An Optimized Heart Rate Detection System Based on Low-Power Microcontroller Platforms for Biosignal Processing

Published Version:
An Optimized Heart Rate Detection System Based on Low-Power Microcontroller Platforms for Biosignal Processing / Mazzoni B.; Tagliavini G.; Benini L.; Benatti S.. - STAMPA. - 546:(2023), pp. 160-170. (Intervento presentato al convegno International Conference on System-Integrated Intelligence tenutosi a Genova nel 7-9 Settembre 2022) [10.1007/978-3-031-16281-7_16].

Availability:
This version is available at: https://hdl.handle.net/11585/899761 since: 2022-11-04

Published:
DOI: http://doi.org/10.1007/978-3-031-16281-7_16

An Optimized Heart Rate Detection System Based on Low-Power Microcontroller Platforms for Biosignal Processing

https://doi.org/10.1007/978-3-031-16281-7_16

An Optimized Heart Rate Detection System Based on Low-Power Microcontroller Platforms for Biosignal Processing

Benedetta Mazzoni[1], Giuseppe Tagliavini[1], Luca Benini[12] and Simone Benatti[3]

[1] University of Bologna, Bologna, Italy,
[2] Swiss Federal Institute of Technology Zürich, Zürich, Switzerland
[3] University of Modena and Reggio Emilia, Modena, Italy

Abstract. The real-time detection of the R peaks of the ECG signal is crucial to provide information on cardiac functionality, and several strategies have been presented in the past. In this work, we adapt the classical Pan and Tompkins (PT) algorithm for efficient execution on low-power microcontroller (MCU) platforms to design a full-fledged heart rate detection system. We target a commercial MCU based on ARM Cortex-M4 and an ultra-low-power solution based on the RISC-V PULP platform. Experimental results show that our approach achieves an accuracy above 99.5%, comparable to the state-of-the-art solutions, and an energy efficiency that is one order of magnitude better than other software solutions.

Keywords: ECG, R peak, heart rate, low- power biosignal processing.

1 Introduction and Related Work

The growing trend of small form factor devices is pushing the development of wearables, driven by systems such as health patches and trackers [1] [2]. In the fitness and healthcare area, these systems facilitate remote and continuous monitoring of wellness conditions providing the extraction of physiological parameters from the analysis of biosignals [3]. However, the main weakness of wearable sensor nodes is the request for high computationally-demanding tasks at high energy efficiency to improve the battery life-time.

Several works exploit digital platforms capable to execute digital signal processing (DSP) to achieve ultra-low power (ULP) consumption [4] [5]. In this context, the designers typically adapt optimization strategies to reduce the algorithm complexity and find the best trade-off between reliability and low power consumption. Among the biopotentials that can be acquired with real-time low power devices [6], heart activity parameters are the most used to detect and monitor acute severe conditions. Analysing the QRS complex and detecting R peaks is crucial for providing cardiac functionality information. Several strategies have been presented in the literature, using well-established signal processing techniques. Park et al. [7] propose a technique based on a wavelet transform (WT) coupled with the Shannon energy envelope method in addition to a moving average filter and a squaring operation for the preprocessing step. This method achieves accuracy over 99%. However, the algorithm is computationally intensive, and it is not suitable for real-time execution on an ultra-low-power embedded device. Martinez et al. [8] adopt the phasor transform. This approach converts each ECG sample into a complex number maintaining the phase and the root mean square values to enhance the wave variations to distinguish them from each other. Also in this case, the overall accuracy is higher than 99%. However, the analysis excludes five records from the MIT-BIH Database because of the low-quality acquisition of highly-variable signals or noise distortion.

A widely explored family of approaches for ECG signal analysis includes *slope-based methods*. In Tekeste et al. [9], the authors optimize peak detection by providing a hardware unit to approximate the computation of the signal derivative. The power consumption is only 3.9 nW at an operating frequency of 3 kHz implemented in 65 nm technology. Nevertheless, they do not consider the contribution to the power consumption of the additional computations that are strictly required by a real-life scenario. In our work, we use microcontroller-class devices that can perform pre-processing, peak detection, and subsequent computations. De Giovanni et al. [10] propose a software-based methodology that can be considered the current state-of-the-art. Their algorithm implements a Bayesian filter, normalization, and a clustering technique to optimize the R peak detection. The authors test the system on a biosignal dataset where sudden event changes occur, such as during intense physical exercise [11]. These physical conditions reduce the robustness of the traditional algorithms affecting their reliability. Hence, they propose an accurate adaptive design for low-power platforms. However, the authors do not consider the aspects related to real-time signal acquisition. They use an existing system (BIOPAC) that requires a 9 V battery and is not energy efficient. Furthermore, the peak detection algorithm including all the proposed phases is very complex and requires a core with native FPU support because the fixed-point representation decreases the accuracy significantly. Overall, we will show that their results in mJ are 7× higher than our method.

An effective and computationally efficient threshold-based approach for QRS extraction and heart rate (HR) calculation is the Pan and Tompkins algorithm (PT). It relies on an adaptive dual-threshold technique for the R peaks detection [12], leveraging a filtering stage and simple adaptive thresholding methods. PT is a robust technique that uses a pre-processing pipeline that includes standard filtering techniques (pass-band, derivative, squaring, integration). This technique can also be applied to signals with arrhythmia. Furthermore, it can be adapted to execute on real-time streaming data, which is crucial in the context of wearable systems. PT is a standard approach that was proposed several years ago, but recent works adopt this methodology yet [13] [14]. We outperformed the accuracy and energy consumption of these works, optimizing the R peak detection on our target architecture.

In this work, we propose a lightweight design for HR computation based on the PT algorithm. We implement the PT aiming for an acceptable trade-off between computational complexity and energy efficiency. Furthermore, we propose a real-time application for ECG monitoring based on an end-to-end system from the data acquisition to the inference. The proposed methodology is optimized by simulating in Matlab the real-time operation and then implementing it with a multi-board setup. Then, the processing is coded in C language and can work in data-streaming or with an existing dataset. The proposed system provides a power budget of less than 5 mW, for wearable and near sensors processors. We aim to process ECG signal to carry out the HR, which is a crucial physiological parameter to detect anomaly conditions in the heartbeats [15]. Our methodology obtains an acceptable HR detection reliability (higher than 99%) in pathological or sudden changes of the biosignal. The target device that we use for experimental assessment is the Parallel processing Ultra-Low Power (PULP) many-core platform designed for smart ULP embedded devices [16]. For the evaluation, we

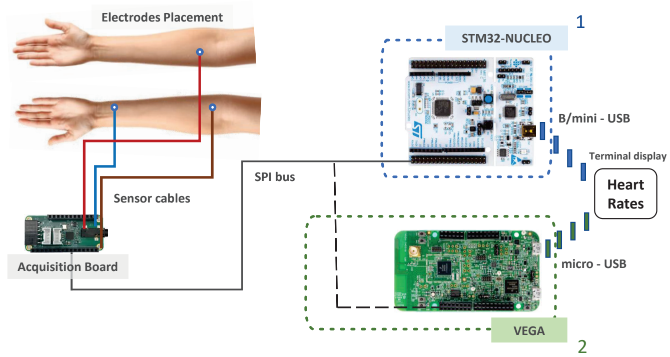

Fig. 1. Hardware diagram of the proposed system. The active electrodes are located on each forearm and one on a wrist, setting Lead I for the collected data. Single-channel ECG is acquired with a custom AFE board (MAX30003), which sends data via SPI to the platform for processing. We consider two alternative designs: (1) STM32NUCLEO for the initial setup and (2) Vega for ULP optimization. Output and communication are managed via a B/mini-USB and a micro-USB cable, respectively, that leads the platform to a terminal to visualize the HR values.

analyze the performance on the Vega SoC [17], a PULP platform running at 0.8 V at an operating frequency of 170 MHz, and on ARM Cortex-M4, using the STM32 NUCLEO-F401RE development board at 1.8 V and 84 MHz. The PULP provides extreme energy efficiency, and we obtained an energy consumption of 0.2 mJ when considering an average on 25 s of running time. We performed tests on four datasets, three existing ones and one acquired from the proposed system in real-time, taking into account several options: normal conditions, arrhythmia [18], intense physical exercise [11]. Overall, we achieved accuracy above 99.5% that we compared with other state-of-the-art solutions.

2 Methodology

2.1 System Architecture

This work proposes a modular setup for ECG detection. The acquisition board relies on Maxim MAX30003 [19], a chip for ULP acquisition of ECG. MAX30003 is a complete, biopotential analog front-end solution for wearable applications. It offers high performance for clinical and fitness applications at extreme energy efficiency, reaching 85 μW average power consumption. The analog acquisition is based on a 2 leads differential channel providing ECG waveforms and heart rate detection. The biopotential channel has ESD protection, EMI filtering, internal lead biasing, and DC leads-off detection. The biopotential channel also has high input impedance, low noise, high CMRR, programmable gain, as well as low-pass and high-pass filter options. The digital back end is based on an SPI interface to enable data streaming and communication with an external MCU.

Fig. 1 depicts the custom board equipped with MAX30003 and with two alternative test benches: the first one with NUCLEO-F401RE board, used for initial setup and algorithm tuning, and the second one with Vega custom board [17], employed for ULP operation and optimized performance. In both test benches,

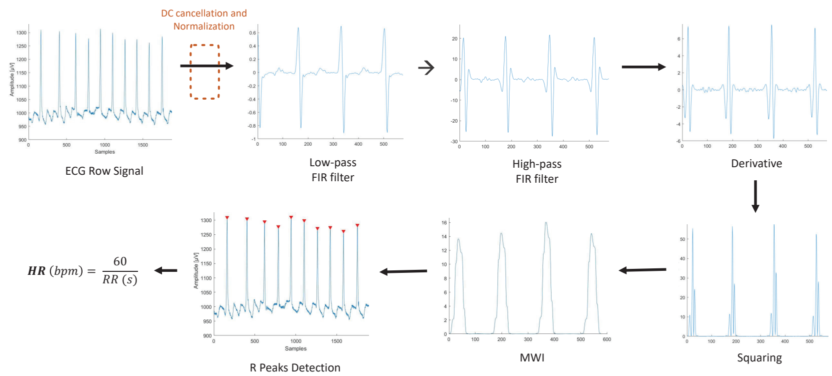

Fig. 2. Signal processing steps based on PT technique: (1) Cancellation of DC component and addition of Normalization; (2) Band-pass filter that combines the low- and the high-pass filters; (3) Derivative function; (4) Squaring function; (5) Moving window integrator (MWI); (6) R peaks detection. In the last step we compute the HR in beats per minute.

ECG data are sent from AFE to MCU via SPI. Vega allows a USB device mode interface with a micro USB connector at an operating frequency of 2.4 GHz with a reference oscillator frequency of 32 MHz. Vega receives data from AFE via a 5 MHz SPI channel connection (Vega acts as master). Data loaded via SPI is stored in the Vega L2 memory as 24-bit signed fixed-point numbers, with the least significant bit. Acquired ECG samples are used as input of the embedded implementation, described and profiled in Section 3.

The presented prototype can be integrated into a single PCB with a 20×10 mm form factor, suitable for minimally obtrusive wearable applications. We use a recent SoC implemented in 22 nm technology, namely Vega [17]. It provides a DSP-oriented instruction set architecture (ISA) based on the Parallel Ultra-Low-Power Platform (PULP) [16]. The PULP SoC is equipped with a 2 MB SRAM scratchpad memory (L2), hosting the resident code and application data. A hardware unit called μ-DMA performs autonomous data transfers between the L2 memory and the peripherals. The peripherals and the MCU core reside in different clock domains so that the frequency of each domain can be tuned to sustain the application workload with low power consumption (up to 500 MHz for a 22 nm technology node). The clocks of the peripherals can be further divided to match the operating frequency of slower external devices.

2.2 Algorithm Description

To compute the HR, we adopt a signal processing pipeline based on the PT technique [12]. This methodology adopts a dual-threshold technique to detect the R peaks and includes multiple pre-processing signal steps required to improve the signal analysis. The block diagram is depicted in Fig. 2.

The signal processing pipeline includes a set of pre-processing digital filters followed by the computation of R-peaks. The original implementation considers a sampling rate of 200 Hz.

1. **Band-pass filter**. The low-pass component applies a second-order transfer function to the signal obtaining a difference equation with a delay of 5

samples and a DC gain of 36. The high-pass design is characterized by a first-order transfer function, with a delay of 16 points and a gain of 1. Overall, the bandpass filter provides a 3 dB pass-band between about 5 and 12 Hz and reduces the noise due to the muscle, the baseline wander and the T-wave interference/frequency content. This filter supplies poles and zeros only on the unit circle, so the system is characterized by a minimum phase, a minimum group delay, and better stability. As a final effect, it increases the signal-to-noise ratio.

2. **Derivative**. The signal is differentiated using a 5-points derivative. The result provides information about the slope of the input waveform. This filter introduces a delay of 2 samples and a gain of 0.1.
3. **Squaring**. The output of the derivative signal is squared to enhance the R peaks, leading the signal to the positive y-axis to emphasize the high frequencies that include the R-peaks. This step makes it easier to distinguish R peaks from T-waves.
4. **Integration**. From the output of the squared signal, a moving window integrator extracts the duration of the QRS complex, obtaining a time-averaged signal. Usually, the window length is equivalent to the widest QRS complex (around 150 ms, corresponding to 30 samples at 200 Hz). The time of the rising direction of the window is the duration of the QRS complex.
5. **Computation of R peaks**. The final part of the algorithm finds a set of fiducial marks corresponding to the temporal location of the peaks in the integrated signal. Fiducial marks determined in this area are potential candidates for R peaks. An initial phase of the implementation is necessary for the tuning (2 seconds at 128 Hz). The fiducial mark is compared with a threshold value $threshold_{I1}$ that considers the current estimation and both signal and noise peaks:

$$threshold_{I1} = npk_I + 0.25 * (spk_I - npk_I) \tag{1}$$

where npk_I is the estimation for any peak that is not related to an R peak (e.g., the peaks of T waves), and spk_I is the estimated value for the R peak level. When a new peak $peak_I$ is detected, it must be classified as a noise peak or a signal peak. If a sample is greater than the current threshold value $threshold_{I1}$, then it is a peak candidate. In addition, it must have a distance of at least 200 ms from the previously detected peak: this value, referred to as min_rr_width, is the minimum latency time between adjacent R peaks due to physiological constraints. Otherwise, the fiducial mark is considered a noise peak. spk_I and npk_I parameters are updated accordingly:

$$spk_I = 0.125 * peak_I + 0.875 * spk_I \tag{2}$$

$$npk_I = 0.125 * peak_I + 0.875 * npk_I \tag{3}$$

If no R peak candidate is found in an interval of duration $1.66 * max\_rr\_width$ starting from the previous peak and ending with the current sample, the algorithm performs a search-back operation on this interval the interval using a lower threshold $threshold_{I2}$ that is empirically computed as:

$$threshold_{I2} = 0.5 * threshold_{I1}; \tag{4}$$

The original PT algorithm performs R peaks detection also on the output of the bandpass filter, introducing a set of variables with the same meaning (i.e., $peak_F$, spk_F, npk_F, $threshold_{F1}$, and $threshold_{F2}$). We verified experimentally that this step can be skipped without invalidating the detection quality. If a peak candidate occurs after the 200 ms refractory period but within 360 ms of the previous peak, the algorithm makes an additional check to determine if it is an abnormally prominent T wave. This decision is based on the mean slope of the waveform at that position, which must be greater than one-half that of the previous peak. Finally, the average distance between R peaks is computed as the mean of the eight most-recent RR intervals. The average heart rate can be used to refine the duration of the search back interval.

3 Results

This section provides an experimental evaluation of our system. We use GVSoC [20], an open-source simulator for PULP architectures, to implement and debug the algorithm. GVSoC can simulate a full platform, including multi-memory levels and multi I/O peripherals, and provides a good trade-off between simulation speed, timing accuracy, and completeness. The average energy consumption for the Vega platform has been derived by a post place-&-route simulation on the RTL. The metrics of interest for our performance analysis are *throughput* (computed as the number of input data samples over the total execution cycles), *energy efficiency* (operations performed in a second over power consumption), *total energy consumption* (in mJ), and *accuracy* of the detection rate (in percentage).

3.1 Implementation on the PULP platform

The AFE IC, described in Section 2.1, is connected to the ECG electrodes using 3 ECG surface sensors: two sensors are placed on the wrist, and the other one around the upper forearm of the subject, as a voltage reference. This setup allows sampling with an 18-bits resolution at 128 Hz. In this application, the signal is read 13 samples at a time using a FIFO. We apply the PT algorithm described in Section 2.2, implemented in C language, to support different (integer or floating-point) data types. The code supports both buffered and the data-streaming simulation with configurable parameters for the sampling frequency. In the case of buffered execution, the input buffer size is selected to contain at least 1.66 times an R-R interval, considering that the maximum physiological beats per minute are 60 or 80 (max 1.66 beats per second). The code includes buffers for the results of the intermediate filters. These buffers have the exact window size for the corresponding filter and are implemented as circular buffers to reduce memory consumption. Buffered execution can be used to execute the algorithm on pre-recorded ECG datasets, while the streaming variant is more efficient for real-time data acquisition.

In addition to the original PT design, we added a preliminary normalization step that removes the DC drift by subtracting the mean value and then dividing by a maximum absolute value. In the case of buffered execution, this value can be computed as the maximum value in the input buffer; otherwise, we can use the maximum value provided by the sensor as reported in the datasheet. The result is a signal normalized in the range [-1, 1], improving the numerical stability and precision of the next steps.

The filter coefficients are pre-computed using MATLAB and saved into the local memory to maximize the efficiency of the initial steps. To guarantee the minimum latency for streaming execution, we designed a step-by-step convolution function that is invoked for each new available value (i.e., a new input

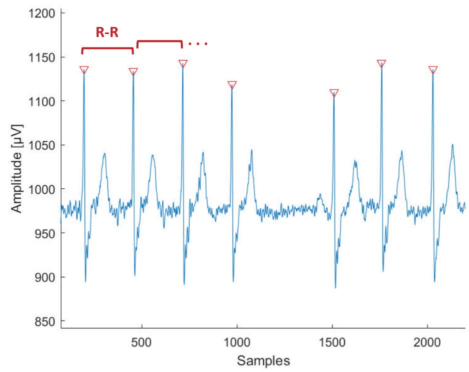

Fig. 3. Output result of the proposed R peaks detection and R-R intervals method from a segment of record 232 characterized by the supraventricular ectopic beats from the MIT-BIH database.

Table 1. Cycles for each sample, Instructions, Energy Efficiency, Throughput, and Time executing on the target platforms (average values on a 25 s time window).

	Pulp VEGA	**Cortex-M4 (processing pipeline)**
Cycles	2771	3154
Instructions	2204	3148
Energy efficiency [Gop/s/W]	34.7	3.8
Throughput [samples/ms]	61.35	26.63

sample or a value computed by the previous filter) and applied a linear convolution filter to the tail on the corresponding data buffer. As introduced in Section 2.2, we only consider the integrated signal for R peak detection. Finally, we apply the computation of the HR (beats per minute) from the RR average value. Figure 3 shows an example of the output result of the R peaks detection and the R-R intervals assessment extracted from a segment of the record 232 of the MIT-BIH Arrhythmia Dataset. Even though some fiducial points can be drifted forward or backward by one sample w.r.t. the exact peak positions, this effect does not affect the correct computation of the R-R distance.

3.2 Performance analysis and energy consumption

Table 1 reports the performance parameters executing the program (in streaming mode) on PULP (VEGA SoC) and Cortex-M4 (STM32NUCLEO-F401RE development board) introduced in Section 1. We deployed an alternative setup for these experiments where an additional STM32NUCLEO board is used in place of VEGA for the signal processing pipeline. In both cases, the energy consumption of the Nucleo board used for system initialization and debugging is not considered. The resulting values show that execution on the PULP platform is 2.3× faster than Cortex-M4.

Table 2 depicts the energy consumption (in mJ) of our algorithm executed on NUCLEOF401RE and Vega platforms compared to the state-of-the-art solution described by De Giovanni et al. [10], which executes on a PULP platform based on the Mr.Wolf architecture [21]. The operating frequency reported for Cortex-M4 is its maximum frequency. For VEGA, we are using an operating frequency

Table 2. Energy consumption of different SoA solutions for R peak detection (average values on a 25 s time window).

	Platform Architecture	ISA	Algorithm	Technology [nm]	Operating frequency [MHz]	Energy consumption [mJ]
De Giovanni et al. [10]	Pulp (Mr.Wolf)	RV32ICMF + Spec. Ext.	Adaptive slope	40	170	1.553
This work	Cortex-M4	ARMv7-M	PT	90	84	2.652
	Pulp (Vega)	RV32ICMX + Spec. Ext.	PT	22	170	**0.203**

Table 3. Comparison of R peaks accuracy.

	Acc [%]
Moreira et al. [14]	93.26
De Giovanni et al. [10]	97.90
Tekeste et al. [9]	99.37
Lu et al. [13]	99.41
This work	**99.53**

lower than the maximum to make a more fair comparison with state-of-the-art solutions. The energy consumption has been estimated using an average power consumption reported by the datasheets. Considering an execution time of 25 s, the average energy consumption of our system is almost 7× lower.

3.3 Algorithm Accuracy

Table 3 reports an accuracy comparison between our solution and other works. In the worst case, our algorithm reaches 99.53% on the high-intensity physical exercise dataset [11]. To evaluate the accuracy, we used the MATLAB *findpeaks* function as a golden reference, which returns the local maxima. It is extremely accurate, but it has two main flaws. First, it is computed intensive, which is highly detrimental to its adoption in the ultra-low-power embedded domain. Second, it cannot be adapted to a streaming context, so its adoption would increase the latency of the results. We computed the *accuracy* as follows:

$$RMSD = \sqrt{\frac{\sum_{i=1}^{n} (x_{G,i} - x_{PT,i})^2}{n}}; \tag{5}$$

$$Acc = 100 - (\frac{RMSD}{x_{max} - x_{min}}); \tag{6}$$

where x_G and x_{PT} are the RR intervals (in samples) computed using the golden model or the proposed method, respectively. The parameter n is the number of detected RR intervals, and x_{max} and x_{min} are the maximum and minimum in the set of RR interval values.

Figure 4 depicts the accuracy of the code tested on four different datasets. The datasets we consider are Normal Sinus Rhythm (NSR) and Atrial Flutter (AFL). They are both from the MIT-BIH Arrhythmia database, sampled at 360 Hz [18]. The third is the ECG signal acquired in real-time (RT) from our signal acquisition system (described in Section 2.1). Finally, the signal on high-intensity exercise (HIE), sampled at 250 Hz [11]. The figure shows the higher value in NSR, for which we achieve 99.95%. In the case of tachyarrhythmia, called atrial flutter

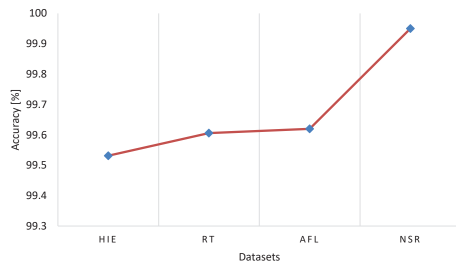

Fig. 4. Accuracy evaluation on four different datasets: High Intensity Exercise (HIE) [11], acquired ECG signal in real-time (RT) with the proposed system design, Atrial Flutter (AFL) [18], and Normal Synus Rhythm (NSR) [18].

(AFL), the accuracy is 99.62%. In the RT, we obtain an accuracy of 99.61%. In HIE, where the beats change suddenly, we assess the lower value of 99.53%.

4 Conclusion

This work presents the design and implementation of a heart-rate detection system leveraging the PT algorithm on low-power MCUs. We consider two alternative platforms, a commercial MCU based on ARM Cortex-M4 and an ultra-low-power solution based on RISC-V, namely the Vega SoC. Experimental results show that our approach is lightweight design and executes in a few thousand cycles. This system provides a lifetime battery of 81 hours with a 100 mAh battery, achieving an accuracy comparable to the state-of-the-art solutions and an energy efficiency that is one order of magnitude better. This work does not aim at classifying specific health problems but rather to detect HR in real-time at high reliability and energy efficiency. In future work, we will add machine learning algorithms (e.g., kNN, SVM, or CNN) to the system pipeline with the aim of detecting anomalies in the HR variability, such as arrhythmia or stress conditions. Moreover, we will design a parallel version of the code using the programmable parallel accelerator available on VEGA to further improve performance and energy efficiency compared to commercial alternatives.

Acknowledgements The work was partially funded by the EU project The European PILOT (g.a. 101034126)

References

[1] Pratyush Rai et al. "Nano- Bio- Textile Sensors with Mobile Wireless Platform for Wearable Health Monitoring of Neurological and Cardiovascular Disorders". In: *Journal of The Electrochemical Society* 161.2 (2013), pp. 3116–3150.

[2] Sangil Lee et al. "Integration of transparent supercapacitors and electrodes using nanostructured metallic glass films for wirelessly rechargeable, skin heat patches". In: *Nano letters* 20.7 (2020), pp. 4872–4881.

[3] Farnaz Forooghifar et al. "A self-aware epilepsy monitoring system for real-time epileptic seizure detection". In: *Mobile Networks and Applications* (2019), pp. 1–14.

[4] Simone Benatti et al. “A Versatile Embedded Platform for EMG Acquisition and Gesture Recognition”. In: *IEEE Transactions on Biomedical Circuits and Systems* 9.5 (2015), pp. 620–630.
[5] Michele Magno et al. “Autonomous smartwatch with flexible sensors for accurate and continuous mapping of skin temperature”. In: *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 337–340.
[6] Philipp Schönle et al. “A Multi-Sensor and Parallel Processing SoC for Miniaturized Medical Instrumentation”. In: *IEEE Journal of Solid-State Circuits* 53.7 (2018), pp. 2076–2087.
[7] Jeong Seon Park, Sang Woong Lee, and Unsang Park. “R Peak Detection Method Using Wavelet Transform and Modified Shannon Energy Envelope”. In: *Journal of Healthcare Engineering* 2017 (July 2017), pp. 1–14.
[8] Arturo Martinez Rodrigo, Raùl Alcaraz, and José Rieta. “Application of the phasor transform for automatic delineation of single-lead ECG fiducial points”. In: *Physiological measurement* 31 (Nov. 2010), pp. 1467–1485.
[9] Temesghen Tekeste et al. “Ultra-Low Power QRS Detection and ECG Compression Architecture for IoT Healthcare Devices”. In: *IEEE Trans. on Circuits and Systems I: Regular Papers* 66.2 (2019), pp. 669–679.
[10] Elisabetta De Giovanni et al. *Adaptive R-Peak Detection on Wearable ECG Sensors for High-Intensity Exercise.* 2021.
[11] Elisabetta De Giovanni et al. *ECG in High Intensity Exercise Dataset.* Zenodo, Nov. 2021.
[12] Jiapu Pan and Willis J. Tompkins. “A Real-Time QRS Detection Algorithm”. In: *IEEE Transactions on Biomedical Engineering* BME-32.3 (1985), pp. 230–236.
[13] Xuanyu Lu, Maolin Pan, and Yang Yu. “QRS Detection Based on Improved Adaptive Threshold”. In: *Journal of Healthcare Engineering* 2018 (Mar. 2018), pp. 1–8.
[14] João Leite et al. “Online Heartbeat Classification Using Low Cost Algorithms”. In: Mar. 2020.
[15] Paulus Kirchhof et al. “2016 ESC Guidelines for the management of atrial fibrillation developed in collaboration with EACTS”. In: *European Heart Journal* 37.38 (Aug. 2016), pp. 2893–2962.
[16] Davide Rossi et al. “PULP: A parallel ultra low power platform for next generation IoT applications”. In: *2015 IEEE Hot Chips 27 Symposium (HCS).* 2015, pp. 1–39.
[17] Davide Rossi et al. “4.4 A 1.3TOPS/W @ 32GOPS Fully Integrated 10-Core SoC for IoT End-Nodes with 1.7Î¼W Cognitive Wake-Up From MRAM-Based State-Retentive Sleep Mode”. In: *2021 IEEE International Solid-State Circuits Conference (ISSCC).* Vol. 64. 2021, pp. 60–62.
[18] G.B. Moody and R.G. Mark. “The impact of the MIT-BIH Arrhythmia Database”. In: *IEEE Engineering in Medicine and Biology Magazine* 20.3 (2001), pp. 45–50.
[19] *Maxim Datasheet:* `https://datasheets.maximintegrated.com/en/ds/MAX30003.pdf`. Accessed: 2022-05-10.
[20] Nazareno Bruschi et al. “GVSoC: A Highly Configurable, Fast and Accurate Full-Platform Simulator for RISC-V based IoT Processors”. In: *2021 IEEE 39th Int. Conference on Computer Design (ICCD)*, pp. 409–416.
[21] Antonio Pullini et al. “Mr. Wolf: An energy-precision scalable parallel ultra low power SoC for IoT edge processing”. In: *IEEE Journal of Solid-State Circuits* 54.7 (2019), pp. 1970–1981.