Portable Device for Continuous Sensing with Rapidly Pulsed LEDs – Part 1: Rapid On-the-fly Processing of Large Data Streams using an Open Source Microcontroller with Field Programmable Gate Array

Portable Device for Continuous Sensing with Rapidly Pulsed LEDs – Part1: Rapid On-the-fly Processing of Large Data Streams using an Open Source Microcontroller with Field Programmable Gate Array

Ansara Noori, Parvez Mahbub, John S. Parry, John Davis, Arko Lucieer, Mirek Macka

PII: S0263-2241(19)30460-9
DOI: https://doi.org/10.1016/j.measurement.2019.05.034
Reference: MEASUR 6643

To appear in: *Measurement*

Received Date: 7 January 2019
Revised Date: 26 April 2019
Accepted Date: 4 May 2019

ELSEVIER

Measurement

journal homepage: www.elsevier.com

Portable Device for Continuous Sensing with Rapidly Pulsed LEDs – Part 1: Rapid On-the-fly Processing of Large Data Streams using an Open Source Microcontroller with Field Programmable Gate Array

Ansara Noori[a], Parvez Mahbub[a, b], John S. Parry[c], John Davis[c], Arko Lucieer[d], Mirek Macka[a, e, f]*

[a] *Australia Centre for Research on Separation Science (ACROSS) and School of Physical Sciences- Chemistry, University of Tasmania, Tasmania, Australia*

[b] *Institute for Sustainable Industries and Livable Cities, Victoria University, Footscray Park Campus, Melbourne, Victoria 3011, Australia*

[c] *Central Science Laboratory, University of Tasmania, Private Bag 74, Hobart 7001, Australia*

[d] *School of Land and Food, University of Tasmania, Private Bag 76, Hobart 7001, Australia*

[e] *Department of Chemistry and Biochemistry, Mendel University in Brno, Zemedelska 1, 613 00 Brno, Czech Republic*

[f] *Central European Institute of Technology, Brno University of Technology, Purkynova 123, 612 00 Brno, Czech Republic. 118, 612 00, Brno, Czech Republic*

ARTICLE INFO

Article history:
Received
Received in revised form
Accepted
Available online

Keywords:
Field programmable gate array
Red Pitaya microcontroller
On-the-fly data acquisition and processing
IR LED
Rapid microsecond pulsing
Portable sensor

ABSTRACT

We designed a portable system using an open source microcontroller (μC) with built-in field programmable gate array (FPGA) for on-the-fly data acquisition and processing of optical data generated from rapidly pulsed infrared light emitting diodes (IR LEDs) for optical sensing of gases. The system is used for rapid pulse generation (ca. 2 μs short pulses with a typical repetition rate of 1 kHz) to drive the IR LED, as well as for the optical sensing data acquisition and processing on-the-fly large data streams of ca. 2 Gbit/s. The flexibility and performance of the system is demonstrated. Each of the digitally processed signal pulses yielded one data point of analytical signal in time as a quasi-continuous data stream produced at a rate of between 1000 and 0.1 Hz. This microcontroller–based portable open source platform is then implemented in on-the-fly data acquisition and processing, of analytical signals enabling continuous gas sensing.

1. Introduction

For analytical measurements where the sample property can change at a fast rate, such as in the case of atmospheric monitoring of trace gases, rapid digital sampling and analysis techniques are required [1]. This requirement is well satisfied with optical analytical platforms, such as infrared (IR) spectrometers, supplemented with adequately fast electronics and data handling capabilities. Although IR spectroscopy-based gas detection is a well-established technique [2], designing small low-cost low power consumption analytical platforms for portable and remote analysis presents a number of challenges [3]. One of them is the rapid, on-the-fly processing of continuous and live data streams in a flexible custom-defined manner. Additionally, low-cost, small size and weight, and low-power analytical platforms capable of rapid, on-the-fly and custom-defined data processing are required in a number of field deployment modes including portable hand-held devices and remote sensing devices such as on-board unmanned aerial vehicles (UAVs).

Most gaseous analytes of environmental or industrial significance have strong absorption bands in the infrared (IR) spectral range [4]. Most commercially available instruments for the analysis of gases employ sophisticated and expensive spectrometers that provide measurements solely in a laboratory setting [5, 6]. Light emitting diodes (LEDs) have proven to be in many ways ideal light sources for optical detection and sensing in portable format [7-9]. In this context, the use of LEDs with photodiodes (PDs) in the IR spectral range has enabled the development of portable low-cost sensors [10-12]. Recently we demonstrated that response of MIR LED-based absorption photometric sensor for methane can be predicted from 1st principles using the readily available molecular absorption data HITRAN, resulting in calibration line slope agreement to ± 1% with experimental data [13].

In this paper we advance this work in investigating a programmable fully portable sensor for methane using a powerful hand-held open source microcontroller with field programmable gate array (FPGA), to our best knowledge used for the first time for portable analytical instrumentation. The FPGA was capable of rapid on-the-fly processing of large data streams of up to 2GB/s which then resulted in the demonstration of continuous sensing of methane in indoor and outdoor environment (Part 2).

Most of the commercially available LED drivers for pulse signal generation (in µs pulses with a 1kHz frequency) have fixed settings and need additional electronics for signal collection and data acquisition [14, 15]. In most cases an oscilloscope or standard electronic data acquisition (eDAQ) system can be a good option. However, the maximum sampling rate for data acquisition of a typical eDAQ is only 1 kHz, which is not adequate for the acquisition of data generated by microsecond pulses [16]. Expensive digital oscilloscopes with sampling rates in excess of 100 mega samples/sec can be implemented for data acquisition and collection, however, the acquisition of large data streams (in our case 125 MHz/16bit yielding 2 GB/s) will exhaust the memory of a typical 16 GB SD card in only ca.1 min [17].

Nowadays, computers are omnipresent as an interface with analytical instruments for online digital data acquisition, processing, storage, and display [18, 19]. Some are capable of precisely handling large data streams in modern laboratory-based (not portable) analytical instruments such as Raman spectrometry [20]. However, for miniaturized portable analytical instruments, modern powerful microcontrollers (µC) are ideal where on-the-fly (live data) data processing is needed [8, 21]. Recently the Hauser group published a review covering the use of µC for portable analysis [8]. Although a number of µC based commercial devices, including those for detection of methane, are available [22], these lack flexibility and give no insight into the way the analytical signal is produced ('black box'), so that it is in principle impossible to make a judgement on the data processing.

Currently, open source µC systems, such as Arduino, are popular due to their programmable options [23], however, the Arduino can handle only one operation at a time and the maximum sampling rate of its in-built analog-to-digital converter (ADC) is only 10 kHz [24]. Another popular open source µC, Raspberry pi, has no built-in analog input, therefore one has to be implemented using an additional ADC [25]. Conventional ADCs perform single conversions at a time, which results in a random lag between analog signal acquisition and data processing, making it difficult to generate synchronised data [26]. Importantly, even with a very fast ADC, the Raspberry pi is not capable of processing data in a rapid manner due to the speed limit of its processor. Regarding the most important parameter in respect to this work, namely on-the-fly large data stream data processing, both the Arduino and Raspberry pi would not be able to handle data streams in excess of 50 kHz at 16 bit 0.1 MB/sec [27].

Alternatively, a recently introduced portable microcomputer with a field-programmable gated array (FPGA, which enhances the processing capabilities of existing microprocessors) 'Red Pitaya' (a technology spin-off from Instrument Technologies as the makers of Libera family devices [28]) is capable of on-the-fly processing of large data volumes without any lag, thanks to the FPGA responsible for data synchronization [29]. An FPGA allows for the integration of the ADC interface, input/output (I/O) interface, memory, and processing units in a single chip [30]. FPGA-based devices are especially used in particle colliders for high-energy physics (HEP) [31], gamma radiation spectroscopy, real vision imaging and many other types of reconfigurable high performance virtual instrumentation [32]. Although FPGA based devices offer real-time data handling capability of large data without any lag, other than in the fields of nanosecond pulse generation [33], computational chemistry [34] and simulated mass spectroscopy (MS) [35], the application of a µC system with FPGA in analytical chemistry to the best of authors' knowledge, has not been presented in the analytical literature.

Therefore, we aimed to investigate flexible data acquisition and on-the-fly fully automated data processing through developing an in-house data processing routine capable of handling large and live data using a µC system with an FPGA in a rapid manner. This creates the capability of generating rapid pulsed signals and processes in real time giving large amounts of data per second (2 Gbits/s at 125 MHz with 16bit ADC), where implementation of a miniaturized µC with an FPGA for IR LED based optical gas sensing offers portability, and at the same time maximum flexibility for implementing codes for specific analytical scenarios such as portable and remote analysis of gases.

2. Instruments and Methods

2.1. Instrumentation

2.1.1. Microcontroller with field programmable gate array

The microcontroller (µC) system (Red Pitaya V1.1, RS Components Pty Ltd, Wetherill Park, NSW, 1851, Australia) shown in Fig. 1 is an open source platform, based on an ARM Cortex A9 processor plus a Zynq µC system on chip (SoC) field programmable gate array (FPGA) in the same device (component A in Fig. 1) with 512MB of DDR3 RAM (component B in Fig. 1). The operating µC system is based on Linux (version 2015.1 from Xilinx) supporting network connection (WIFI, LAN and USB), which allows it to operate remotely. The ARM CPU functions as a data analyser to evaluate the data collected by a high-speed ADC. The sampling capability of this µC system through RF output and input (components C and D in Fig. 1) has four different options from 2-125 MHz. The buffer size (maximum data capture capacity) of the FPGA-µC system is 16,384 points. The input and output buffer of the FPGA-µC system was self-triggered using its external triggering facilities in the GPIO (shown as component E in Fig. 1).

Insert figure 1

2.1.2. In-house electronics: Voltage to current converter and resistor-capacitor circuit or RC filter.

We developed a voltage to current conversion unit (V-to-I) and a resistor-capacitor circuit as an RC filter in-house with off-the-shelf electronics. The V-to-I circuit converts a voltage pulse, generated by the Red Pitaya FPGA, to a current pulse to drive the LED.

The RC filter has two 1 ohm resistors and a 390 µF capacitor. There is also a dummy load resistor of 27 ohms connected across the output. The dummy load is necessary so that the power bank does not switch off if the load becomes too small. The filter supplies power to the LED driver (V-to-I) so that the 2 amp pulse is not affecting the detector which is supplied from the same power bank. See the supplementary information (SI) Fig. S1 A&B for a detailed circuit diagram of the V-to-I, and RC filter.

2.1.3. LED and photodiode

We used an IR LED with an emission maximum wavelength λ_{max} = 1.65 µm, (Lms16LED-R, Alfa Photonics, Latvia) and an IR sensitive photodiode (PD) (Lms24-05-PA, Alfa Photonics, Latvia) having a spectral response over the range from 1.1 to 2.3 µm equipped with embedded preamplifier.

2.1.4. Power supply.

We employed a rechargeable portable power bank (CY1767PBCHE, Cygnett, Australia) to supply 5 volt DC power to the µC system, V-to-I conversion circuit, and the preamplifier circuit of the IR PD.

2.2. Method

2.2.1. Pulse generation

We used the signal generation feature of the microcontroller (µC) system (Red Pitaya) to generate the desired voltage pulses. Signal generation of the µC operates by filling a floating-point array of up to 16384 values with the desired voltage at each time point, then commanding the FPGA to produce that voltage pattern at the desired frequency. They were subsequently converted into current pulses by the voltage to current converter (V-to-I) unit to drive the IR LEDs in pulse mode. The shape of the pulse is flexible, generated stepwise digitally with details described in the Results and Discussion section.

After receiving the radiation from the pulsed IR LED, optical pulses from the IR PD were acquired by the µC system through Input Channel 1 as analog voltage pulses. An 800 mm long and 7.5 mm inner diameter electro polished aluminium tube was used as a sample cell for portable analysis, which also housed the IR LED and the IR PD (shown in Fig. 2 in the Results and Discussion section).

2.2.2. Fast data acquisition and on-the-fly data processing

The analog voltage pulses from the IR PD which were collected by the µC system through the RF input high speed ADC (input channel 1) were designated 'raw pulses' for clarity. We constructed a data processing program for the ARM cortex A9 processor of the µC system to perform digital smoothing on the raw pulses. The digital smoothing utilized three techniques: repetitive smoothing (averaging a number of consecutive pulses) alone, and in addition to the repetitive smoothing, boxcar averaging, or Savitzky-Golay smoothing (a special form of 2nd polynomial regression-based smoothing). After the application of the smoothing techniques, the smoothed pulses were termed 'processed pulses' (for more information see Fig. 4).

The baseline and pulse top of each of the processed pulses were evaluated using three different statistical operations: averaging, linear regression, and 2nd degree polynomial regression. From the evaluated values of baseline and pulse-top, the height of the pulse was calculated (by subtraction), resulting in one data point for each processed pulse, termed the 'final signal value'(S). After acquiring additional quasi-continuous final signal values from an arbitrary number of pulses, the 'digital data signal' in volts was formulated and by taking the negative natural logarithm (base e) of the data stream values, the digital data stream was converted into the 'final analytical signal' (A) in absorbance units (A.U.) (for more information see Fig. 4). The total calculation time for the µC was ~20 ns.

Baseline noise was evaluated by observing the distribution pattern of all the data points in the baseline of each of the pulses for random and fixed instrumental noise in our detection system. The distribution pattern for the final digital signal was tested on the obtained results using two different statistical evaluation techniques: simple averaging, and linear regression. Both the instrumental and analytical signal to noise ratio was calculated and the result was optimized by comparing one with another.

3. Results and discussions

3.1. Design of Pulse Generation for IR-LED

The radiometric power output of LEDs increases proportionally with the magnitude of the applied current [36]. However, the temperature across the chip of the LED rises significantly when it is driven at a higher applied current [37], which causes efficiency droop (i.e. the efficiency of the LED decreases while operated with higher electric current) due to overheating across the semiconductor material of the LED chip [36].

Therefore, to minimize the effect of overheating of the semiconductor materials used in the IR LED, the LEDs have to be operated either in a quasi-continuous wave (QCW) mode (duty cycle = 50%) or in pulsed mode (switched on for a very short time, usually microseconds). The maximum driving current in QCW for the IR LED is 250 mA [38], whereas in pulse mode the driving current can be up to 2A [38], which yields higher radiometric power output during the pulse [36] and this in turn yields in better performance of the optical measurement due to lower minimum absorbance values that can be measured by absorbance-based analytical detection [15, 39].

In rapid pulsing mode, the duty cycle (the percentage of the 'on' time) is significantly shorter, which helps to reduce the thermal effect. In our work, the IR LED was in "on" mode only for 2 µs with a duty cycle equal to 0.2%, so the LED was in "off" mode for a comparatively longer period (998 µs), which allows sufficient time to cool down and protect the LED from efficiency droop. The corresponding pulse repetition frequency (PRF) in our study was 1 kHz, which helps to produce a higher number of pulses within a short period of time with resulting maximum radiometric power output. Hence, 0.2% duty cycle provides data processing suitability at such high PRF since the pulse width determined the number of data points to be processed.

To further demonstrate the flexibility of this approach with custom-defined data processing, we developed a computer program written in C and compiled in a Linux OS environment to generate the voltage pulses in the required shape. This program was employed to forward an array of voltage values, with stepwise amplitudes between 0 and 1 volt, from the µC system to the voltage-to-current converter (V-to-I) circuit shown in Fig. 2. Since LEDs are current driven, we applied the in-house voltage-to-current (V-to-I) converter circuit to transform the voltage pulsed signal generated (V_{in}) by the µC system into current signal pulse (I_{in}) for the IR LED.

Insert Figure 2

As mentioned before, generated pulses can be produced in arbitrary shapes i.e. any time duration (pulse width or duty cycle), frequency, and forward voltage are possible simply by changing the parameters of the program. The stepwise pulses generated by the µC are shown in Fig. 3A (i) as a continuous stream and in Fig. 3B (i) as single pulse which was constructed by the following: 500 steps with 0 volts to achieve the base line, 20 steps to achieve 0.9 volt, 180 steps to make the pulse top with 0.9 volt, 20 steps to bring the pulse signal down to 0 volt and the remaining steps to fill the buffer at 0 volt (1 step = 10ns. The pulse generated was repeated with 1 kHz frequency, and the total time duration depends on the number of pulse data that need to be processed. The corresponding converted currents pulsed from the V-to-I conversion unit, and measured in channel 2 of the µC are shown in Fig. 3 A (ii) and 3 B (ii). These currents were used to drive the IR LEDs in pulse mode. IR radiation from the LED was detected by the IR PD and transformed from an optical pulse signal to voltage pulses (V_{out}) as measured in Channel 1 of the µC shown in Fig. 3A (iii) and 3B (iii). This voltage pulsed signal was collected by the µC system as a raw pulsed signal and employed for further processing.

Insert Figure 3

From Fig. 3 B i) it is observed that the stepwise generated pulse from the µC system follows a smooth shape, with a sharp rise and fall as it is generated. However, when it was converted into current pulses by the V-to-I conversion circuit the LED has a rise time of 200 ns to generate the final optical output (~2A). After detecting the response from the IR LED, the IR PD has a "rise time" and a "fall time" of 250 ns a shown in Fig. 3 B. The response delay appears due to inherent properties of the semiconductor material of the IR LED and IR PD and the response of the embedded PA and therefore, cannot be controlled by the user.

3.2. Data Acquisition and on-the-fly Data processing with µC system

By default, Red Pitaya FPGA performs data acquisition in continuous mode, which may result in overwriting and loss of necessary data for further processing in pulsed mode. Therefore, data acquisition was performed through a command to the FPGA to acquire a full buffer of 16384 points as an array of digital numbers. Triggering was used to ensure the µC system only collected the informative part of the raw signal that included the baseline and the entire pulse.

To eliminate time lag between data acquisition and data processing, the input and output buffer of the µC were synchronized using self-triggering. Self-triggering was performed by employing the external triggering facility of the FPGA where the digital output from the GPIO (component E in Fig. 1) was fed back as the external trigger. It was programmed by raising the pin from low (0 volt) to high (3.3 volt), keeping the pin high for 5 µsec then allowing it to fall back to low, triggering both RF input and output to function simultaneously

3.2.1. Digital filtering by repetitive smoothing, boxcar averaging and Savitzky-Golay smoothing

In order to achieve smooth pulses from the acquired raw pulses, we incorporated three digital filtering techniques namely, repetitive smoothing, boxcar averaging, and Savitzky-Golay smoothing through C programming in the CPU of the µC. When digital filtering software is incorporated in commercially available analytical instruments, analysts lose the flexibility of investigating different digital filtering techniques with variable input data according to specific analytical requirements [15]. Our digital filtering approach with the µC system allowed us to select flexible numbers of raw pulses starting from time zero (triggered on) to perform the repetitive smoothing by averaging consecutive pulses [40]. Then boxcar averaging, and Savitzky-Golay (S-G) methods [15] using point wise data after the repetitive smoothing also produced smoothed processed pulses. The selection criteria for the large number of raw pulses for processing are described in the SI.

In the data processing program, we started with repetitive smoothing of different numbers of raw pulses being averaged (A = 10, 100, 1000, and 10,000) (Shown in Fig. 4A). We then employed Boxcar averaging and Savitzky-Golay methods on 1000 pulses already smoothed by repetitive smoothing to investigate whether further smoothing after repetitive smoothing is required (discussed in section 3.3.2).

In order to obtain the final signal (height of the pulses) from both raw and processed pulses we applied three different statistical operations in the program: simple averaging, linear regression, and 2nd degree polynomial regression, to evaluate the base line and pulse top (shown in Fig. 4B). At 0.2% duty cycle with 100M/s data rate, 200 points were delivered while the LED is on only for 2 µs. Therefore, we selected 125 data points (discarding the pulse rise and fall) from the pulse top and 150 data points from the base line (before the rise) for each statistical operation. The difference between the pulse top and baseline i.e. pulse height is considered the final signal value for each individual pulse. Schematic representations of the final evaluated signal as pulse height and the final stream of quasi-continuous data are shown in Fig. 4C.

Insert Figure 4

3.2.2. Baseline noise evaluation and instrumental signal-to-noise (SNR) from the processed pulses

A. Baseline noise evaluation.

Theoretically, Gaussian (white) noise attenuates with square root of the number of repetitive pulses, while other types of noise will not, and therefore the additional Boxcar and Savitzky-Golay smoothing along with the repetitive smoothing might not be equally beneficial. To assess the nature of the noise of the processed data, we have conducted statistical analysis based on histograms of the baseline data point values. The histograms were constructed using 150 data points (A=1000) in the baseline of processed pulses, using repetitive smoothing, Boxcar and repetitive smoothing, and Savitzky-Golay and repetitive smoothing as shown in Fig. 5A. From the histograms in Fig. 5A we observed that the baseline signals follow a normal distribution for repetitive smoothing and for repetitive smoothing + Boxcar, while the repetitive smoothing + Savitzky-Golay resulted in a distribution skewed to the right. The characteristic appearance of the normal distribution of the baseline data values in this study confirms that the baseline signals resulting from repetitive smoothing and repetitive smoothing + Boxcar averaging methods include only white noise [41, 42].

We determined the baseline noise of the processed pulses after repetitive smoothing by multiplying the standard deviation (σ) of 150 baseline data points by 5, following a classical noise evaluation technique in analytical chemistry for flow-through detection, and found a good agreement with theoretical noise values (as shown in SI Fig. S2A). This also confirms the Gaussian nature of the baseline noise in the processed pulses through repetitive smoothing. The baseline noise values after applying the three different smoothing techniques are shown in SI Fig. S2B. As expected, the repetitive smoothing followed by additional smoothing techniques resulted in lower white noise in the baseline with Boxcar averaging being 7% and Savitzky-Golay 5% lower.

The flexibility in this type of baseline noise evaluation with the µC system provides users with the capability to choose from a variety of digital filtering-by-smoothing techniques. Boxcar averaging and Savitzky-Golay smoothing didn't result in a statistically significant reduction in noise, so we have chosen repetitive smoothing only for further investigation in this study with a view to providing rapid data processing with the proposed µC based detection system.

3.2.3. Instrumental signal-to-noise (SNR)

The quality of an analytical method is very often quantified by analyzing the signal-to-noise ratio (SNR). We determined the instrumental SNR for each resulting signal value (pulse height) and baseline noise obtained from each processed pulse after employing different smoothing techniques. In Fig. 5B we compare the SNR values obtained after employing repetitive smoothing as a function of the number of pulses being averaged (A) using three different statistical methods. We observed that the SNR improves as the number of pulses being averaged (A) increases. The enhancement of SNR follows the theory where SNR improves linearly by a factor of √A (shown in Fig. 5C) [42]. However, different statistical operations have no observable effect on the SNR values. All the measurements were reproducible since the standard deviations were too small to notice the error bars for 10 repetitions of each result.

Insert Figure 5

3.2.4. Determination of the digital data stream

A. Pulse top and baseline signal distribution

In order to select the most suitable statistical method for the determination of the pulse top and baseline signal values for each subsequent pulse signal (pulse height), we investigated the distribution of the pulse top and baseline signal values, applying simple averaging and linear regression for both cases. We omitted the 2nd degree polynomial as we did not observe any significant difference in the SNR values using simple averaging, linear intercept, or 2nd degree polynomial methods as illustrated in Fig. 5B. In Fig. S3A of the SI we have shown the baseline and pulse top signal values obtained from evaluating 10,000 consecutive raw pulses (without applying any digital smoothing) using simple averaging and linear regression. We constructed the histogram using these data values (10,000 baselines and pulse tops) as shown in Fig. S3B and S3C, and we observed that the simple averaging method resulted in normal distributions as well as smaller standard deviations when compared to linear regression for both evaluations of baselines and pulse tops.

B. Final quasi-continuous data stream.

From the distribution pattern of baseline and pulse top signal values (Fig. S3B & S3C) it is evident that simple averaging has less deviation when it is used for signal evaluation. Therefore, we considered the simple averaging method to investigate the final signal (pulse height) distribution while averaging consecutively increased numbers of pulses for repetitive smoothing, and referred to the result as the final digital data stream as shown in Fig. S4. The distributions of the final pulsed signal (height of the pulse) became smooth and, as expected, the analytical noise (in voltage) of the final data stream lowered as the number of consecutive pulses being averaged for repetitive smoothing increased.

C. Evaluation of analytical signal by converting the voltage signal into an absorbance signal.

Since the ultimate usage of the proposed optical system is analytical sample detection based on the absorbance principle, we converted each voltage signal of the final data stream (i.e., Fig. S4) into an absorbance unit (A.U.). By taking the negative natural logarithm (base e) of the final signal (pulse height) values, the minimum measurable absorbance was determined [41] and referred to as the final analytical signal for this proposed absorbance based detection system. In Fig. 6 the final absorbance signal is shown with corresponding analytical noise values (ΔA in A.U.). It is evident that as the number of repetitions of pulses for smoothing increases, the noise drops by a factor of the square root of the number of pulses, which consequently helps to improve the performance of any analytical detection. However, while the number of repetitive pulses increases, the time required for processing each pulse increases simultaneously and the rapid instantaneous processing capability of the system therefore decreases. Hence, where fast data processing is the principal focus, the number of repetitive pulses being smoothed needs to be optimized.

Insert Figure 6

3.3. Optimization.

In this section, the number of pulses to be averaged for repetitive smoothing are optimized for the proposed IR detection system using on-the-fly data processing which will be exercised further in the field for real sample analysis. For this, the analytical signal-to-noise ratios (ratios of the averaged absorbance values obtained from Fig. 6 to the corresponding absorbance noise values (A/ΔA)) were compared with the instrumental signal-to-noise ratio (SNR) shown in Fig. 7A. In both cases, the signal-to-noise ratio increased as the number of pulses being averaged increased and almost the same pattern was followed although they were appraised independently for different data sets following different evaluation approaches. The analytical or absorbance noise values (ΔA) were also compared with the instrumental or baseline noise values (N) of each processed pulse as shown in Fig. 7B. From Fig. 7B, we observe that the analytical and instrumental noise values merge with each other at A=10,000 (for repetitive smoothing) which is the consequence of fixed instrumental noise and the pulse being almost smoothed. Although A=10,000 gives the optimal result in terms of noise elimination and signal-to-noise ratio enhancement, the time needed for each pulse to be smoothed for A=10,000 is 10 seconds, which in some cases may not be ideal in terms of rapid data processing. Therefore, to keep the system response fast enough for most real-time sensing scenarios, A=1000 (1 second for each data point) was chosen for on-the-fly data processing for in-field real sample analysis.

Insert Figure 7

4. CONCLUSIONS

This study demonstrates the prospects for rapid data processing of large data streams on modern portable µC systems with field-programmable gate arrays, for on-the-fly and rapidly changing sample scenarios. Further it shows the benefits of flexibility and full insight based on a custom data handling routine implemented in an open source µC system with a FPGA. The user-defined data processing thus acquired and implemented through the µC system in a flexible manner also allows the generation of the required pulsed signal with any desired shape, duration, frequency, and amplitude to drive the LED. The user can define and adapt the data processing software and apply it in a flexible way as required. The use of such miniaturized µC-FPGA systems with custom data processing routines and high reproducibility makes the analytical sensing of rapidly changing samples, such as atmospheric gases, a real and relatively low-cost possibility.

Acknowledgments`

MM acknowledges his ARC Future Fellowship Level 3 (FT120100559). The authors have declared no conflict of interest.

Supplementary information available

Detail of the in-house made instruments and figure for baseline noises, Pulse top and baseline signal values, constructed histograms and quasi-continuous digital data stream using two different statistical methods and command line parameter for program are given in the SI.

References

[1] J. Leis, D. Buttsworth, C. Snook, G. Holmes, Detection of potentially explosive methane levels using a solid-state infrared source. IEEE Trans. Instrum. Meas., 63 (2014) 3088-3095.

[2] J. M. Dang, L. Fu, Z. H. Yan, C. T. Zheng, Y. C. Chang, C. Chen, Y. D. Wang, A Review of Mixed Gas Detection System Based on Infrared Spectroscopic Technique Spectrosc. Spectral Anal. 34 (2014) 2851-2857.

[3] S. Fanchenko, A. Baranov, A. Savkin, V. Sleptsov, LED-based NDIR natural gas analyzer. In IOP Conference Series: Materials Science and Engineering. 108 (1) IOP Publishing, 2016. http://iopscience.iop.org/article/10.1088/1757-899X/108/1/012036/pdf

[4] L. S. Rothman, I. E. Gordon, Y. Babikov, Y., Barbe, A., Benner, D. C., Bernath, P. F., et al (2013). The HITRAN2012 molecular spectroscopic database. J. Quant. Spectrosc. Radiat. Transfer, 130 (2013) 4-50.

[5] K.-H. Kim, S. K. Pandey, R. Pal, Analytical bias among different gas chromatographic approaches using standard BTX gases and exhaust samples, J. Sep. Sci. 32 (2009) 549-558.

[6] N. S. Lawrence, Analytical detection methodologies for methane and related hydrocarbons.Talanta 69 (2006) 385-392.

[7] M. Macka,, T. Piasecki, , P. K. Dasgupta, Light-emitting diodes for analytical chemistry, Annu. Rev. Anal. Chem. 7 (2014) 183-207.

[8] D. A. Bui, P. C. Hauser, Analytical devices based on light-emitting diodes–a review of the state-of-the-art. Anal. Chim. Acta. 853 (2015) 46-58.

[9] A. Noori, P. Mahbub, M. Dvořák, A. Lucieer, M. Macka, Radiometric analysis of UV to near infrared LEDs for optical sensing and radiometric measurements in photochemical systems, Sens. Actuators, B Chem, 262(2018)171-179.

[10] C. Massie, G. Stewart, G. McGregor, J. R. Gilchrist, Design of a portable optical sensor for methane gas detection. Sens. Actuators, B, 113 (2006) 830-836.

[11] G.-T. Park, K.-C. Park, G.-J. Lyu,; J.-R. Kwon,; Y.-G. Kim, B.-J. Ryou, J.-I. Park, In 24th World Gas Conference in October 2009: Buenos Aires, Argentine Republic, 2009.

[12] K. M. G. de Lima, A portable photometer based on LED for the determination of aromatic hydrocarbons in water. Microchem. J. 103 (2012) 62-67.

[13] P. Mahbub, J. Leis, and M. Macka, Chemometric approach to the calibration of light emitting diode based optical gas sensors using high-resolution transmission molecular absorption data, Anal chem 90(2018) 5973-5976.

[14] D-51i Universal Led Driver Instruction Manual, http://lmsnt.com/datasheets/Electronics/D51i_en-230317.pdf (: 13 November, 2018

[15] LED Pulse Drivers, http://www.roithner-laser.com/led_pulsed_driver.html, (11 November, 2018)

[16] EDAQ data recording made simple, https://www.edaq.com/, (accessed: 13 November, 2018)

[17] Digital Oscilloscope, www.tek.com/digital-oscilloscope, (12 November, 2018)

[18] T. A O'Haver, Pragmatic Introduction to Signal Processing, Univ. of Maryland at College Park, College Park, MD, 1997, 58–69.

[19] L. F. Capitán-Vallvey, N. Lopez-Ruiz, A. Martinez-Olmos, M. M. Erenas, A. Palma, Recent developments in computer vision-based analytical chemistry: A tutorial review, J. Anal. Chim. Acta 899 (2015) 23-56.

[20] D. Lauwers, P. Brondeel, L. Moens, P. Vandenabeele, In situ Raman mapping of art objects. Phil. Trans. R. Soc. A, 374 (2016) 20160039

[21] Y. Suzuki, T. Takahashi, T. Takayanagi, S. Motomizu, S. Kawakubo, Simple Kits of Colorimeter and Potentiometer for High School and Undergraduate Student Education, Bunseki Kagaku 59 (2010) 125-130.

[22] TMG: solutions by industry, http://www.tmgtestequipment.com.au/products/, (accessed: 28 October, 2018)

[23] I. J. Koenka, J. Sáiz, P. C. Hauser, Instrumentino: An open-source modular Python framework for controlling Arduino based experimental instruments. Comput. Phys. Commun. 185 (2014) 2724-2729

[24] P. Teikari, R. P. Najjar, H. Malkki, K. Knoblauch, D. Dumortier, C. Gronfier, H. M. Cooper, An inexpensive Arduino-based LED stimulator system for vision research. J. Neurosci. Methods 211 (2012) 227-236.

[25] V. Gandhi, S. Heda, R. Anand, A. Zarin, A. Upadhyay, A. L. Chakraborty, A Raspberry Pi-based field-deployable tunable diode laser spectroscopy system for the detection of CO2 at 2003.5 nm. In Microwave and Photonics (ICMAP), 2015 International Conference on; IEEE, 2015, pp 1-2. https://ieeexplore.ieee.org/abstract/document/7408713/

[26] M. Abdallah, O. Elkeelany, A. Alouani, A conceptual design of a compact multi-channel real-time analog signal acquisition and processing system. In 41st Southeastern Symposium on System Theory, 2009, pp 360-362.

[27] S. Scholl, The Xilinx Zynq: A Modern System on Chip for Software Defined Radios, (2016). In https://kluedo.ub.uni-kl.de/frontdoor/index/index/docId/4442, (1 August, 2018)

[28] A. Kosicek, Libera electron beam position processor, In Particle Accelerator Conference, 2005. PAC 2005. Proceedings of the; IEEE, 2005, pp 4284-4286.

[29] Redpitaya, Stemlab, https://redpitaya.com/, (7 November, 2018)

[30] H. Zhu, L. Sun, Z. Wang, Z. Nie, H. Liu, Design of infrared methane gas concentration detection system, J. App. Opt. 35 (2014) 890-894.

[31] K. T. Pozniak, FPGA-based, specialized trigger and data acquisition systems for high-energy physics experiments Meas. Sci. and Technol. 21 (2010) 062002.

[32] J. D. D. Gazzano, M. L. Crespo, A. Cicuttin, F. R. Calle, Field-Programmable Gate Array (FPGA) Technologies for High Performance Instrumentation; IGI Global, 2016.

[33] Y. Zhu, Wang, L. M. Design and Implementation of Nanosecond Pulse Generator based on Reconfiguration PLL in FPGA, In Proceedings of the 2015 International Conference on Electronic Science and Automation Control, Liu, C., Ed.; Atlantis Press: Paris, 2015, pp 323-326.

[34] A. Berces, B. Feher, P. Szanto, I. Pechan, L. Lajko, Z. Runyo, P. Laczko, J. Lazanyi, FPGA implementation of cheminformatics and computational chemistry algorithms and

its cost/performance comparison with GPGPU, cloud computing and SIMD implementations Abstracts of Papers of the American Chemical Society 240 (2010) 1.

[35] C. Pascoe, D. Box,H Lam,A. George, FPGA-Accelerated Isotope Pattern Calculator for Use in Simulated Mass Spectrometry Peptide and Protein Chemistry, IEEE. In 2012 Symposium on Application Accelerators in High Performance Computing; IEEE Computer Soc: Los Alamitos, 2012, pp 111-120.

[36] Energy, U. S. D. O., Ed.; Office of Energy Efficiency & Renew-able Energy: Washington, DC 20585, 2013.

[37] J. V. Lawler, J. Currano, Thermal simulations of packaged IR LED arrays, In Technologies for Synthetic Environments: Hardware-in-the-Loop Testing XIII: Orlando, FL, 2008.

[38] Alfa Photonics Ltd., http://www.alfaphotonics.lv/, (1 November, 2018)

[39] X. H. Geng, D. P. Wu, Q. Wu, Y. F. Guan, A compact and highly sensitive light-emitting diode-induced fluorescence detector for capillary flow systems, Talanta 100 (2012) 27-31.

[40] M. F. Wahab, P. K. Dasgupta, A. F. Kadjo, D. W. Armstrong, Sampling frequency, response times and embedded signal filtration in fast, high efficiency liquid chromatography: A tutorial Anal. Chim. Acta 907 (2016) 31-44.

[41] J. M. Palmer, B. G. Grant,. The Art of Radiometry, Bellingham: SPIE Press, 2010.

[42] P. D. Wentzell,; C. D. Brown, Signal Processing in Analytical Chemistry, In Encyclopedia of Analytical Chemistry, Meyers, R. A., Ed.; Ó John Wiley & Sons Ltd, Chichester, 2000, pp 9764–9800.

Figures

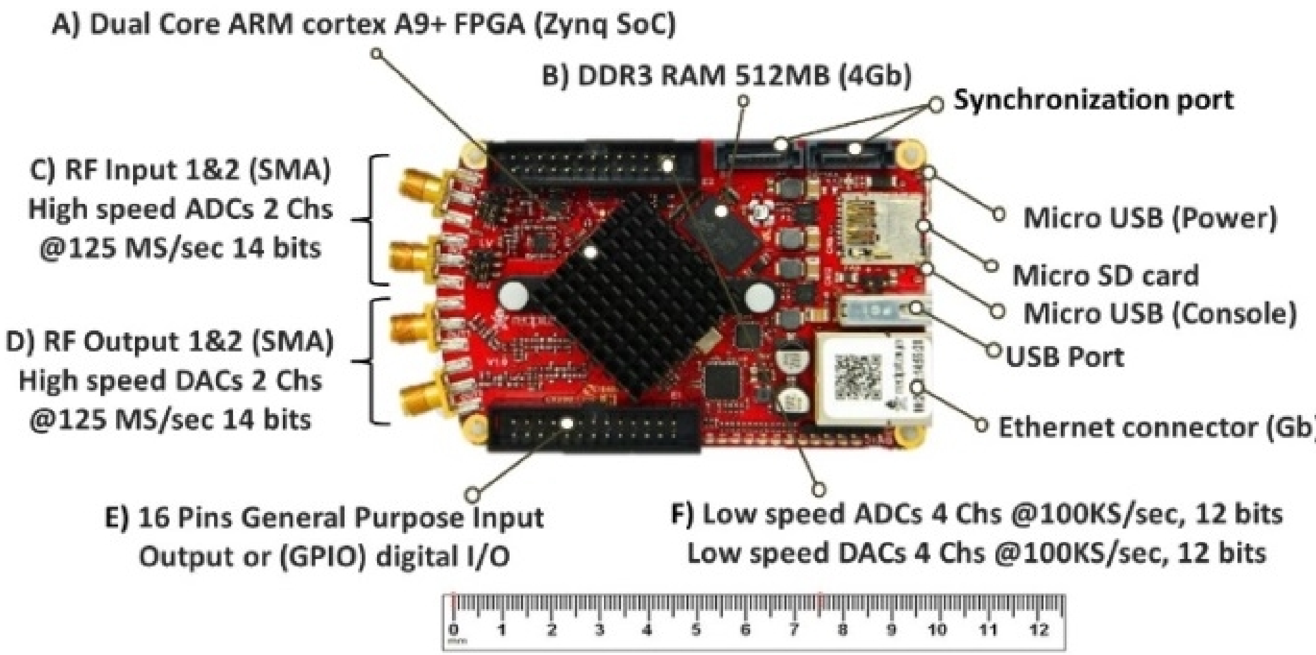


Fig. 1. µC system (Red Pitaya) showing the principal components: A) µC system Processor + FPGA, B) RAM High speed & resolution ADC input, C) High speed and resolution ADC input, D) High speed and resolution DAC output, E) General Purpose input and output (GPIO) which provides external self-triggering facilities, F) High speed ADC input and DAC output.

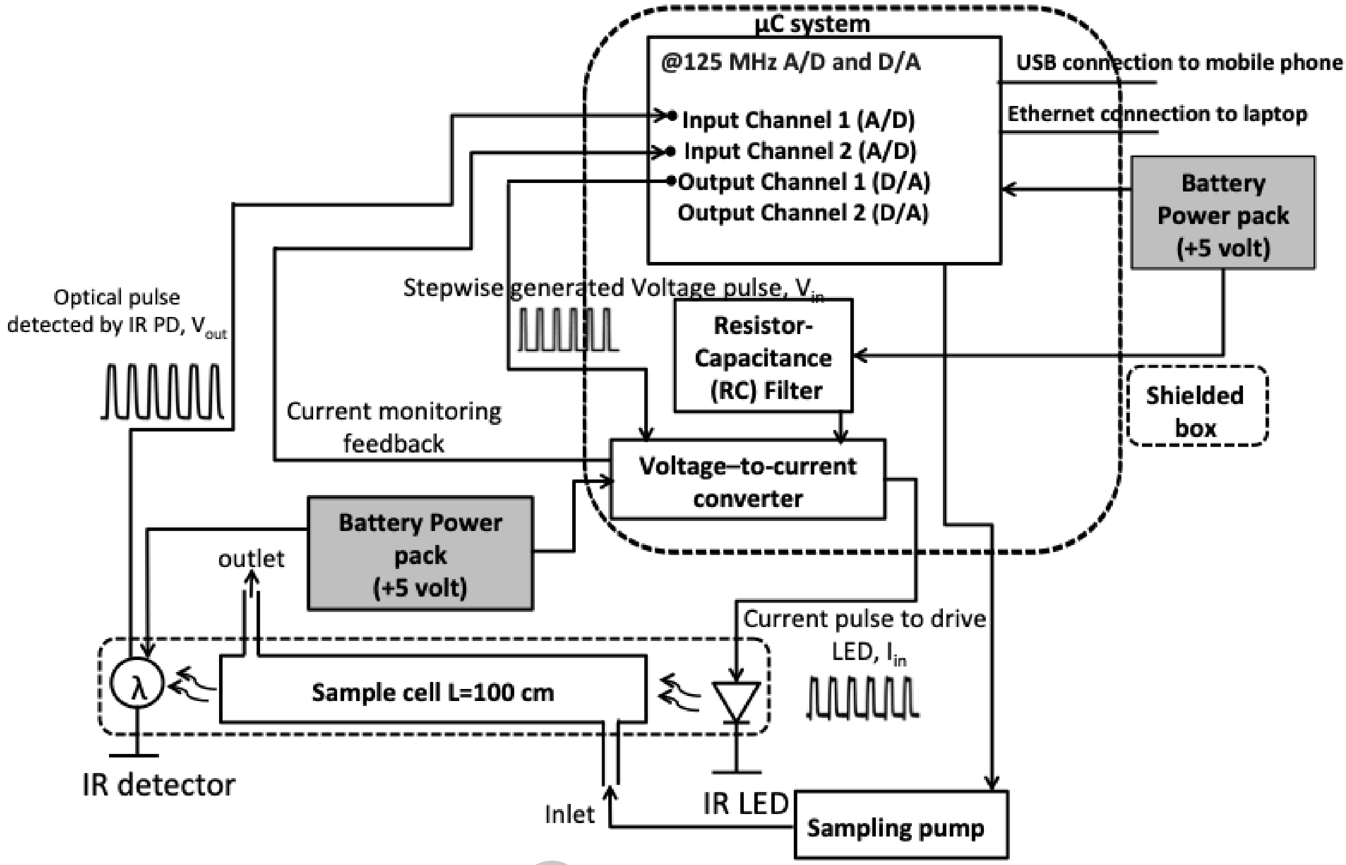


Fig. 2. Schematic representation of the IR detection associated with µC system as pulse generation and data collection system.

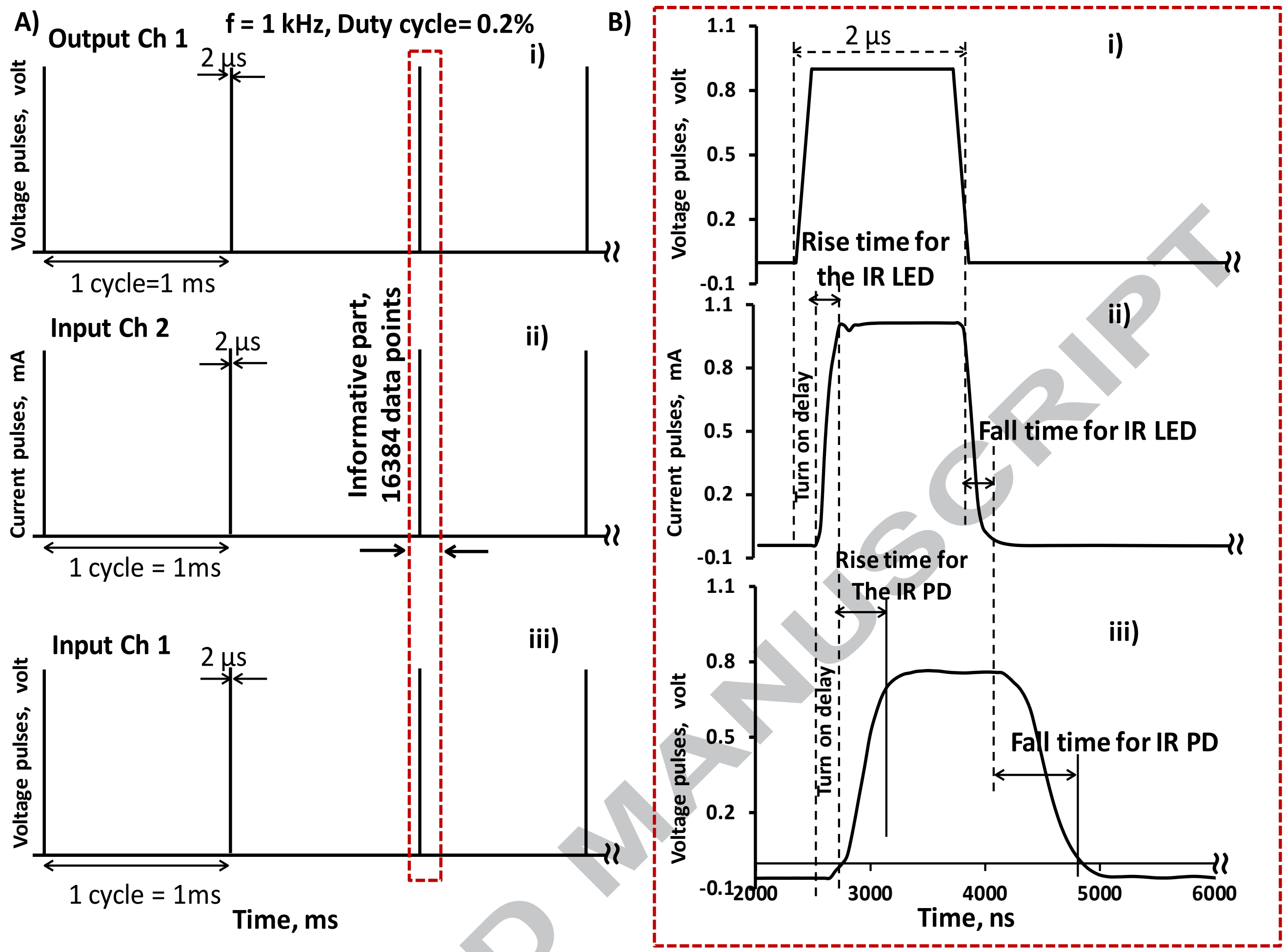


Fig. 3 A) Schematic representation of three types of pulsed signals generated and measured through **i)** output channel 1, **ii)** input channel 2 and iii) input channel 1 of the micro-controller system, informative parts (buffer size data points) are shown within the red dash lined rectangle. **B)** detail of each pulsed signal: **(i)** Step wise generated pulse defined by the µC system (Red Pitaya); **(ii)** current pulse to drive the IR LED, and **(iii)** the corresponding optical output pulses from the IR LED detected by the IR PD.

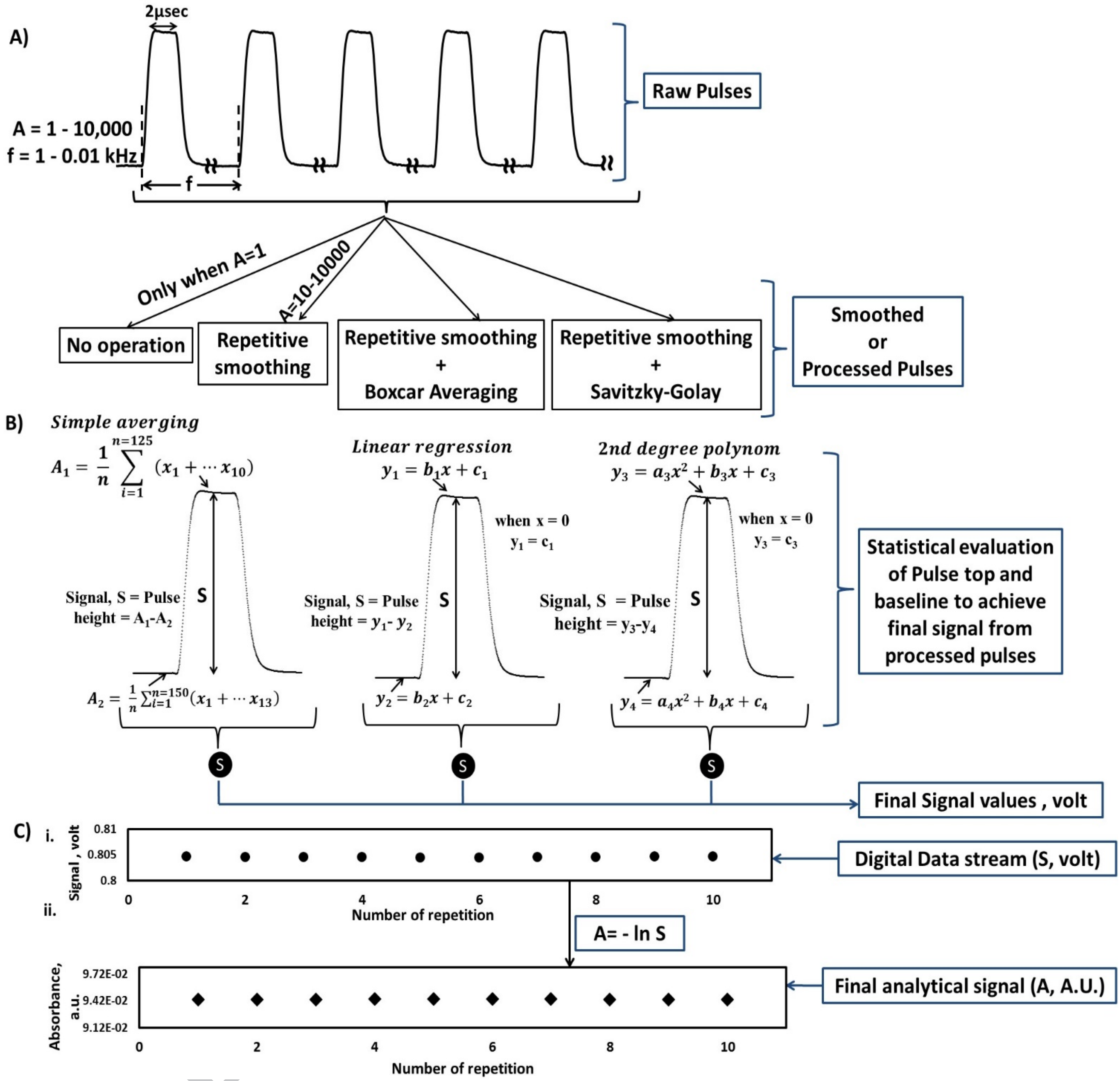


Fig. 4 Schematic representation of data processing methods A) repetitive smoothing and two additional digital smoothing techniques: Boxcar averaging and Savitzky-Golay applying on pointwise obtained data from repetitive smoothened pulses B) evaluation of baseline (from 150 data points) and pulse top (from 125 data points) of each processed pulse applying three different statistical methods to obtain final signal values (height of the pulses, S) C) Final data i. digital data stream in volt and ii. analytical signal in absorbance units (A.U.) from each individual pulse after the statistical evaluation of pulse top and baseline.

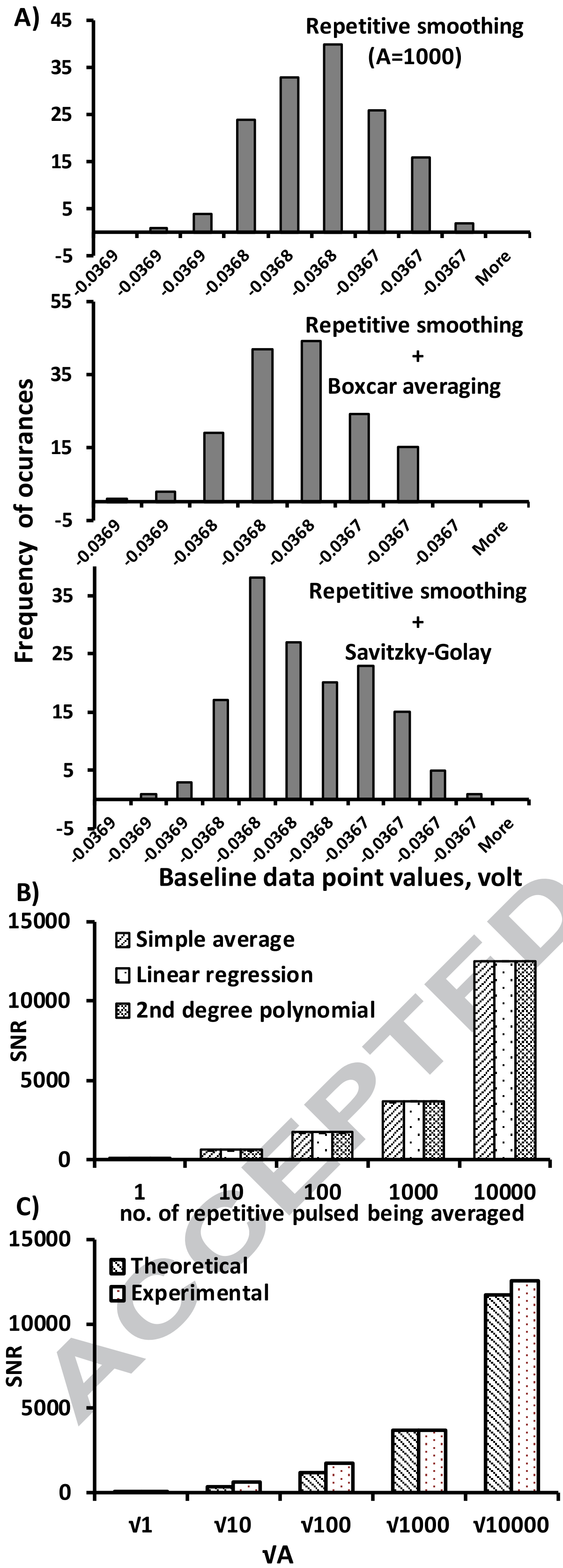


Fig. 5 A) Histograms of the baseline data points. The value of the baseline data points was obtained by binning the baseline data values, the difference between each of the bin values was = (max baseline – min baseline)/6.The histogram covers the whole range of baseline data point values. The number of points averaged (A) was 1000. B) Signal-to noise ratio obtained by applying three different statistical methods after smoothing raw pulses by repetitive smoothing and **C)** comparison of experimental SNR with theoretical values where SNR should increase by a factor of √A.

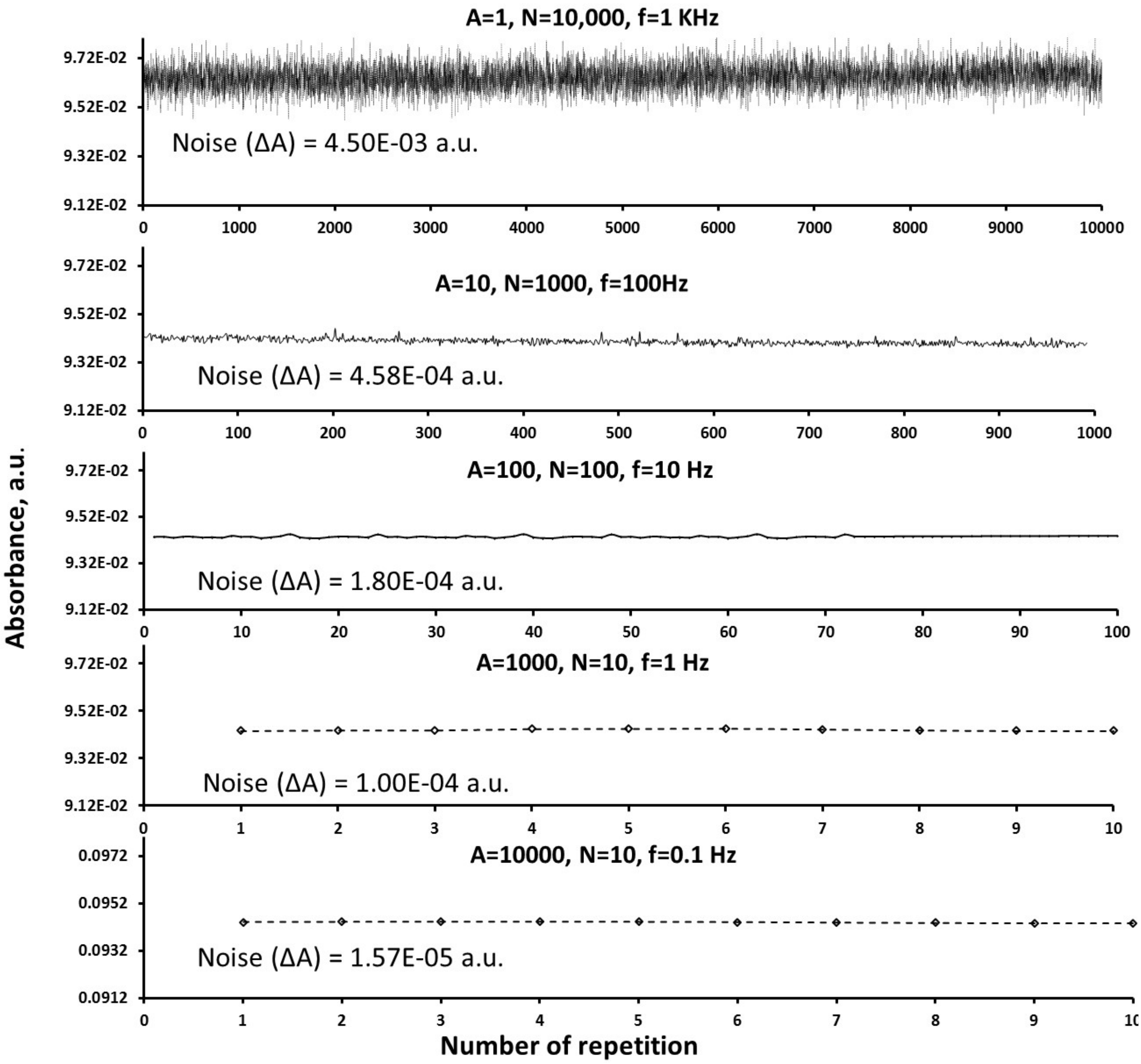


Fig. 6. Ultimate analytical signal in absorbance unit (A.U.) as negative natural logarithm of each signal values with corresponding noise values.

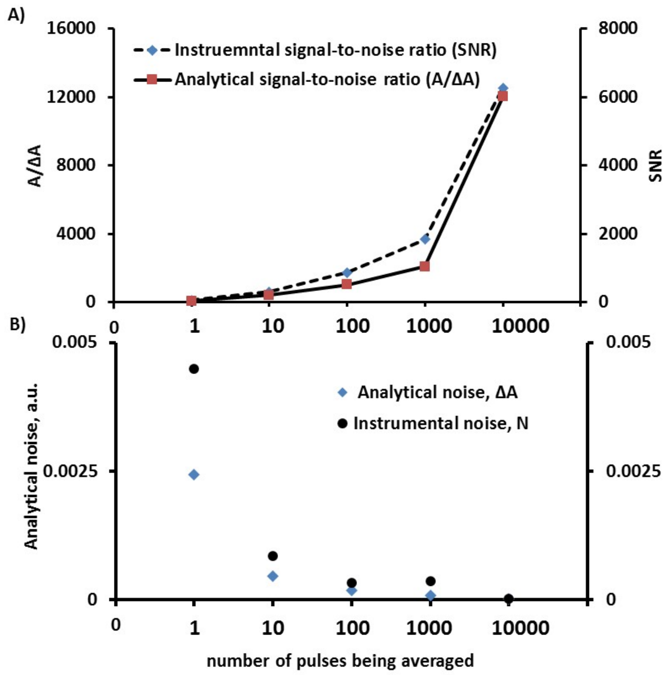
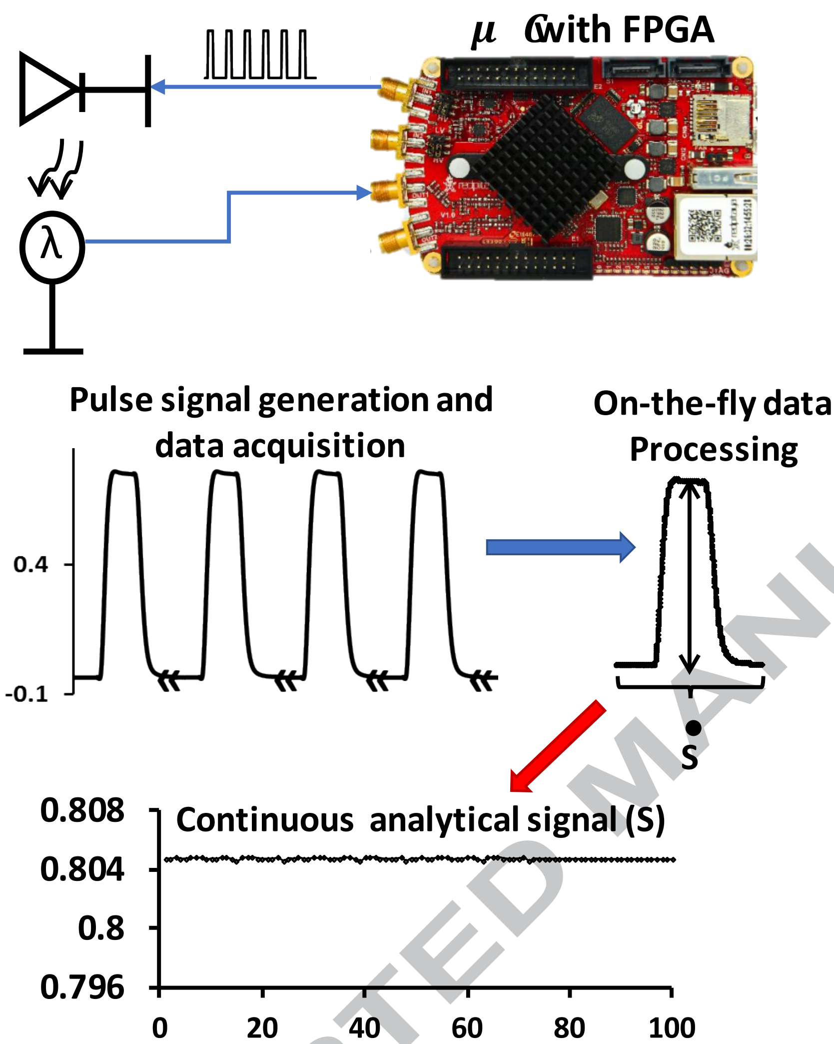


Fig. 7. Comparison of analytical and instrumental A) signal-to-noise ratio (A/ΔA) and (SNR) respectively, calculated for different set of acquired data at different start time. B) noises (noise = 5*σ) of same set data as A.

GRAPHICAL ABSTRACT ONLY



Supplementary Information

Portable Device for Continuous Sensing with Rapidly Pulsed LEDs – Part 1: Rapid On-the-fly Processing of Large Data Streams using an Open Source Microcontroller with Field Programmable Gate Array

Ansara Noori[a], Parvez Mahbub[a, b], John S. Parry[c], John Davis[c], Arko Lucieer[d], Mirek Macka[a, e, f]*

[a] *Australia Centre for Research on Separation Science (ACROSS) and School of Physical Sciences- Chemistry, University of Tasmania, Tasmania, Australia*

[b] *Institute for Sustainable Industries and Livable Cities, Victoria University, Footscray Park Campus, Melbourne, Victoria 3011, Australia*

[c] *Central Science Laboratory, University of Tasmania, Private Bag 74, Hobart 7001, Australia*

[d] *School of Land and Food, University of Tasmania, Tasmania, Australia*

[e] *Department of Chemistry and Biochemistry, Mendel University in Brno, Zemedelska 1, 613 00 Brno, Czech Republic*

[f] *Central European Institute of Technology, Brno University of Technology, Purkynova 123, 612 00 Brno, Czech Republic.*

1. Voltage-to-current conversion

The amplified output voltage from pin 6 of the op amp turns on the Mosfet (Fig. S1 A Q1) allowing current to flow through the LED (Fig. S1 D2) and resistors R10, R11. The voltage increased across the resistors until it is equal to the voltage applied to pin 3. This voltage is fed back to the op amp (D) through pin 2 which holds the current constant until the voltage applied to pin 3 changes to a different value

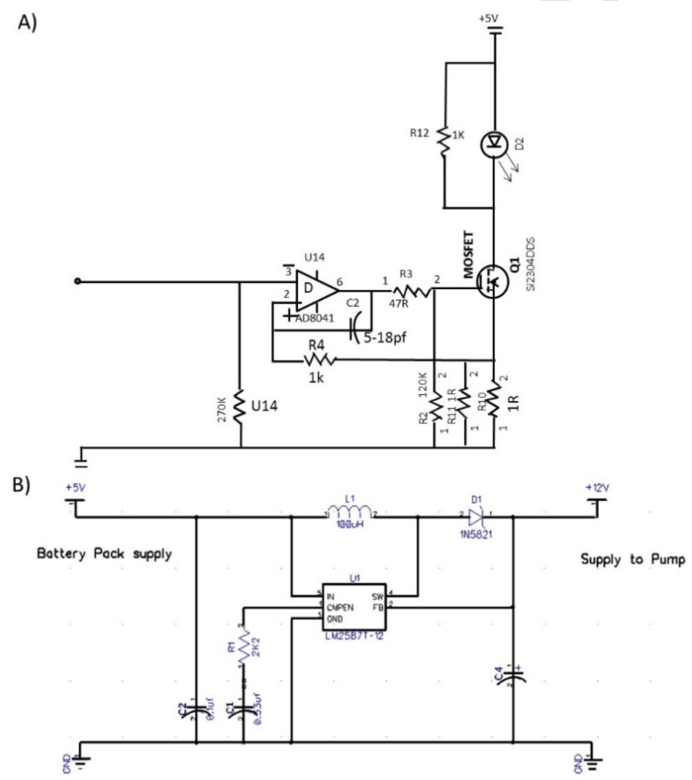


Fig. S 1 Block diagram of the in-house made **A)** voltage-to-current (V-to-I) conversion circuit **B)** resistor-capacitor (RC) filter.

2. Other Figures

Figures for baseline noises, pulse top and baseline signal values, constructed histograms and quasi-continuous digital data stream using two different statistical methods are given in Figure S2, S3, S4 respectively. In Figure S2 A the theoretical noise was calculated using equation 1,

$$N_{theoretical} = \frac{S_{max} - S_{min}}{6} \qquad (1)$$

Where, $N_{theoretical}$ is the base line noise, S_{max} is the maximum signal value in the baseline and S_{min} is the minimum signal value in the baseline.

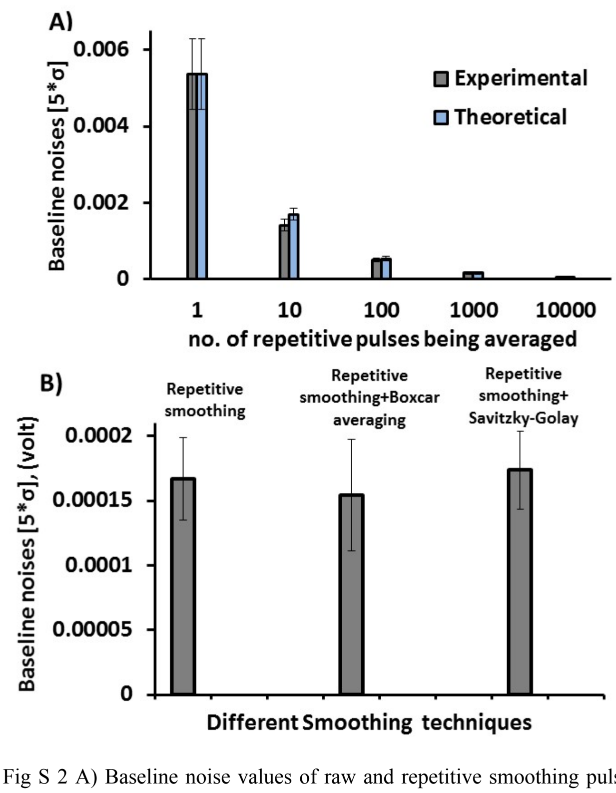


Fig S 2 A) Baseline noise values of raw and repetitive smoothing pulses compared with theoretical values B) Comparison of baseline noise values obtained after three different smoothing techniques.

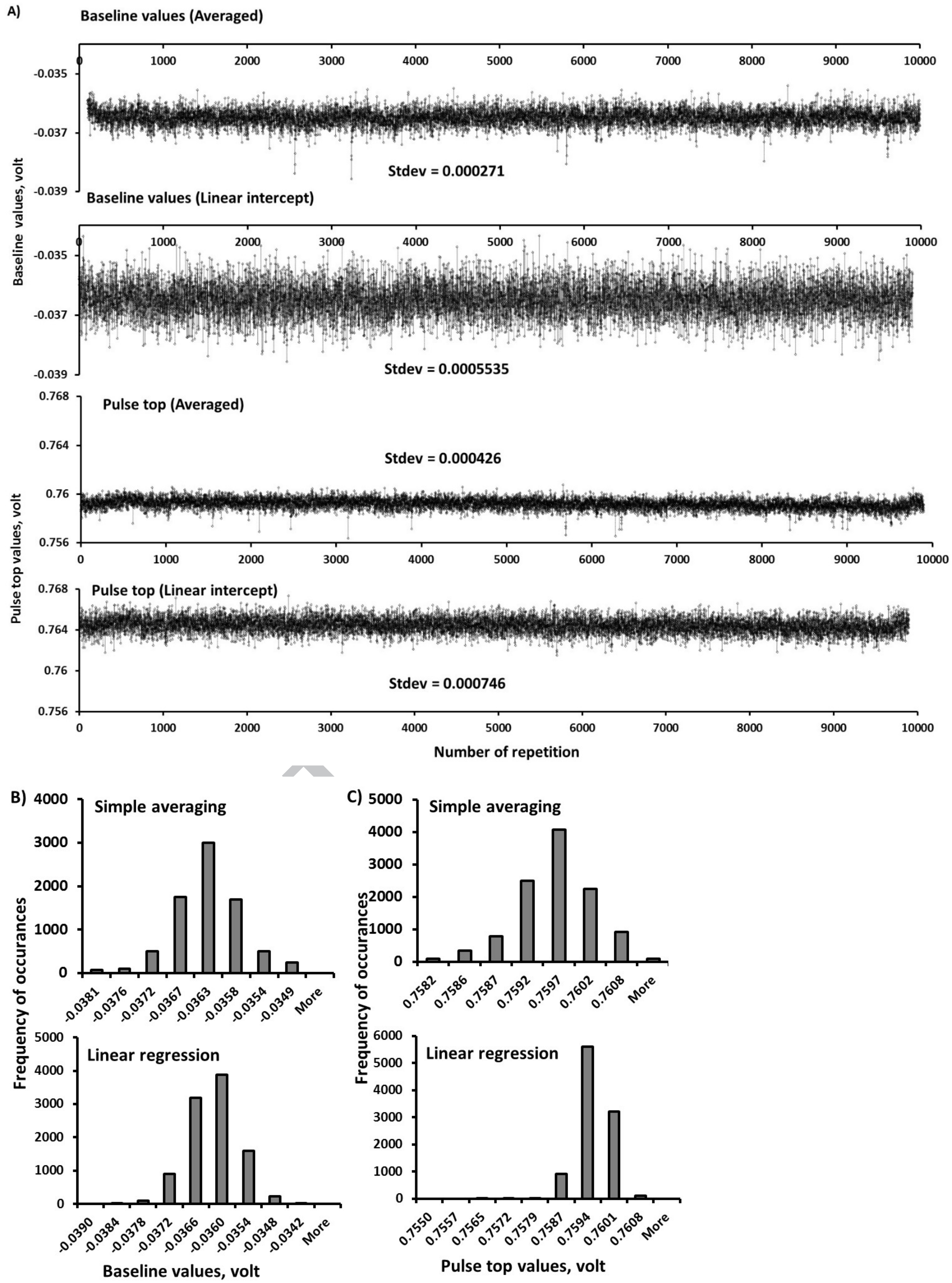


Fig S 3 **A)** Pulse top and baseline signal values of respective optical voltage pulses, Constructed histogram **B)** pulse top data values **C)** Baseline data values obtained from 10,000 consecutive raw pulses after applying two different data evaluation statistical methods simple averaging and linear regression.

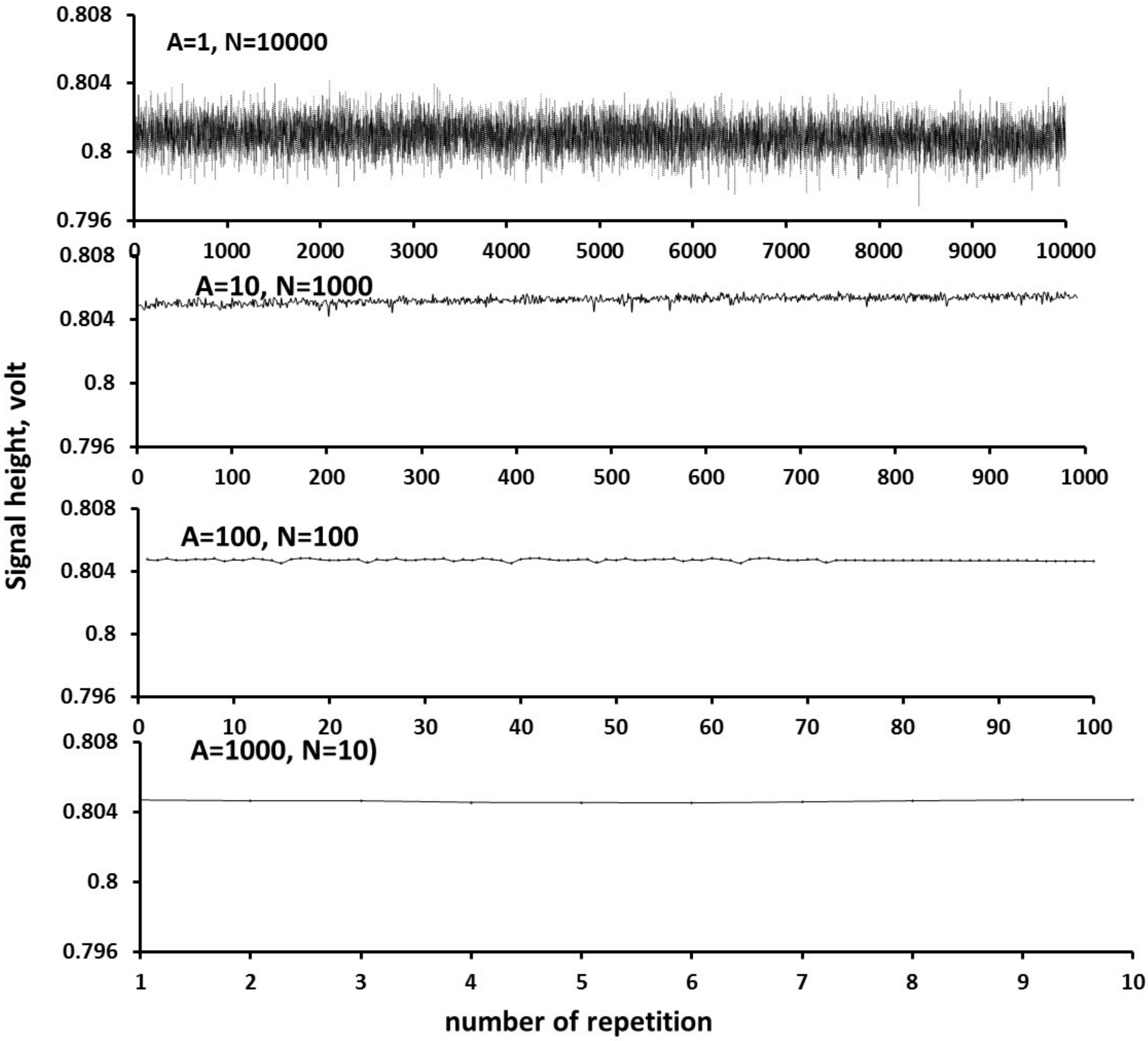


Fig. S 4 Analytical signal from each digitally processed signal pulses of processed data point as a quasi-continuous data stream using simple averaging applied on of the pulse top (125 data points) and base line (150 data points).

3. Command line parameters and code in C language

Command line parameters for the Red Pitaya peakshape program (note that lower case letters in the commands below refer to a number chosen by the user):

t,v Time (t) and voltage (v) pair. Time is in 0.01 micro seconds or steps, Voltage is in volts. Each pair specifies the next step in generating the voltage graph to be sent from the red pitaya, taking t steps to get from current voltage to desired voltage. For example if the graph is currently at 0.4V and the parameter given is 4,0.6 then the next four steps in the graph will be 0.45V 0.50V 0.55V and 0.6V.

N=n Specifies how many samples to process. The pump will be turned on before each sample. The value of n must be a whole number greater than or equal to 1. A value less than 1 will be replaced by 1. If it is not specified the default is 10. If N=n is specified more than once then only the last one on the command line is used.

S=n Specifies how many sub groups of pulses are acquired for each sample. The pump will NOT be turned on between subgroups. If S=n is specified more than once then only the last one on the command line is used. If not specified, the default is 1, ie not sub group analysis.

A=n Specifies how many pulse are generated and acquired for each sub group which are averaged together. The total number of pulses generated / acquired and then averaged together for each sample is value of S x value of A. If A=n is specified more than once then only the last one on the command line is used.

P:s,t Specifies the speed (s) and time (t) that the pump will operate between the analysis of each sample. The value of s must be between 0.5 and 1.8, if the value is less than 0.5 the pump will not turn on, if the value is greater than 1.8 the pump will turn on to 1.8. The second parameter is the time in milliseconds to turn the pump on. Example if P:0.9,4000 is specified the pump will be provided with 0.9 volts for 4 seconds between each sample. The pump cannot be turned on for less than 500 milliseconds and any value less than 500 will be taken as 500. Example if P:1.1,0 is specified the pump will turn on the 500 milliseconds. You cannot turn the pump off by specifying no time, you must specify no voltage or not include the parameter. If this parameter is not present, the default is not to turn the pump on. . If P:s,t is specified more than once then only the last one on the command line is used.

R=Y Record the time when the pump is turned on and also turned off.

C:s,f Specifies a calculation zone within the recorded averaged pulse. The calculation preformed and a simple average, linear regression, polynomial regression, boxcar smoothing with simple average, linear regression and polynomial regression along with s-k smoothing with simple average, linear regression and polynomial regression. Up to 10 calculation zones can be specified.

B:n Specifies how many points to average together in the box car analysis.

V:s,f:v Not yet implemented.

D=N Specifies to only read and process fast analogue input channel 1 rather than both channel 1 and channel 2.

F=Y Specifies that the input signals should be collected at 125 MHz instead of the default 15.625 MHz.

Z=N Specifies to not zero the calculation array between samples. This parameter should rarely be used if ever.

O=N Specifies that the sequence of averaged pulses should NOT be written to the output.

O=n Specifies how many steps of each averaged pulse should be output, default if not specified is 1000. If O=n is specified more than once then only the last one on the command line is used.

T=Y Specifies that timing information should be provided to the user on stderr The information produced is how long is spent in each routine on every call, which can be used to improve the program performance or determine how long a particular run is going to take. This parameter should rarely be used.

W=Y Indicates that the program should wait for a signal (button being pushed) before collecting and analysing each sample.

W=S Indicates that the program should initially wait for a signal (button being pushed) before starting to collect samples.

4. Program flow chart

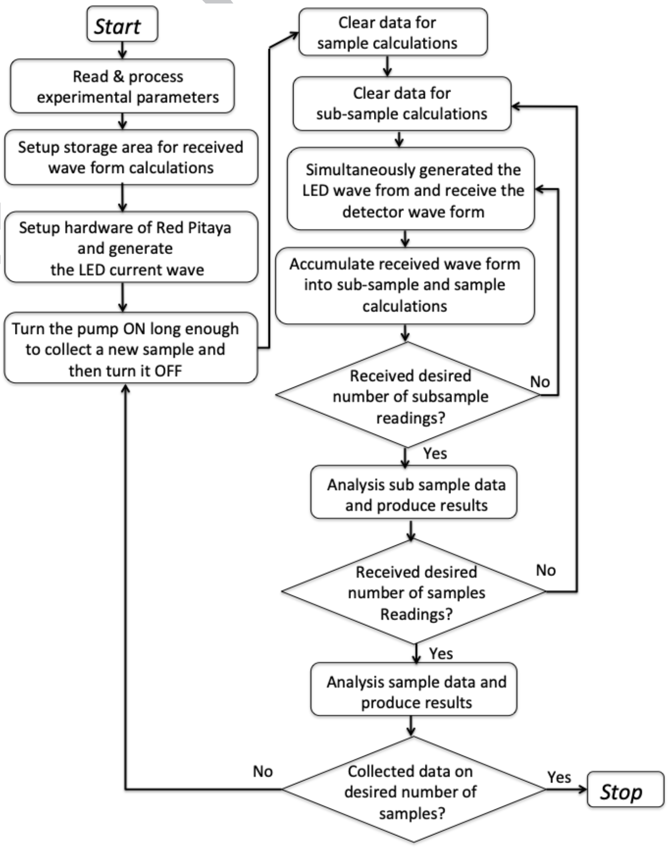


Fig. S 5 Program flowchart of the corresponding algorithm.

5. Program code: Red Pitaya function for generating LED pulse and acquiring signal from detector

```
int GenerateAcquire ( int   DualChannel,
                      int   FastRate,
                      float in1[ ],
                      float in2[ ] )
  {                                                                     //* This routine generates the output wave and acquires the responses.
                                                                        //* Prerequisite: The wave form has been loaded into the FPGA using
                                                                        //* rp_GenArbWaveform and the frequency set with rp_GenFreq.
                                                                        //* The array in1[ ] will be filled with the detected wave form.
    uint32_t buff_size = MaxCalculation ;
    int      TriggerWait ;

    if ( FastRate )
      { rp_AcqSetDecimation( RP_DEC_1 ) ; }
    else
      { rp_AcqSetDecimation( RP_DEC_8 ) ; }
    rp_AcqSetTriggerLevel( 0 ) ;                                        //* Set FPGA trigger.
    rp_AcqSetTriggerDelay( 8192 ) ;                                     //* Set the FPGA acquisition trigger delay.
    rp_AcqStart( ) ;                                                    //* Tell the FPGA to start an acquisition.
    rp_AcqSetTriggerSrc( RP_TRIG_SRC_EXT_PE ) ;                         //* Set the acquisition trigger to be external signal.

    rp_GenOutEnable( RP_CH_1 ) ;                                        //* Turn the signal generation on for channel 1.
    rp_GenTriggerSource( RP_CH_1, RP_GEN_TRIG_SRC_EXT_PE ) ;            //* Set the generation trigger to be external signal.

    rp_acq_trig_state_t state = RP_TRIG_STATE_WAITING ;
    TriggerWait = 0 ;
    while (( state != RP_TRIG_STATE_TRIGGERED ) && ( TriggerWait < 20 ))
      { rp_DpinSetState( RP_DIO1_P, RP_HIGH ) ;                         //* 5 usec pulse on digital I/O pin connected to external trigger.
        usleep  ( 5 ) ;
        rp_DpinSetState( RP_DIO1_P, RP_LOW ) ;
        usleep  ( 5 ) ;
        TriggerWait++ ;
        rp_AcqGetTriggerState( &state ) ;                               //* Get the acquisition trigger state from the FPGA.
      }

    if ( state != RP_TRIG_STATE_TRIGGERED )
      { fprintf( stderr, "******** Trigger failed : state is %d\n", state ) ;
        return( false ) ;
      }
    else
      { usleep( 15 ) ;                                                  //* Sleep long enough for the buffer to be filled.
        rp_AcqGetOldestDataV( RP_CH_1, &buff_size, in1 ) ;              //* Transfer the acquired detector values from the FPGA.
        if  ( DualChannel )
          { rp_AcqGetOldestDataV( RP_CH_2, &buff_size, in2 ) ; }        //* Transfer the acquired current feedback values from the FPGA.
        rp_GenOutDisable( RP_CH_1 ) ;
        return( true ) ;
      }
  }
```

Full detailed program code which is multiple pages long is available from the authors upon request.

Research Highlights

- Flexible data processing with field programmable gate array (FPGA) incorporated with portable open source microcontroller
- Rapid pulse generation of 2 µs short pulses with a typical repetition rate of 1 kHz to drive the IR LED
- Optical sensing, data acquisition and processing on-the-fly of 2 Gbit/s datastream
- Continuous analytical signal of 1 point every 1 ms to 10 s obtained
- Minimum measurable absorbance of 10^{-4} a.u. achieved