Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 159 (2019) 1938–1946

Procedia Computer Science

www.elsevier.com/locate/procedia

23rd International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

ECG sensor for detection of driver's drowsiness

Markus Gromer[a], David Salb[a], Thomas Walzer[a], Natividad Martínez Madrid[a,b,*], Ralf Seepold[b,c,**]

[a]*Reutlingen University, Alteburgstr. 150, 72762 Reutlingen, Germany*
[b]*I.M. Sechenov First Moscow State Medical University, 2-4, Bolshaya Pirogovskaya st., 119435 Moscow, Russian Federation*
[c]*HTWG Konstanz, Alfred-Wachtel-Str. 8, 78462 Konstanz, Germany*

Abstract

Fatigue and drowsiness are responsible for a significant percentage of road traffic accidents. There are several approaches to monitor the driver's drowsiness, ranging from the driver's steering behavior to the analysis of the driver, e.g., eye tracking, blinking, yawning, or electrocardiogram (ECG). This paper describes the development of a low-cost ECG sensor to derive heart rate variability (HRV) data for drowsiness detection. The work includes hardware and software design. The hardware was implemented on a printed circuit board (PCB) designed so that the board can be used as an extension shield for an Arduino. The PCB contains a double, inverted ECG channel including low-pass filtering and provides two analog outputs to the Arduino, which combines them and performs the analog-to-digital conversion. The digital ECG signal is transferred to an NVidia embedded PC where the processing takes place, including QRS-complex, heart rate, and HRV detection as well as visualization features. The resulting compact sensor provides good results in the extraction of the main ECG parameters. The sensor is being used in a larger frame, where facial-recognition-based drowsiness detection is combined with ECG-based detection to improve the recognition rate under unfavorable light or occlusion conditions.

Keywords: ECG ; Drowsiness ; Automotive ; Sleep ; Biomedical Signal Capturing

1. Introduction

Statistics gathered by the German Automobile Club (ADAC) have shown that approximately 26% of drivers have already fallen asleep at the driving wheel. However, the number of unreported car accidents caused by fatigue is estimated to be much higher. This is because accident statistics refer to the driver's statements when determining the cause of the accident. To avoid (partial) guilt, it can often happen that the driver states another accident cause [1].

* Natividad Martínez Madrid. Tel.: +49-7121-271-4014 ; fax: +49-7121-271-94014 ; ORCID: 0000-0003-1965-9481.
** Ralf Seepold. Tel.: +49-7531-206-633 ; fax: +49-7531-206-87633 ; ORCID: 0000-0002-5192-2537.
E-mail address: natividad.martinez@reutlingen-university.de, ralf.seepold@htwg-konstanz.de

10.1016/j.procs.2019.09.366

The European Society for Sleep Research dealt with the frequency and causes of sleep-related accidents. The study was carried out in 19 European countries with over 12,700 road user surveys: more than 42% of the respondents who originated an accident by falling asleep while driving stated that the night before, they had slept badly [2, 3, 4]. Mortality from traffic accidents is a global public health problem that leaves human losses estimated at 1.25 million each year [5].

Relevant parameters for detecting drowsiness are the blink rate, yaw rate, respiration, and pulse rate. Literature research reflects two main approaches for drowsiness or fatigue detection in the automotive sector. One of the approaches is based on image processing [6, 7, 8] and the other is based on electrocardiogram (ECG) and pulse analysis, especially using the heart rate variability (HRV) [9, 10, 11, 12]. There are other embedded on-board systems to detect driver drowsiness (e.g., [13, 14]) but most of them need more powerful and expensive hardware, particularly systems using a separate camera for eye-movement detection (e.g., [15]). A good overview of the signals relevant to the detection of drowsiness (such as EOC, EEC, ECG) can be obtained from [16].

Drowsiness recognition through facial image analysis gives quite good results under the right conditions, but if relevant parts of the face are covered up (e.g., eyes covered by sunglasses), the recognition rate decreases. A combination of image processing and heart rate analysis would improve the results in all circumstances. However, current devices to measure the ECG or at least the heart rate variability do not synchronize in real-time with their servers or backup devices. Therefore, they could only be used for a posteriori analysis of the cause of an accident, but to avoid the accident by warning the driver.

The work presented in this paper was developed in the broader frame of a research project whose goal is to design a low-cost, mobile system for driver's drowsiness detection combining both facial and ECG analysis.

This paper describes the development of both hardware and software components of the sensor. The hardware consists of a PCB for amplification and filtering, designed as a shield for the Arduino Uno. Different prototypes of the amplifier were developed and tested. The combination of both PCB and Arduino provides an ECG signal to the embedded PC. The software in the embedded PC is responsible for the following tasks: more signal filtering, QRS extraction, HRV calculation, and output generation. The output is plotted for quality control, and further elements of the project use it for drowsiness detection. Additionally, software was written for the Arduino to generate a simulated ECG signal with noise an artifacts. This was used for software testing purposes during the initial phase and allowed to parallelize the development of hardware and software.

This device will be later on combined with the face recognition part; therefore, a modular and extensible concept was pursued from the beginning. The presented system was tested in the Driving Simulator of the IoT Laboratory at Reutlingen University, where various current and new functions of today's motor vehicles are developed and tested.

The rest of the paper is organized as follows. In section 2, the physiological working principles of an electrocardiogram are introduced. Section 3 describes the design and evaluation of the hardware developed for the ECG device. Besides the hardware, the device also needs a software design that is presented in section 4. It shows the general software architecture and highlights the functionality for the recognition of the QRS-complex of the ECG signal and the subsequent calculation of the heart rate and the heart rate variability. Section 5 concludes with a summary and outlook of future work.

2. Physiological principles

An electrical stimulus triggers the heart muscle contraction. It is generated by the heart in its own pacemaker centers and carried from one heart muscle cell to the next. This so-called conduction of stimulation or excitation can be measured. These electrical stimuli can be tapped on the body. According to the classical Einthoven derivation, three electrodes are used to measure the voltages, forming a triangle around the heart. Traditionally, the electrodes are attached to both arms and the left leg. However, it makes sense to connect the electrodes close to the heart with good electrical contact. Therefore, positions in the shoulder or chest area as well as below the left ribs are typically chosen. The position shown in Fig. 1(a) represents the direction of the electrical stimulus and thus also the orientation of the heart muscle. The figure shows that the largest tension along the heart axis can be measured between the right arm and the left leg.

If the measured voltages are visualized over time, an electrocardiogram (ECG) is produced. In Fig. 1(b) an ECG is shown in a temporal context. The most striking elevation, the so-called QRS complex, marks the propagation of

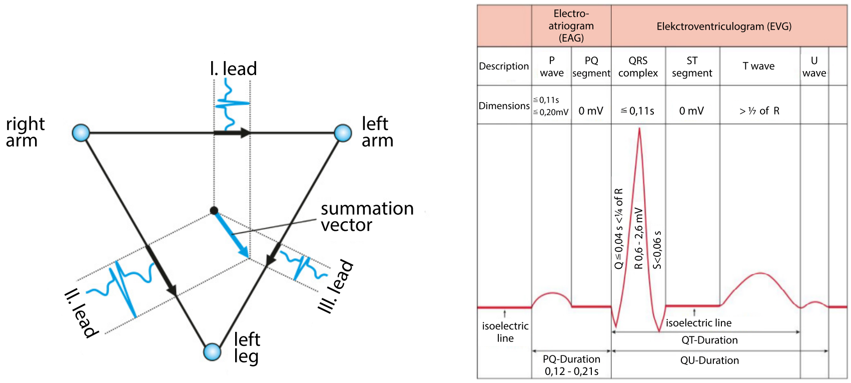


Fig. 1. (a) Different ECG-leads according to Einthoven [17]; (b) Cyclic periods of an electrocardiogram [18]

excitation in the heart chambers. The induced voltage varies from person to person and lies between 0.6 - 2.6 mV. This weak signal must be amplified many times over to be used. The QRS complex varies only slightly over time and should normally last less than 0.11 seconds [18].

3. Hardware development

The hardware was developed iteratively. Several solutions were designed and tested, starting with developments on a breadboard through a first soldered design to finished board design. In this section, the design steps and the central development of the PCB layout are described and the results are presented.

The complete design consists of an amplifying and filtering board for the ECG signal, an analog-to-digital conversion (ADC) and further software to process the resulting signal in the embedded PC located in the car or the driving simulator. An Arduino Uno was used as ADC. It offers the possibility to read up to six signals in parallel via several analog inputs. One channel was sufficient for the first attempts at a solution. A second channel was added later in the project. The analog signal is read in with the Arduino, preprocessed and then sent to the Nvidia Embedded PC via a serial interface. The software of the Arduinos is kept very simple. The actual primary evaluation of the signal takes place on the embedded PC and is described in more detail in section 4.

3.1. *Basic amplifier design*

Adhesive electrodes are used to record the ECG signal. The difference of voltages from the arm electrodes is amplified so that an analog ECG signal is obtained at the output. The designed amplifier circuit uses the INA128 instrumentation amplifier [1]. The INA128 was used here as a differential amplifier of the two electrodes of the right and left arm. A right leg (RL) drive noise-reduction technique was implemented, and the resulting voltage is applied to the person's right leg. The circuit acts as a feedback loop to drive the common-mode noise on the patient to a low level. The output signal is connected to the analog input of the Arduino.

To optimize the signal, a potentiometer was installed instead of the two R_g resistors of 2.8k Ohm. In this way, the amplification of the circuit can be adjusted variable, and the optimal setting for a well recognizable ECG can be found. This was very helpful during circuit development because the gain setting can change over and over again. The resulting signal of the first breadboard prototype is shown in figure 2. The ECG signal can already be appreciated, but it still has too much noise and artifacts. A well measurable ECG depends on many factors. Some examples that turned out during the development were: worn clothes; sitting posture; power supply via battery or an adapter; ambient noise; movement and breathing; type of connector that was used; and persons who were near the cables and the circuit design.

[1] https://www.ti.com/lit/ds/symlink/ina128.pdf

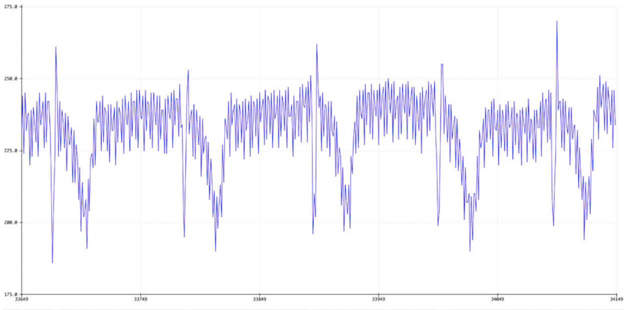


Fig. 2. Resulting signal of the breadboard setup

3.2. Printed Circuit Board (PCB) layout

A first soldered prototype following the breadboard design resulted in considerable improvement of the ECG signal. On this basis, it was decided to add a channel offering higher bandwidth and reverse polarity protection. Incorrect connection of the electrodes would thus play no role. This lead to the decision to produce a separate circuit board. The board should have two measuring channels of the ECG, avoid electromagnetic interference and noise, and further improve the signal by correct filtering. This section describes the development of this board in more detail.

The circuit design is an extension of the previous developments on a breadboard and soldered prototype. The layout was designed so that the board can be used as an extension shield for an Arduino. This offers an overall compact design. Furthermore, the circuit was designed with different adjustment possibilities despite a non-changeable circuit board. For example, it is possible to choose between basic voltages of 3 Volt or 5 Volt, to connect the shielding of the electrodes or to draw different reference outputs to the ground.

A second measuring channel makes the results of the ECG more robust, helps to display negative voltages of the ECG signal and increases the measuring range. The analog input of the Arduino can only measure positive voltages. If the voltage of the ECG signal moves into the negative range (which can happen if the signal fluctuates around the zero line), then it cannot be detected by the Arduino. The signal is cut off at this point. To avoid this, a second measuring channel was needed to cover the negative range of the signal. The two measuring channels have an identical amplifier structure on the hardware side, but different electrode inputs and output. The design is such that the electrode for the right arm is placed on the positive input of the amplifier circuit of one measuring channel and the electrode for the left arm is placed on the negative input of the amplifier circuit of that measuring channel. On the amplifier circuit of the second measuring channel, the two electrodes are switched to the inputs in reverse. The right leg remains connected in the same way for both channels. The outputs of both channels are connected to separate analog inputs of the Arduino. With this design, the positive signal is measured from one measuring channel. As soon as the signal moves below the zero line, the signal is obtained from the second input line. The total signal can be assembled from the signals of both inputs by simply subtracting the two channels.

In the design of this circuit board, there are several possibilities to suppress interferences to the signal. At each of the two channels, a first-order RC low-pass filter consisting of a resistor and a capacitor is provided at the output. The RC low pass filter has a 5k Ohm resistor and a 1μ Farad capacitor, with a cut-off frequency of about 31 Hz. In this way, the noise of a power supply can already be filtered. Furthermore, two capacitors of 100 and 10 Nanofarad are provided for the suppression of the positive voltage source. As a further possibility, the manufacturer of the INA 128 instrumentation amplifier recommends installing a bypass capacitance between the positive and negative voltage supply and ground for interference suppression. Supported values are between 0.1μ farad and 1μ farad. This possibility is also included in the board design.

3.2.1. Results

The measured signal has become much less noisy and much more robust due to the improvements mentioned above. Figure 3(a) shows the assembled board mounted on an Arduino. The first derivative of the resulting signal is

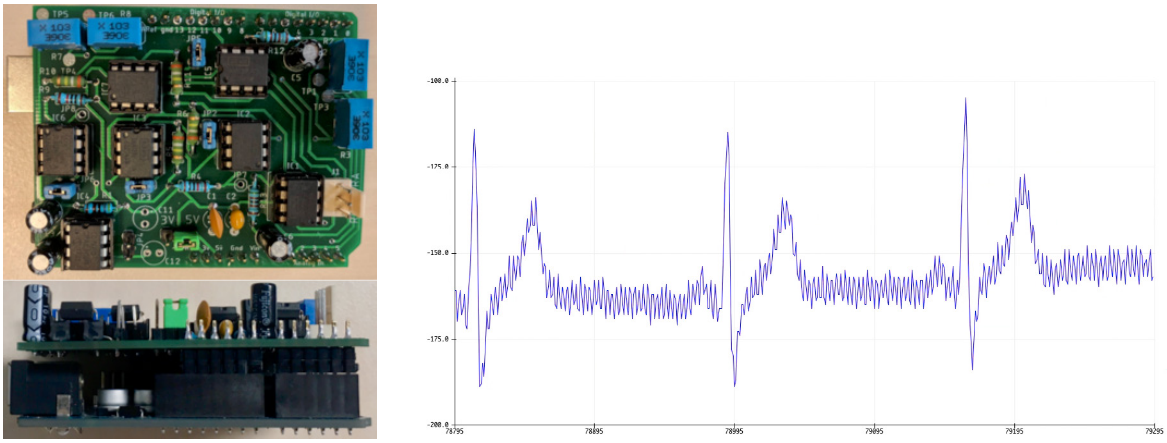


Fig. 3. (a) Populated board on an Arduino Uno; (b) Resulting signal (1st lead) on the serial interface

shown in Figure 3(b). The noise to be detected now still on the signal is not further of importance for the evaluation of the ECG signal, since the characteristics of an ECG differ clearly from it.

4. Software development

The main objectives of the software to be developed were the detection of the QRS, the derivation of the heart rate, and the computation of the HRV. One function was developed for each task. Also, a plotting frontend was implemented for development purposes. It was used for customizing the necessary filters during the application and checking the plausibility of the QRS complex detection. Furthermore, it serves as a convenient debugging tool during the entire development work. The implementation task was divided into three modules, serving the activities of connection, processing, and output. In this way, signal processing could be decoupled from the serial link, allowing asynchronous processing and thus filtering of the signal. In the last part, a continuous output was realized using a buffer implementation and the dynamic plotting frontend. Based on the data shown in Fig. 4 an overview of all system components and their interfaces are given in the following before the system sequences are documented in detail. It shows the primary process containing the connection and offset shift functionality. Filtering, QRS detection, BPM calculation, and the computation of HRV are gathered in the second module. The third module is dedicated to the output of the processed data, for example, with the help of a plotter. Fig. 5 shows the architecture in more detail.

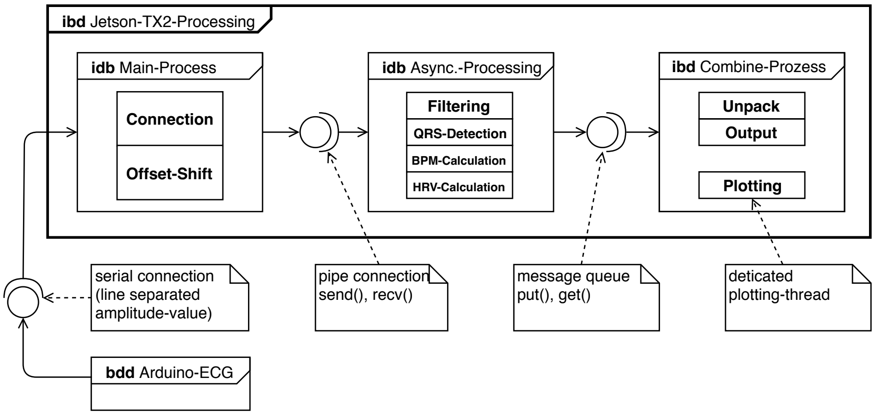


Fig. 4. Process overview

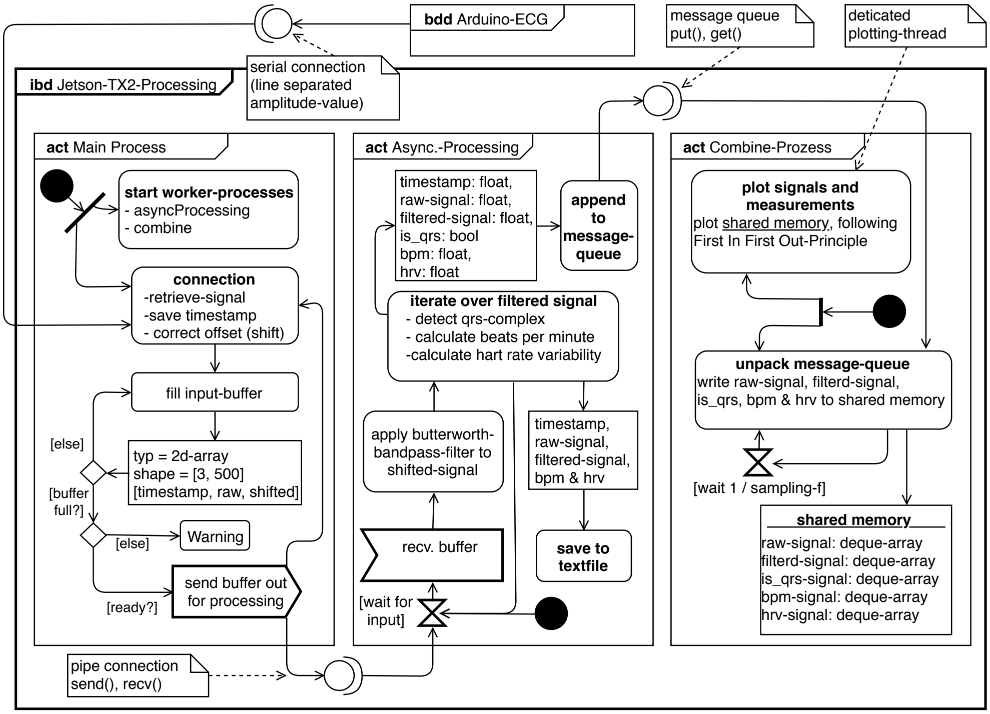


Fig. 5. Entire system process

4.1. ECG extraction

Before the ECG signal can be analyzed for QRS complex detection, it should be filtered. It is necessary to eliminate various sources of interference from the analog ECG signal. For example, the 50Hz signal typically emitted from electrical lines. The filtered signal can then be analyzed for QRS complexes using adaptive threshold analysis and various smaller filters to calculate the heart rate and variance. These computation-intensive operations are subject to fluctuating load behavior. Therefore, it is necessary to decouple this processing from time-critical processes, such as the reading and output of the signal. Thus a stable, continuous, and low-latency signal analysis and forwarding should be realized.

With the described separation of the processes, a higher sampling rate can be achieved when reading in the measured signal. A so-called dual buffer ensures uninterrupted processing of the incoming data stream. The connection side is responsible for continuously filling the input buffer while the content of the previous buffer is already being processed and examined. In the following, this fact is referred to as asynchronous processing. To form a continuous output signal from the signal, which is now divided into blocks, each value is passed to a final output buffer during processing. The output routine runs independently and in parallel to the rest of the application. Finally, the values are continuously plotted. Since, for example, the QRS complex detection or heart rate calculation is time-dependent, this strategy requires a unique timestamp for each input value. Depending on how the buffer size is selected, the signal output is, of course, time-shifted. However, this latency should be constant and can be limited to a few seconds, and this can be regarded as non-critical.

4.2. QRS identification

To recognize the threshold for the QRS complex, it helps to know whether the current measured value is a peak or not. The last and the following measured values are compared with the current one. If both have a lower value, it is a peak point. Asynchronous buffer processing offers the advantage that the following values in the buffer are already known. This means that the next value in the array can be called for this comparison. This implementation is realized by appending with each iteration, not the current but the following value to a ring buffer with a size of three. For the acquisition of the QRS complex, the peak point information is used to examine the current measured value for threshold values. Therefore, each incoming value is first appended to the so-called `qrsBuffer` This is a ring buffer containing 400 values, which are checked for its maximum in each iteration. This value can already serve as a benchmark for potential QRS complexes. However, to make the system more robust against outliers, this maximum value is again subjected to a median filter of the last ten maxima. The signal is already zero line corrected at the signal input. Thus, the maximum value can be used as a direct amplitude measure of the QRS complexes. Since the maximum value of the signal is continuously measured and updated, this measure behaves adaptability to the measured signal. To detect a QRS complex, it is only necessary to check whether the current measured value is a peak point and whether it is within a defined threshold value range. This range is set to 70 - 120% of the current maximum value. To prevent the same heartbeat from being counted multiple times, a minimum duration of 0.3 seconds is set between two possible QRS complexes. This limits the differentiation of the heartbeats and thus, the maximum detectable heart rate to 200 bpm ($60s * \frac{1}{0{,}3s} = 200$). If all these conditions apply, a QRS complex is detected, and the heart rate can be calculated.

4.3. HRV computation

If a QRS complex is detected, the duration since the last QRS complex is used to calculate the heart rate ($60 * \frac{1}{duration}$). The value is appended to a 5-value ring buffer to obtain the heart rate from the median of these last five values. This serves to bridge erroneous measurements or undetected QRS complexes. Heart Rate Variability (HRV) is calculated from the standard deviation of the last twenty RR intervals[2] and this is understood as a measure of fatigue, whereby a lower value means a calm, constant heartbeat and thus a higher probability for fatigue is assumed. If more than two seconds elapse without a detected QRS complex, a signal loss or disturbance is, and the HR and HRV values are set to `NaN` (Not a Number). With the implementation of this two-second waiting time, all heart rates above 30 bpm can still be detected.

4.4. Value passing

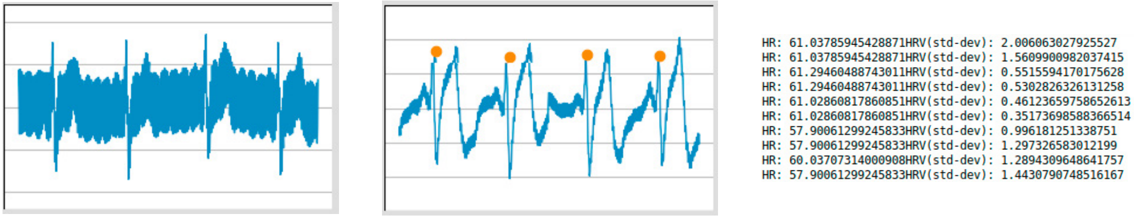
Finally, all data is appended to a cross-process queue [3]. The attached tuple consists of the time stamp, the raw signal, the filtered signal, the HR, the HRV and a boolean value indicating whether the current reading is a QRS complex or not. HR and HRV are sent continuously, although these values are of course only updated with the detection of a QRS complex. The boolean `is_qrs`-value serves as a mask for the detected QRS complexes and is used in this context for subsequent plotting. Besides, this information can be used in subsequent processing routines to recognize the update of HR and HRV values.

4.5. Output visualization

Since the incoming data stream is processed in blocks, the measured or generated values are not transferred to the queue at a constant speed. To compensate for this fluctuation, but at the same time to ensure that the buffered values do not continue to grow or be used up, the data must be taken at the average input speed. This is done by checking the current length of the queue. If it exceeds a specific limit value, the withdrawal speed will be slightly increased. Since the reading frequency should be constant and usually only fluctuates in the tenth-millisecond range, this is a simple solution to prevent buffer complications. Starting from this point, new algorithms can be implemented to detect fatigue. To realize the plotting, the extracted values are written into ring buffers which can be accessed by a dedicated plotting thread. These plots store data continuously. Fig. 6 shows the generated window. The raw signal, filtered signal (with a QRS annotation) as well as HR and HRV values are continuously displayed. An average repetition rate of 30-60 fps is achieved.

[2] time between two heartbeats

[3] https://docs.python.org/3.6/library/multiprocessing.html#exchanging-objects-between-processes



```
HR: 61.03785945428871HRV(std-dev): 2.006063027925527
HR: 61.03785945428871HRV(std-dev): 1.5609900982037415
HR: 61.29460488743011HRV(std-dev): 0.5515594170175628
HR: 61.29460488743011HRV(std-dev): 0.5302826326131258
HR: 61.02860817860851HRV(std-dev): 0.46123659758652613
HR: 61.02860817860851HRV(std-dev): 0.35173698588366514
HR: 57.90061299245833HRV(std-dev): 0.996181251338751
HR: 57.90061299245833HRV(std-dev): 1.297326583012199
HR: 60.03707314000908HRV(std-dev): 1.2894309648641757
HR: 57.90061299245833HRV(std-dev): 1.4430790748516167
```

Fig. 6. Plotting of raw and filtered signals, and HR / HRV values

4.6. Additional generation of simulated input

Since the processing hardware was developed in parallel, it was not always possible to test with real ECG data. For this reason, a simulator was additionally developed that runs on the same Arduino microcontroller that is also responsible for the later AD conversion and signal transmission. The simulator consists only of an adjustable periodic signal. To simulate the noise behavior, different sinus waves with different wavelengths are superimposed. Random, erratic outliers can also be activated as an additional source of interference. Fig. 7 shows the generated signal. It is also important that the heart rate can be adjusted to test the limits and accuracy of the system. Please note that the results shown in Fig. 6 correspong to a real ECG-input, not a simulated one.

5. Summary and outlook

This paper describes the development of a mobile solution for the measurement of a driver's drowsiness. The development captures the ECG with a custom PCB design. The PCB can be attached as a plug-in to a standard Arduino board for fast prototyping. The signal capturing was improved to be reliable and to be applicable in an automotive environment. However, an incorrect connection of the electrodes still leads to an ECG detection. The signal capturing is now based on a two-channel solution. The signal processing was carried out with the help of an algorithm running as a modular piece of software. It is used to detect the pattern of a QRS complex and to implement the HR and HRV, derived from the QRS complex. As a result, it is possible to make a statement about the fatigue of the driver. In the case of sensor fusion, this can contribute to a quantified statement of the driver's condition. However, the current implementation could be improved by the following aspects: First, for galvanic isolation and to increase freedom of movement during operation, it is advisable to replace the serial USB connection with another interface, such as Bluetooth. Second, change the PCB layout in order to replace the components with so-called source mounted devices by SMD components. This change would lead to an even more compact design for a smaller Arduino series or other micro-controller boards. Finally, investigate whether an ordinary wrist band fitness tracker could replace the ECG. In this case, it needs to be a device that provides access to the raw data.

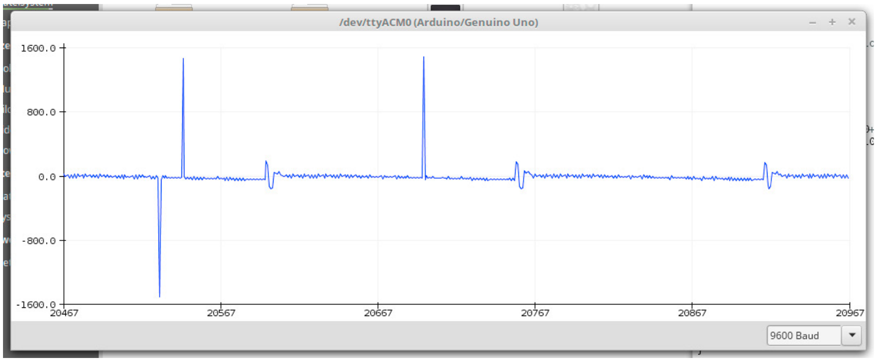


Fig. 7. Simulated ECG-signal with artificial noise peaks

Acknowledgments

This research was partially funded by the EU Interreg V-Program ”Alpenrhein-Bodensee-Hochrhein”: Project ”IBH Living Lab Active and Assisted Living”, grants ABH040, ABH041 and ABH066. Part of this report was funded by the Federal Ministry of Education and Research under the number 01DH17056. The author is responsible for the content of this publication.

References

[1] J. Zulley, R. Popp, Müdigkeit im Straßenverkehr, ADAC e.V. München, Artikelnummer 2831141, München, 2012.

[2] M. Goncalves, R. Amici, R. Lucas, T. Akerstedt, F. Cirignotta, J. Horne, D. Leger, W. T. McNicholas, M. Partinen, J. Teran-Santos, P. Peigneux, L. Grote, C. National Representatives as Study, Sleepiness at the wheel across europe: a survey of 19 countries, J Sleep Res 24 (3) (2015) 242–53. doi:10.1111/jsr.12267.

[3] M. N. Rastgoo, B. Nakisa, A. Rakotonirainy, V. Chandran, D. Tjondronegoro, A critical review of proactive detection of driver stress levels based on multimodal measurements, ACM Computing Surveys 51 (5) (2018) 1–35. doi:10.1145/3186585.

[4] D. Riemann, E. Baum, S. Cohrs, T. Crönlein, G. Hajak, E. Hertenstein, P. Klose, J. Langhorst, G. Mayer, C. Nissen, T. Pollmächer, S. Rabstein, A. Schlarb, H. Sitter, H.-G. Weeß, T. Wetter, K. Spiegelhalder, S3-leitlinie nicht erholsamer schlaf/schlafstörungen, Somnologie 21 (1) (2017) 2–44. doi:10.1007/s11818-016-0097-x.

[5] W. H. Organization (Ed.), Global Status Report on Road Safety 2018, World Health Organization, 2018.

[6] C. Anitha, M. Venkatesha, B. Suryanarayana Adiga, A two fold expert system for yawning detection, Procedia Computer Science 92 (2016) 63–71. doi:10.1016/j.procs.2016.07.324.

[7] S. Abtahi, B. Hariri, S. Shirmohammadi, Driver drowsiness monitoring based on yawning detection, in: 2011 IEEE International Instrumentation and Measurement Technology Conference, 2011, pp. 1–4. doi:10.1109/IMTC.2011.5944101.

[8] L. I. Stephens, N. A. Payne, S. A. Skaanvik, D. Polcari, M. Geissler, J. Mauzeroll, Evaluating the use of edge detection in extracting feature size from scanning electrochemical microscopy images, Anal Chem 91 (6) (2019) 3944–3950. doi:10.1021/acs.analchem.8b05011.

[9] J. Vicente, P. Laguna, A. Bartra, R. Bailón, Drowsiness detection using heart rate variability, Medical & Biological Engineering & Computing 54 (6) (2016) 927–937. doi:10.1007/s11517-015-1448-7.

[10] K. T. Chui, K. F. Tsang, H. R. Chi, C. K. Wu, B. W. Ling, Electrocardiogram based classifier for driver drowsiness detection, in: 2015 IEEE 13th International Conference on Industrial Informatics (INDIN), Berlin Heidelberg New York, 2015, pp. 600–603. doi:10.1109/INDIN.2015.7281802.

[11] W. D. Scherz, J. A. Ortega, R. Seepold, N. Martínez Madrid, Stress Determent via QRS Complex Detection, Analysis and Pre-processing, Mobile Networks for Biometric Data Analysis 392 (2016) 225–234. doi:10.1007/978-3-319-39700-9_18.

[12] W. D. Scherz, D. Fritz, O. R. Velicu, R. Seepold, N. M. Madrid, Heart rate spectrum analysis for sleep quality detection, EURASIP Journal on Embedded Systems 2017 (1). doi:10.1186/s13639-017-0072-z.

[13] E. Temitayo, Y. Thomas, k. Ayodele, A. Ogunseye, Implementation of an on-board embedded system for monitoring drowsiness in automobile drivers, International Journal of Technology 9 (4) (2018) 819–827. doi:10.14716/ijtech.v9i4.1691.

[14] K. Kurosawa, N. Takezawa, T. Sano, S. Miyamoto, M. Yasushi, H. Hashimoto, Drowsiness prediction system for vehicle using capacity coupled electrode type non-invasive ecg measurement, in: 2017 IEEE/SICE International Symposium on System Integration (SII), 2017, pp. 306–311. doi:10.1109/SII.2017.8279230.

[15] M. Tayab Khan, H. Anwar, F. Ullah, A. Ur Rehman, R. Ullah, A. Iqbal, B.-H. Lee, K. S. Kwak, Smart real-time video surveillance platform for drowsiness detection based on eyelid closure, Wireless Communications and Mobile Computing 2019 (2019) 1–9. doi:10.1155/2019/2036818.

[16] M. Holub, M. Srutova, L. Lhotska, Microsleeps and their detection from biological signals, in: A. Mashko (Ed.), Driver-Car Interaction & Safety Conference 2016, Vol. 12 of Acta Polytechnica CTU Proceedings, Czech Technical Univ Prague, Prague 6, 2017, pp. 32–37. doi:10.14311/app.2017.12.0032.

[17] C. Cornelia, C. Wolfgang, Herz-Kreislauf- und Gefäßsystem. In: Humanbiologie kompakt, 2009th Edition, Springer-Verlag, Berlin Heidelberg New York, 2010. doi:10.1007/978-3-8274-2239-2.

[18] R. Klinge, Das Elektrokardiogramm - Leitfaden für Ausbildung und Praxis, 10th Edition, Georg Thieme Verlag, Stuttgart, 2015.