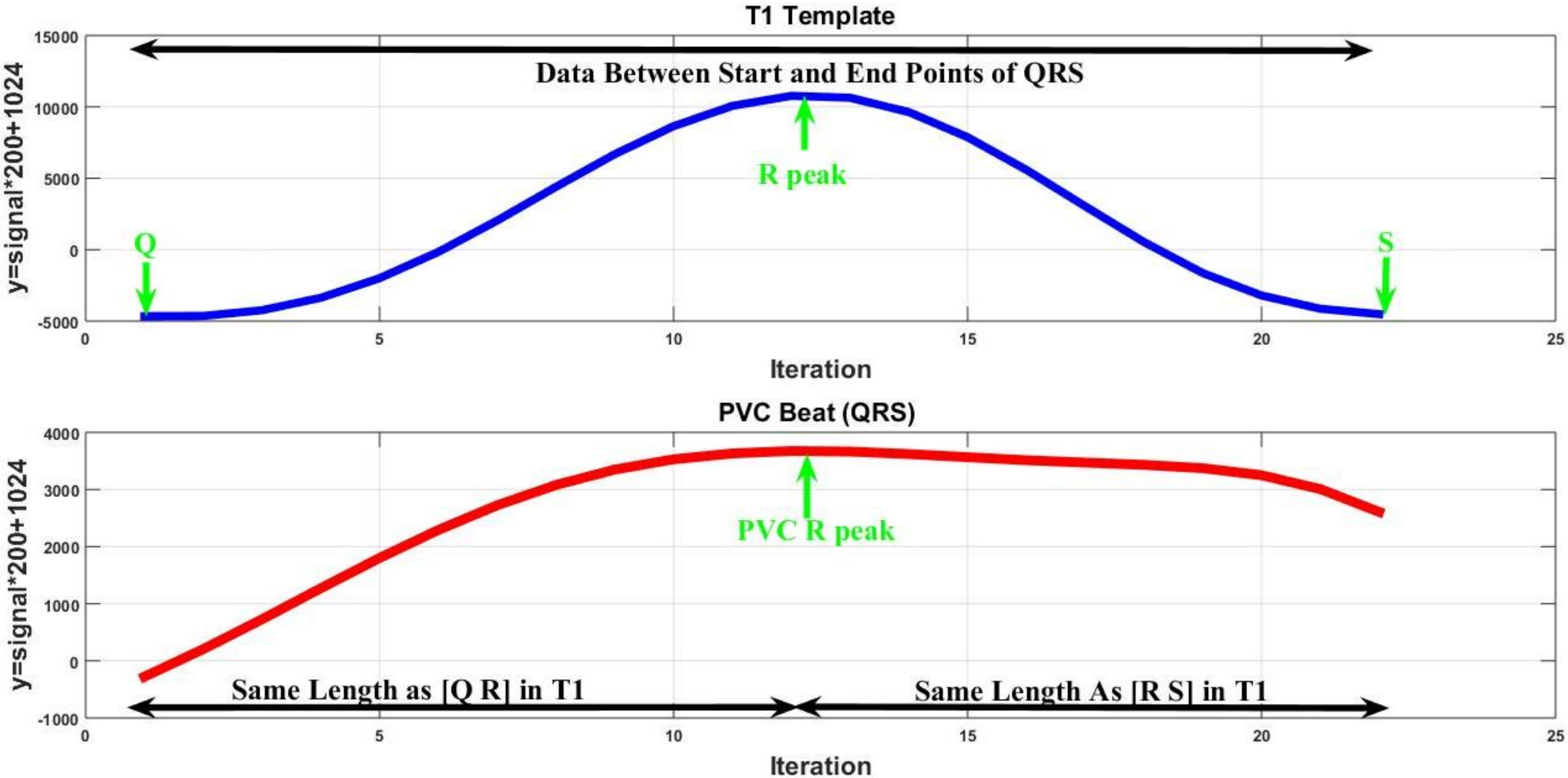


Figure 16. T1 template and PVC QRS complex aligned to T1's R peak index

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

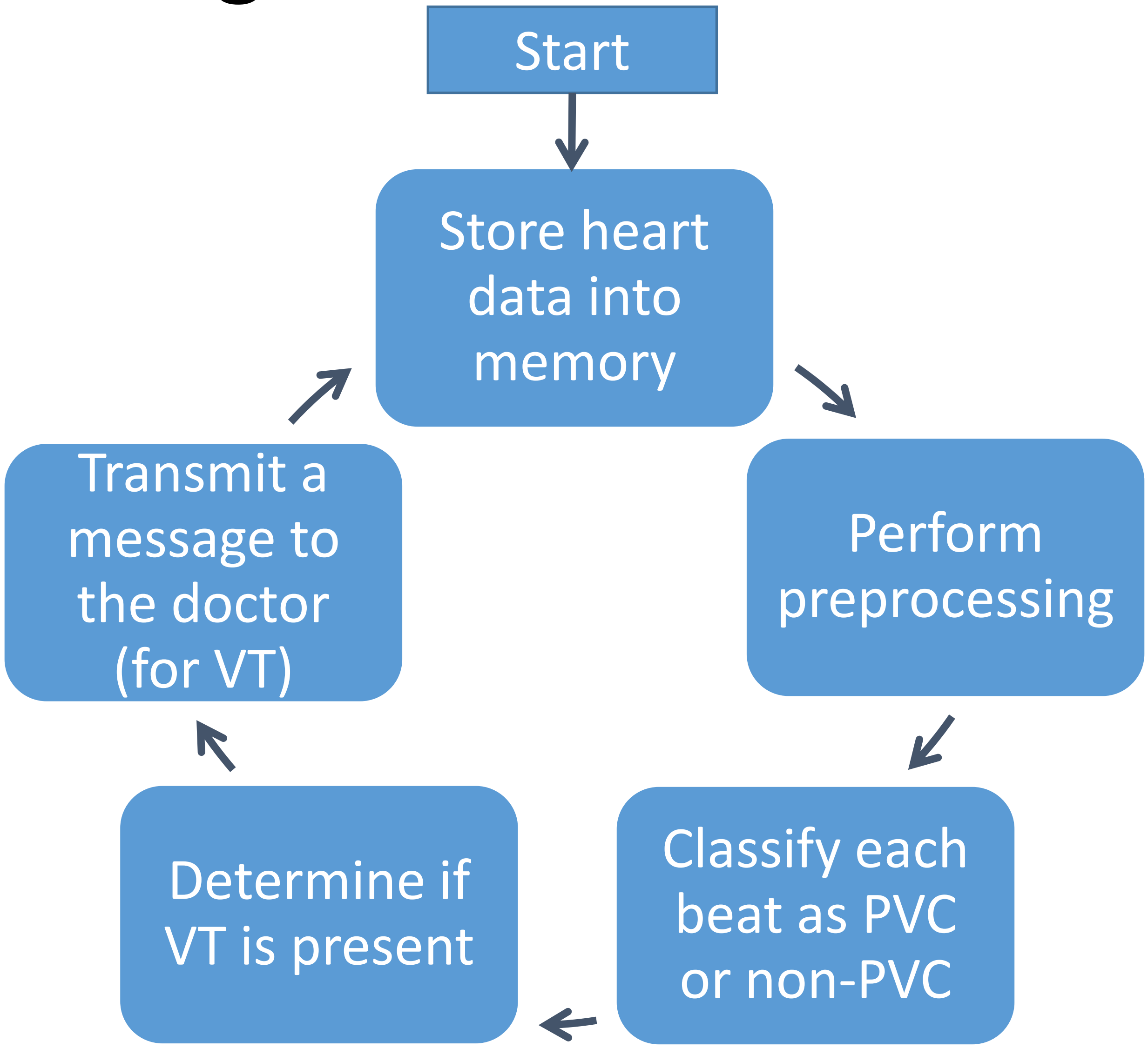


Figure 43. State diagram for heart monitoring system

Detailed Gantt Chart (1)

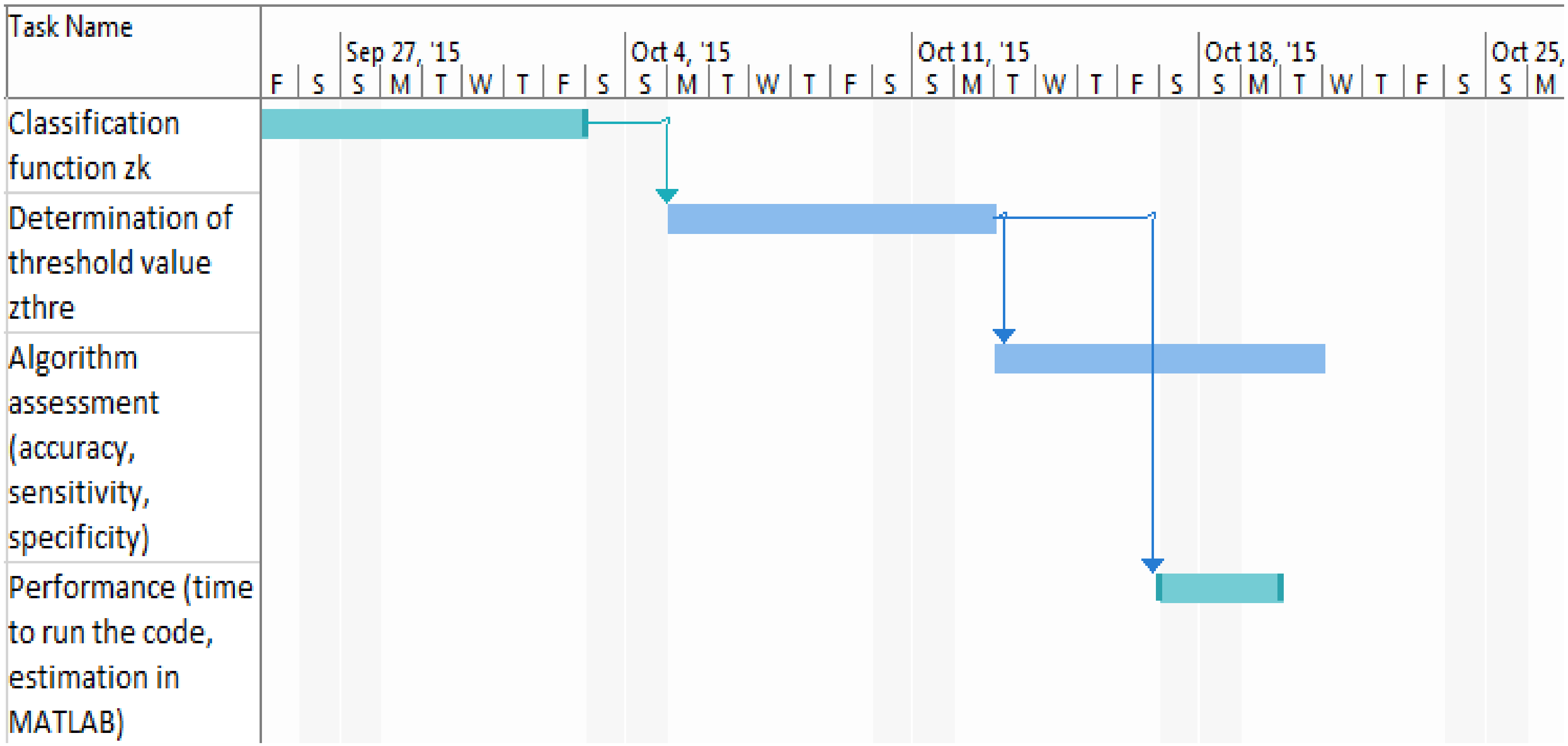


Figure 44. Gantt chart for the MATLAB simulation (PVC algorithm) phase of the project

Detailed Gantt Chart (2)

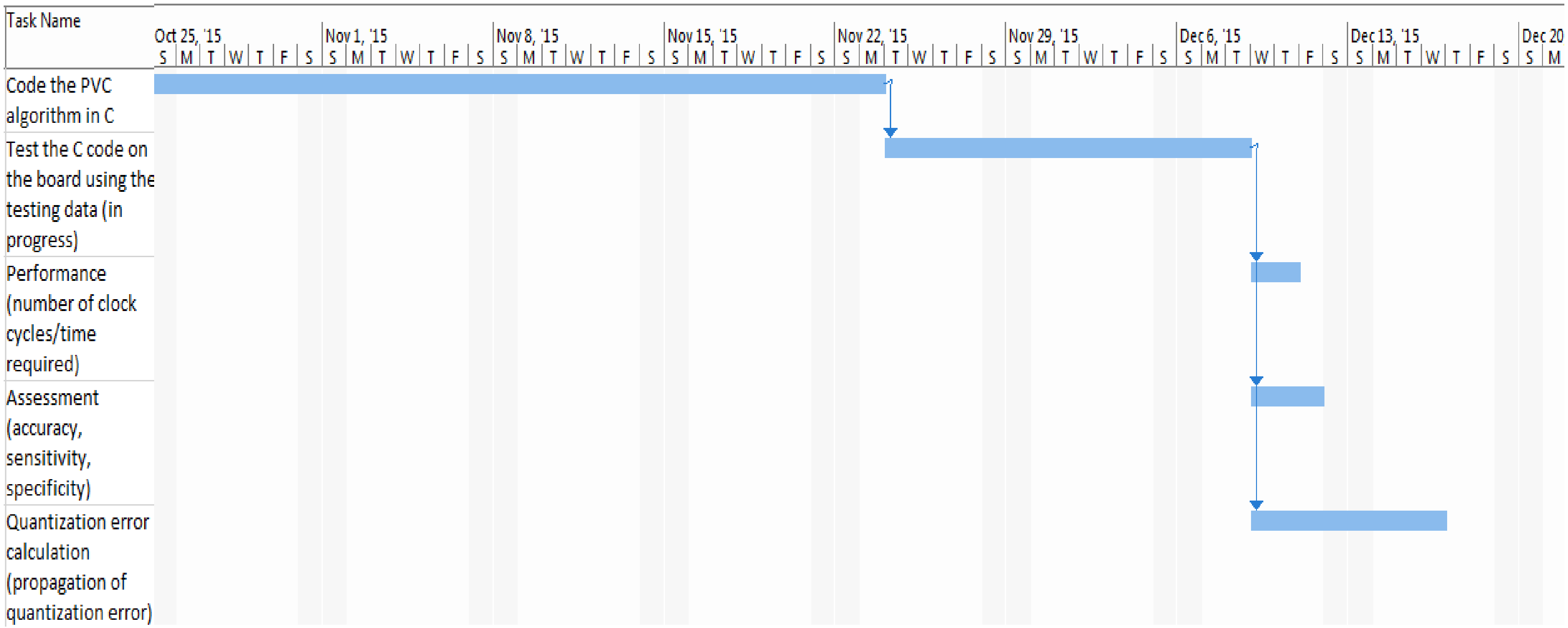


Figure 45. Gantt chart for the C implementation (PVC algorithm) phase of the project

Detailed Gantt Chart (3)

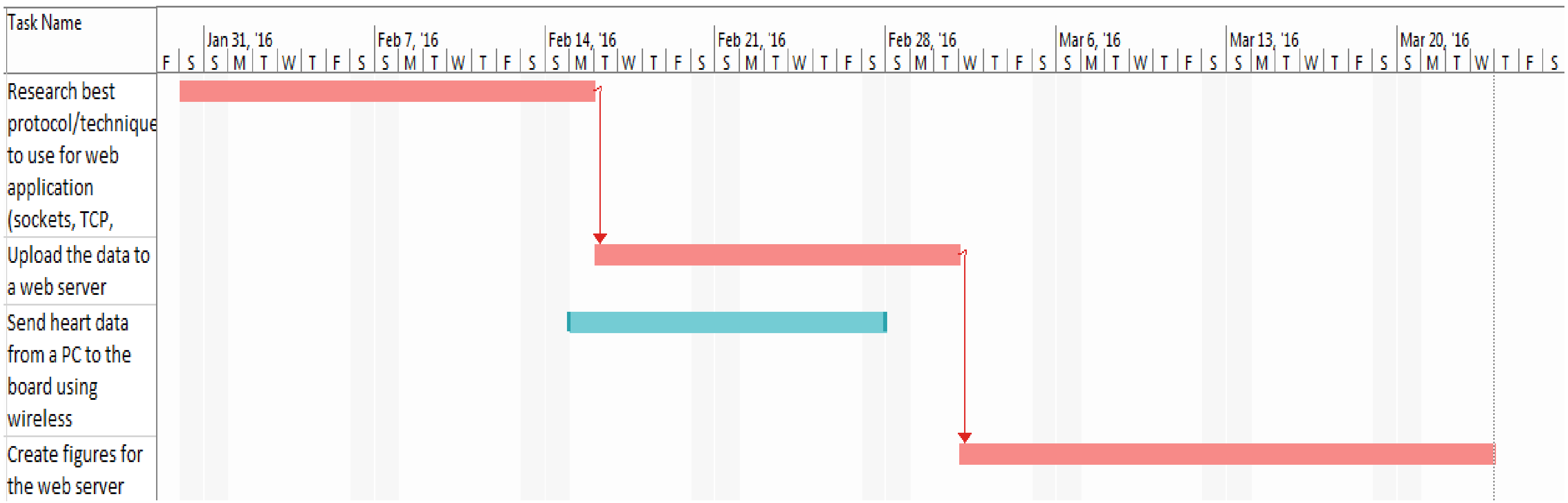


Figure 46. Gantt chart for the wireless development phase of the project

Specificity and Sensitivity [8]

$$SP = \frac{TN}{TN + FP} \qquad SE = \frac{TP}{TP + FN}$$

- 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]

Power Consumption in 24 Hours of Continuous Use (W)	Price ($)	Value Scaled
1.50	500	10
2.50	600	7.5
3.25	700	5
4.00	800	2.5
4.75	900	0

Design Evaluation: Morphological Chart

TABLE V. MORPHOLOGICAL CHART FOR HEART MONITORING SYSTEM [10,11,12,13,14,15,16]

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

Design Evaluation: Design Alternatives

- Total design space: 162 designs
- Two designs analyzed in detail

Schedule

TABLE IV. PROJECT SCHEDULE

Task	Duration (hours)
PVC Algorithm (MATLAB)	65
PVC Algorithm (C)	100
Wi-Fi Communication	150
Progress Report I	80
Progress Report II	80
Final Presentation	80
Final Report	80

Schedule

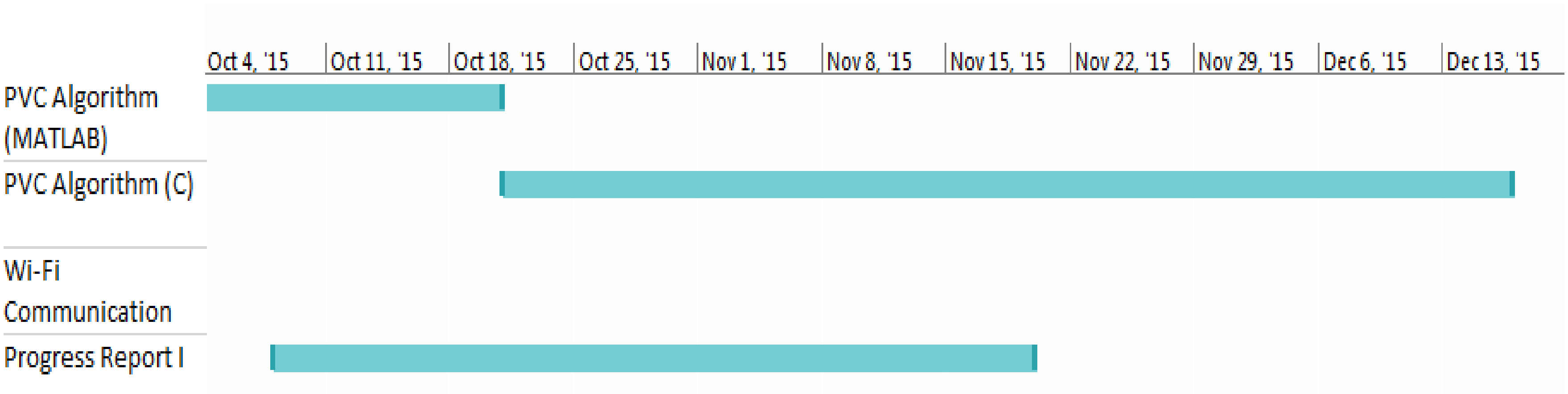


Figure 47. Gantt chart for the fall semester

Schedule

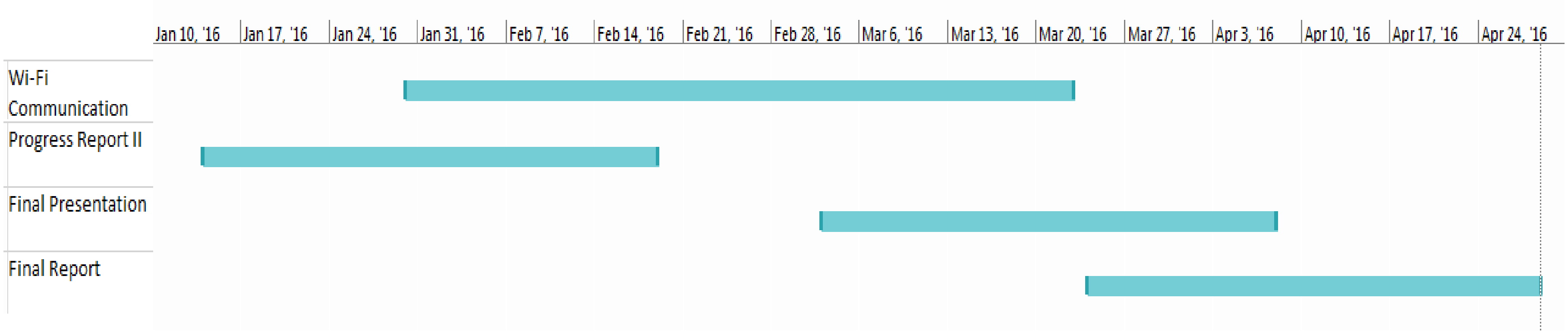


Figure 48. Gantt chart for the spring semester

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

$$y(nT) = \frac{1}{N}[x(nT - (N-1)T) + x(nT - (N-2)T + \cdots + x(nT)]$$

Algorithm Efficacy, 100s

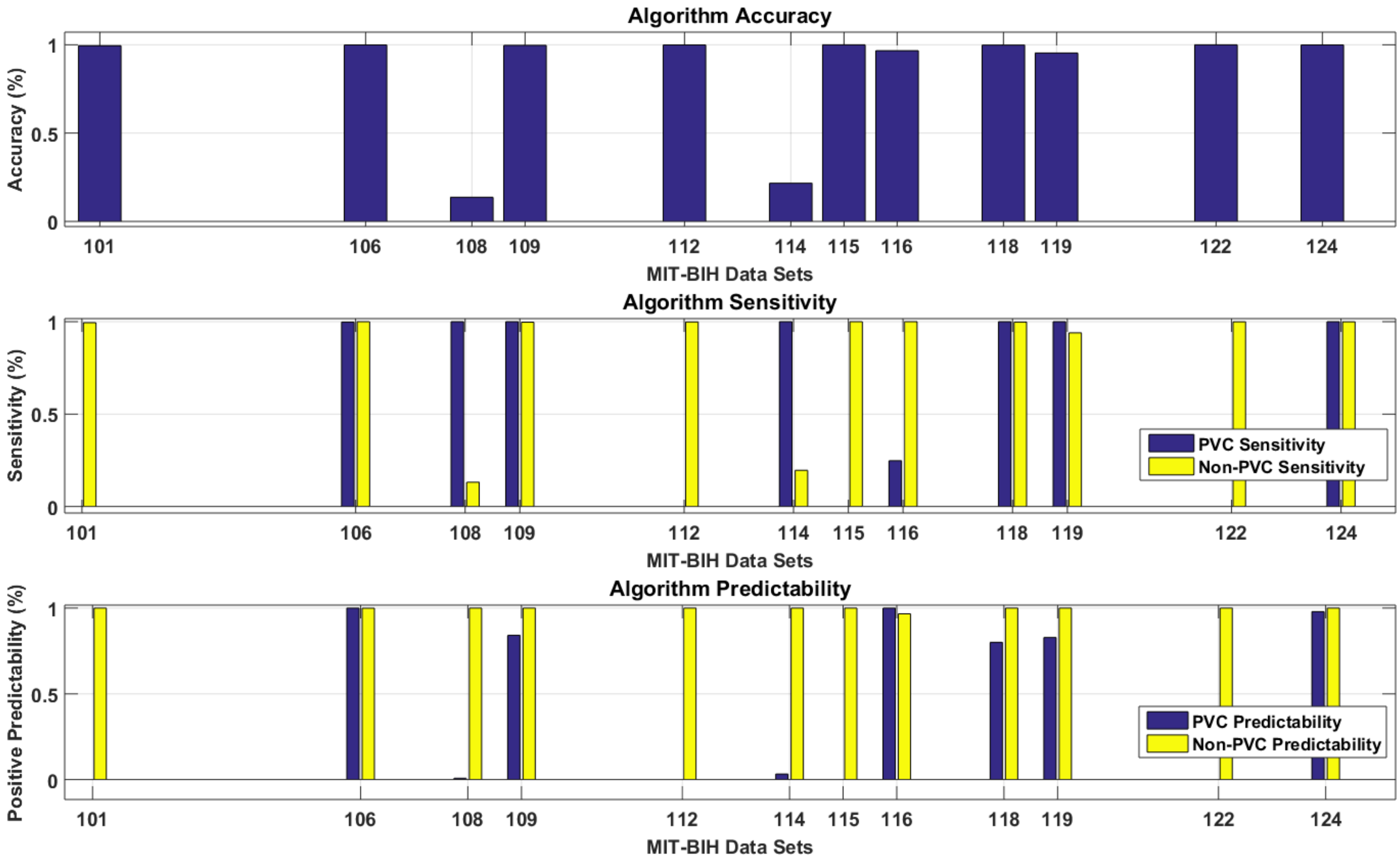


Figure 23. Performance of template-matching algorithm MATLAB simulation

Algorithm Efficacy, 200s

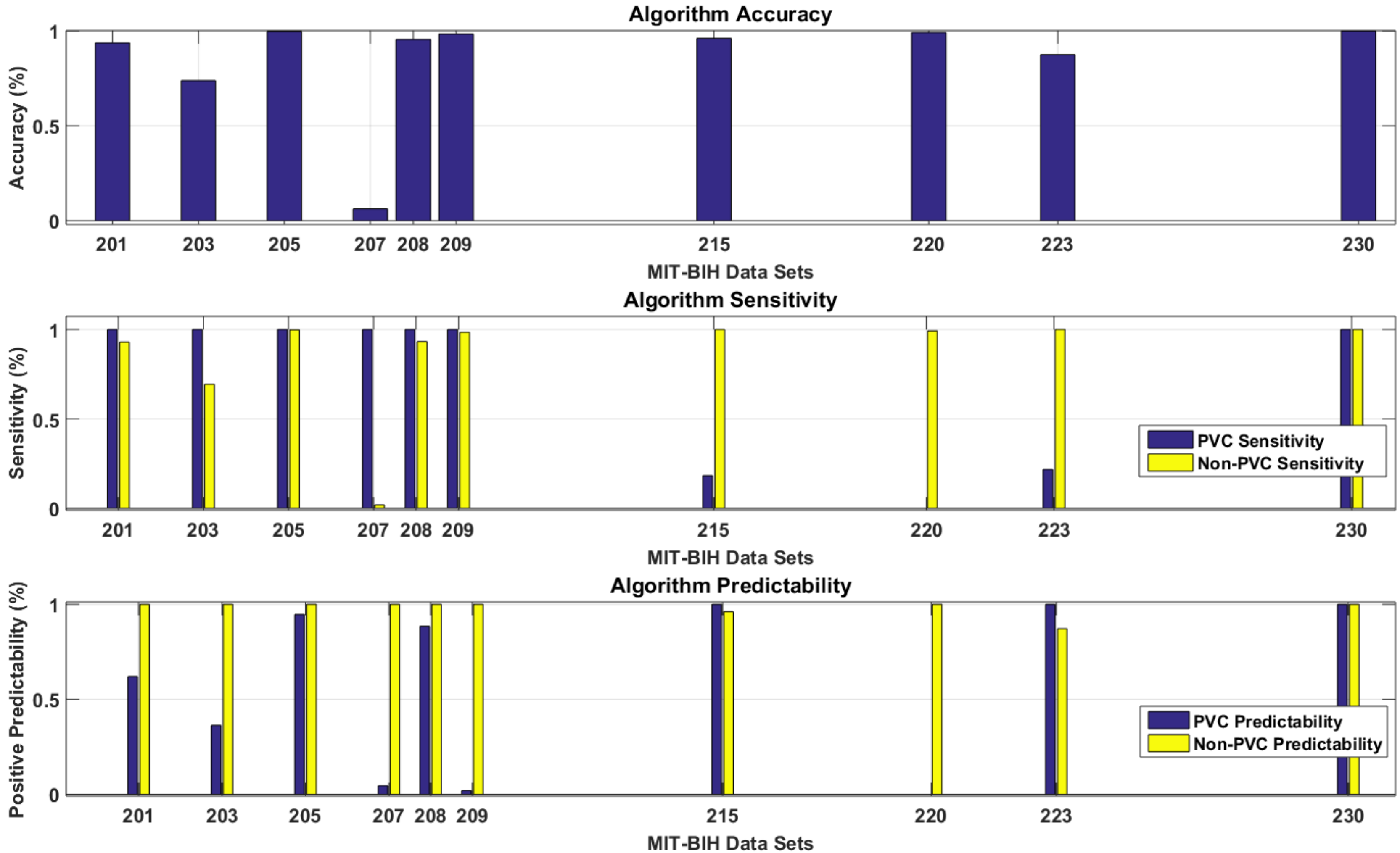


Figure 49. Performance of template-matching algorithm MATLAB simulation

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

```
Beat-by-beat comparison results for record mitdb/100
Reference annotator: atr
     Test annotator: test

                  Algorithm
        n    s    v    f    q    o    x
   ___________________________________
N |   356    0    0    0    0    8    0
S |     4    0    0    0    0    0    0
V |     0    0    0    0    0    0    0
F |     0    0    0    0    0    0    0
Q |     0    0    0    0    0    0    0
O |     0    0    0    0    0
X |     0    0    0    0    0

           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

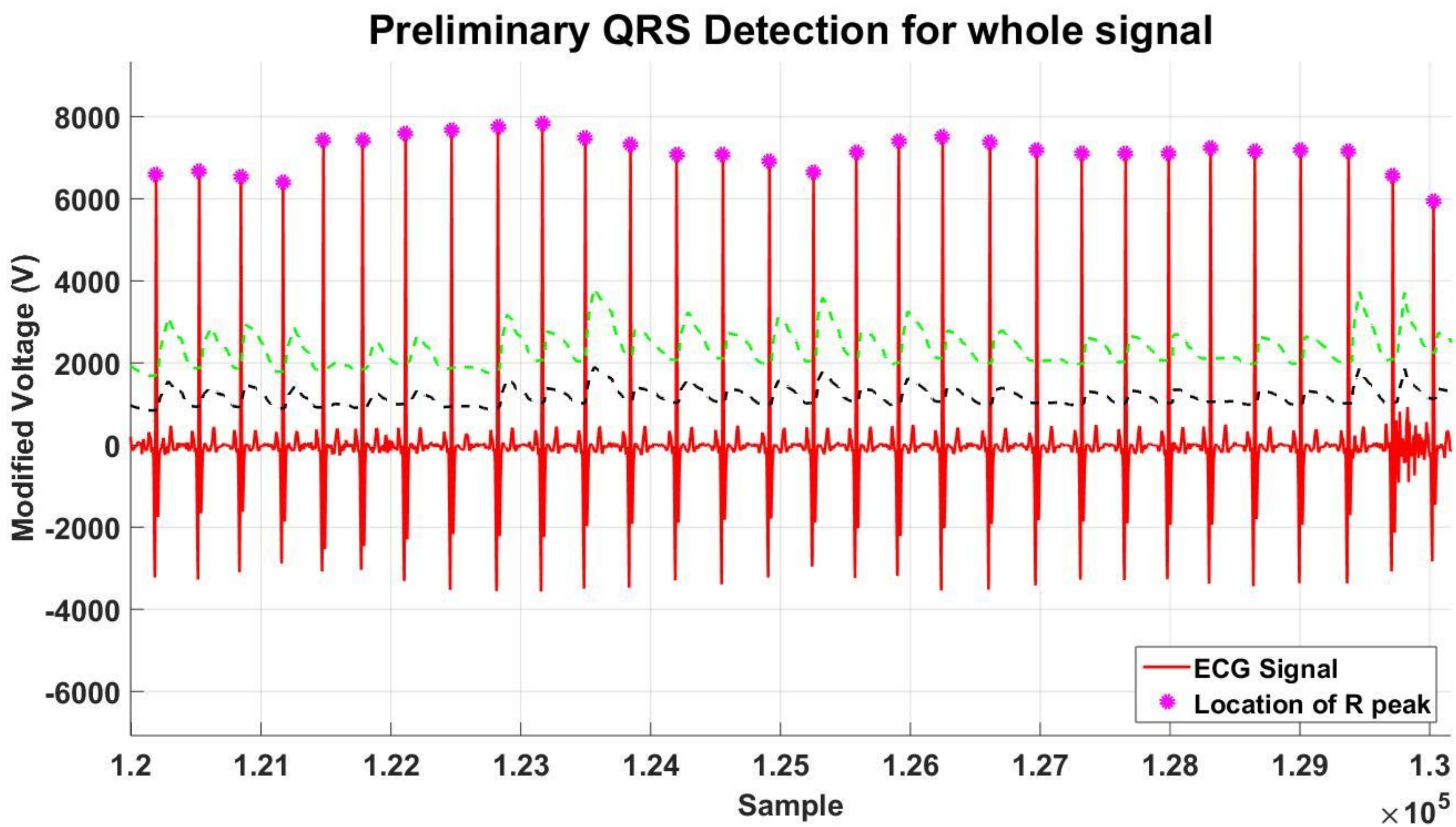


Figure 50. MATLAB plot of QRS detection results

UART Data Transfer

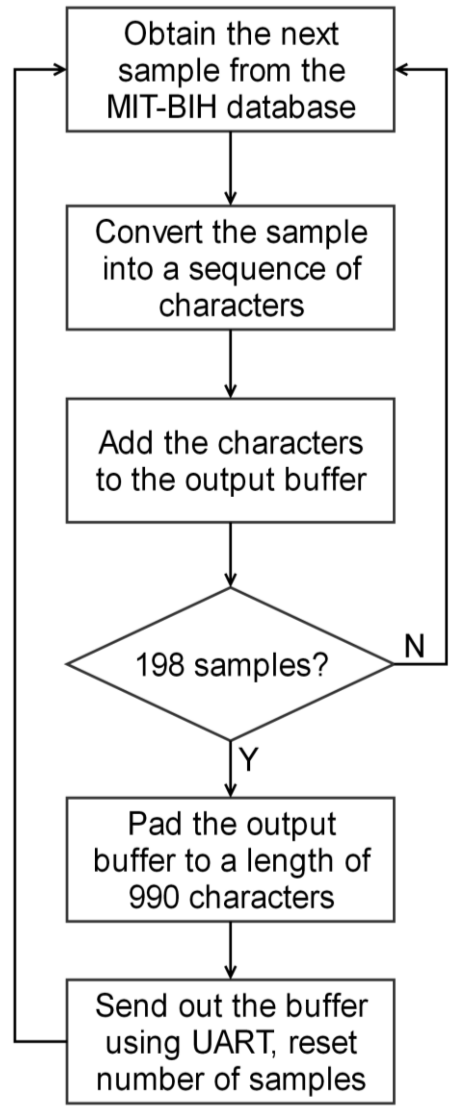


Figure 30. UART data transfer flowchart (PC)

UART Data Transfer

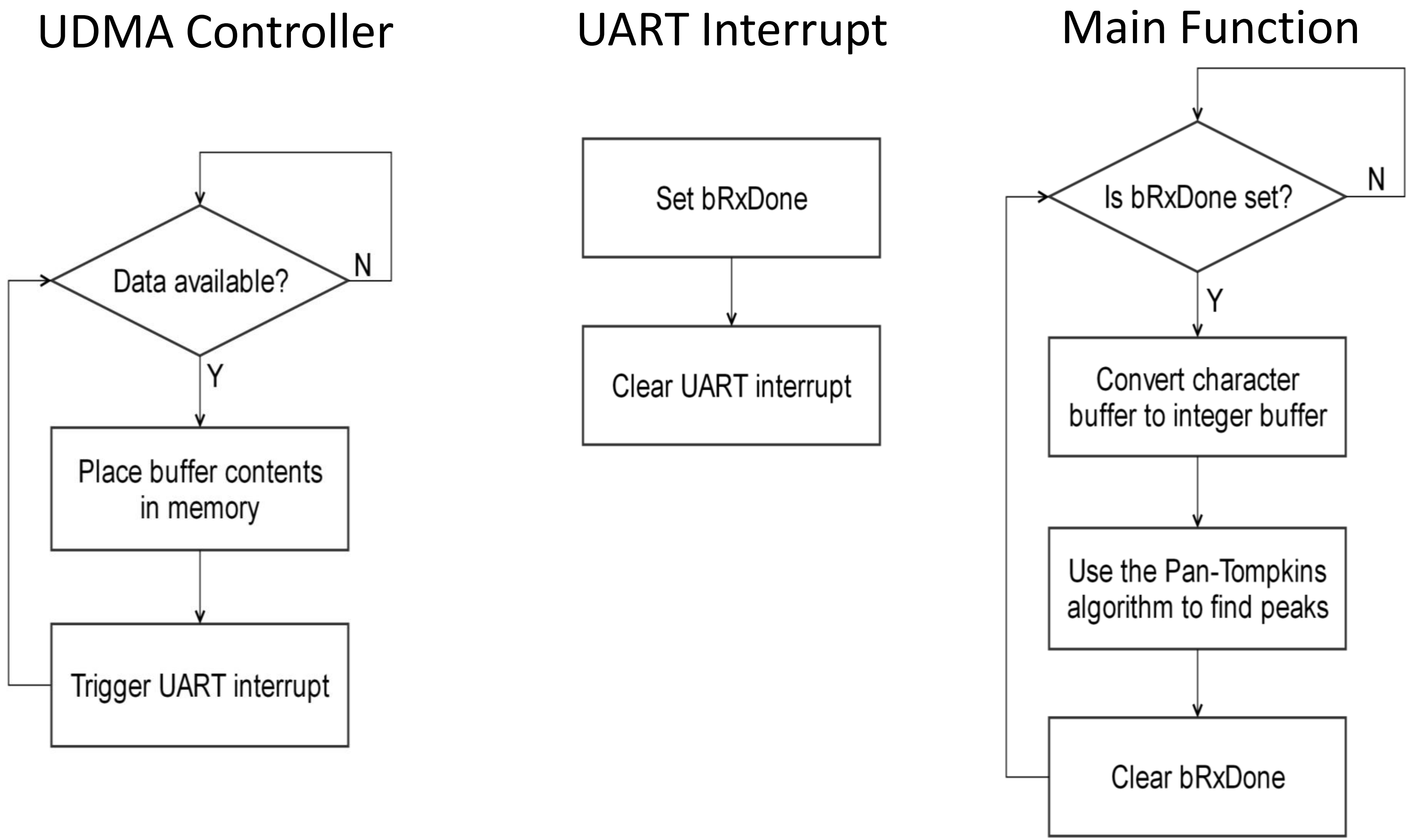


Figure 31. UART data transfer flowcharts (CC3200)

Alternative QRS Detection

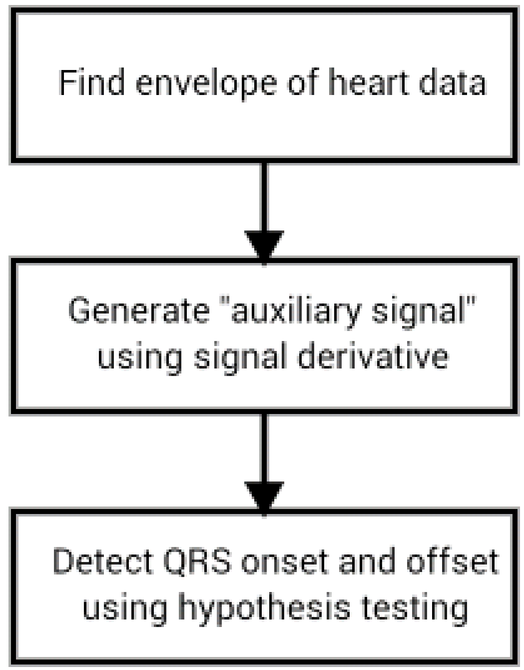


Figure 24. Flowchart for alternative QRS detection method

Alternative QRS Detection

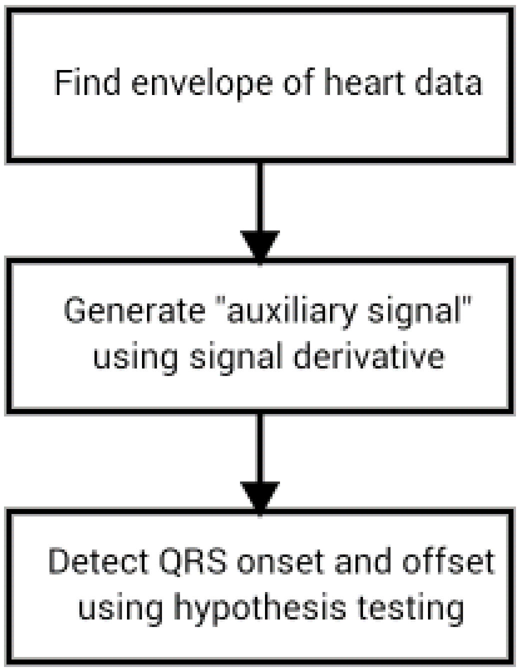


Figure 24. Flowchart for alternative QRS detection method

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

Alternative QRS Detection

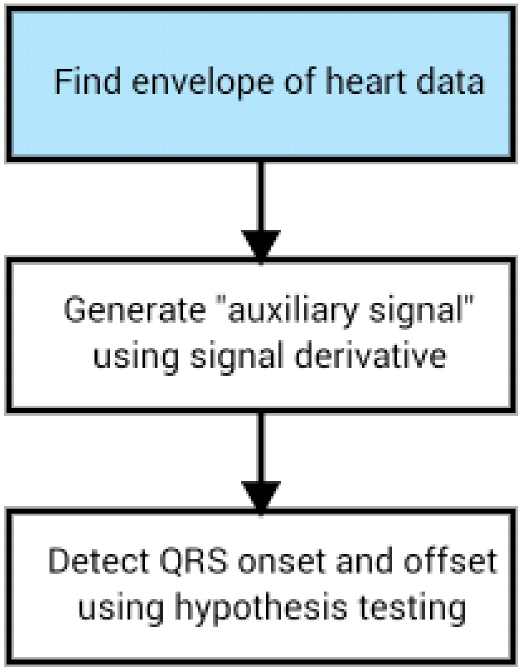


Figure 25. Flowchart for alternative QRS detection method

- Use the Hilbert transform to obtain the envelope

Alternative QRS Detection

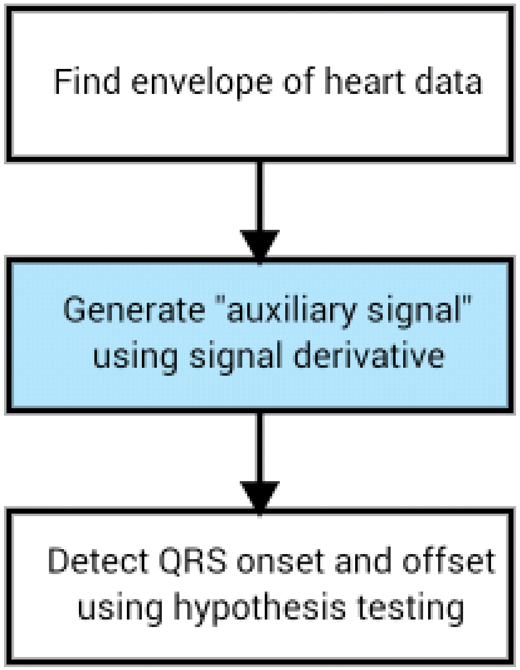


Figure 26. Flowchart for alternative QRS detection method

- Estimate the derivative using a parabola:

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

Alternative QRS Detection

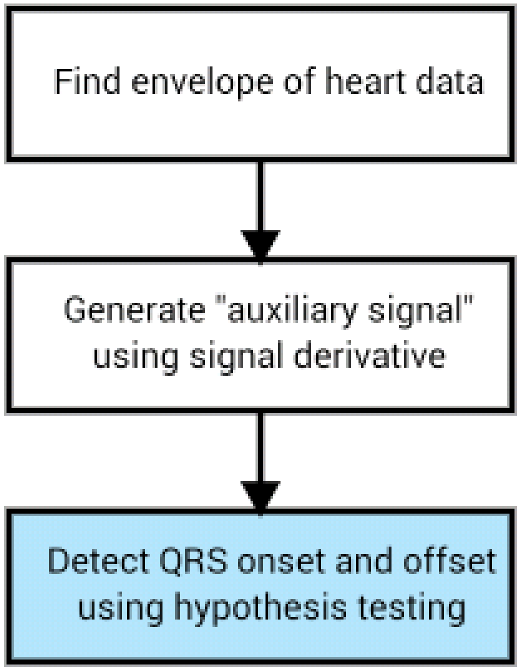


Figure 27. Flowchart for alternative QRS detection method

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

Envelope Signal

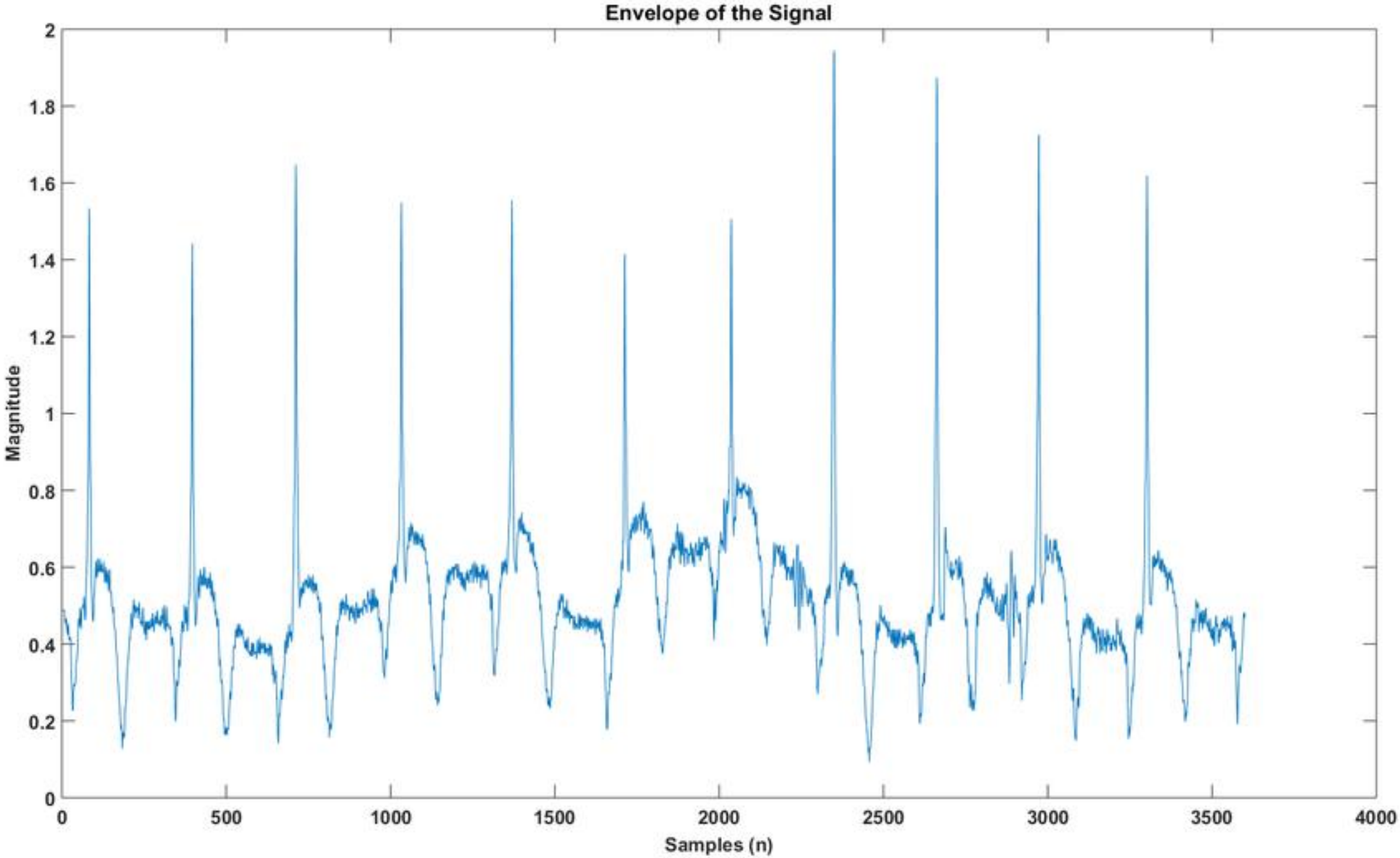


Figure 41. Envelope of the initial ECG signal

Auxiliary Signal

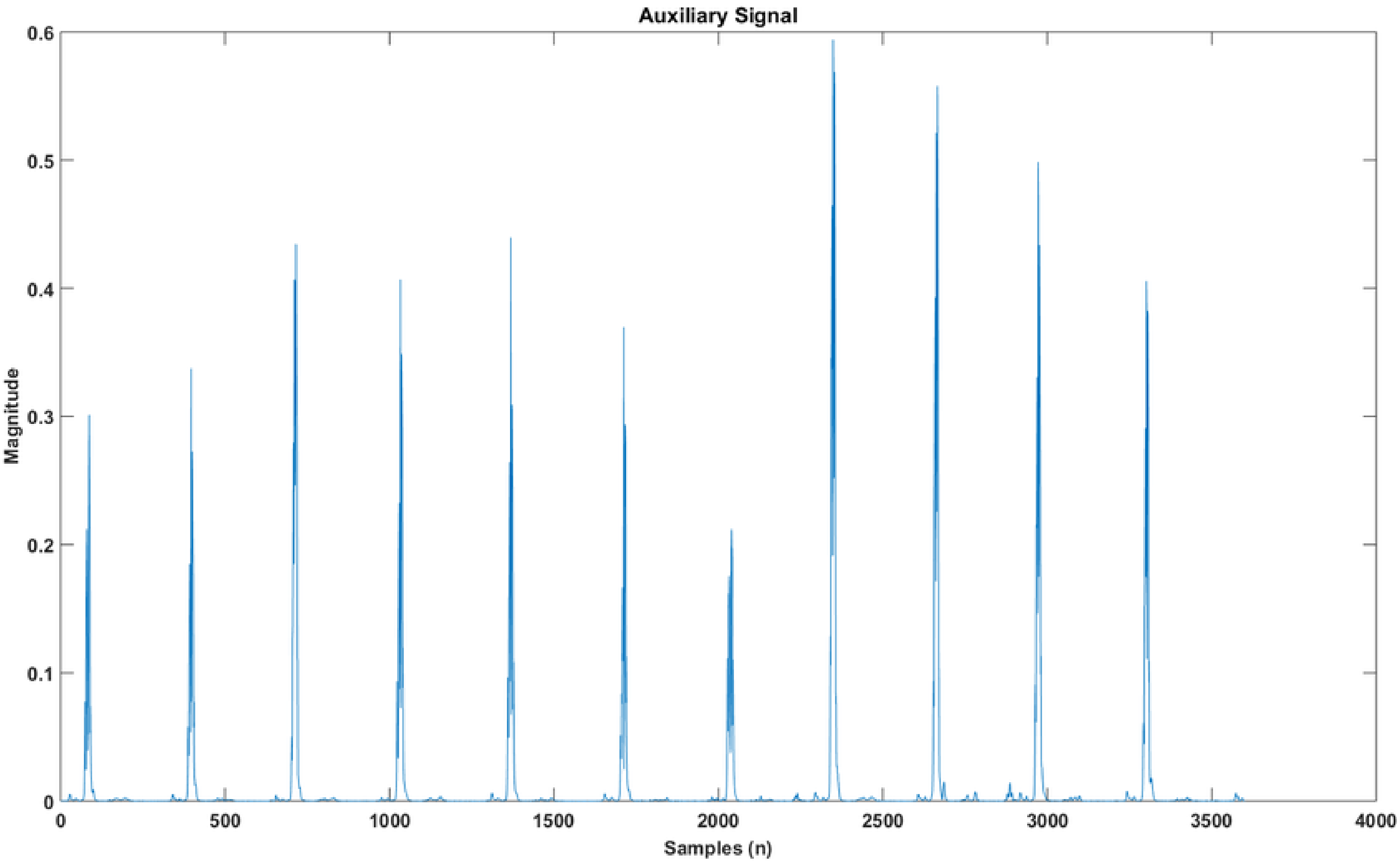


Figure 42. Auxiliary signal of the envelope