Computers & Geosciences 94 (2016) 1–10

Contents lists available at ScienceDirect

Computers & Geosciences

journal homepage: www.elsevier.com/locate/cageo

Research paper

Development and programming of Geophonino: A low cost Arduino-based seismic recorder for vertical geophones

J.L. Soler-Llorens [a,*], J.J. Galiana-Merino [b,c], J. Giner-Caturla [a,b], P. Jauregui-Eslava [a], S. Rosa-Cintas [a], J. Rosa-Herranz [b,c]

[a] Dpto. Earth Sciences and Enviroment, University of Alicante, P.O. Box 99, E-03080 Alicante, Spain
[b] Dpto. Physics, Systems Engineering and Signal Theory, University of Alicante, Spain
[c] University Institute of Physics Applied to Sciences and Technologies, University of Alicante, Spain

ARTICLE INFO

Article history:
Received 2 December 2015
Received in revised form
11 May 2016
Accepted 30 May 2016
Available online 2 June 2016

Keywords:
Seismic recorder
Vertical geophone
Arduino Due
Instrumentation amplifier
Anti-aliasing filter

ABSTRACT

The commercial data acquisition systems used for seismic exploration are usually expensive equipment. In this work, a low cost data acquisition system (Geophonino) has been developed for recording seismic signals from a vertical geophone. The signal goes first through an instrumentation amplifier, INA155, which is suitable for low amplitude signals like the seismic noise, and an anti-aliasing filter based on the MAX7404 switched-capacitor filter. After that, the amplified and filtered signal is digitized and processed by Arduino Due and registered in an SD memory card. Geophonino is configured for continuous registering, where the sampling frequency, the amplitude gain and the registering time are user-defined. The complete prototype is an open source and open hardware system. It has been tested by comparing the registered signals with the ones obtained through different commercial data recording systems and different kind of geophones. The obtained results show good correlation between the tested measurements, presenting Geophonino as a low-cost alternative system for seismic data recording.

1. Introduction

Technology grows faster day after day. That makes good equipment become old-fashioned very soon. This issue also affects the equipment used in active seismic exploration. A few decades ago, the commercial seismic exploration seismographs were controlled by simple programs that work under old operating systems (e.g. MS-DOS, Windows 98, Windows XP, etc). Moreover, the data transfer used to be carried out through the RS232 serial port, or even using old HD 1.44 MB disks. Although they are very good quality equipment and they continue working nowadays, its current use becomes very difficult, due to the lack of this technology in the new computers (e.g. operating systems, RS232 serial port, etc). Moreover, due to their storage memory limitations, they are not suited to manage the huge volume of data required by recent seismic experiments, like the ambient noise measurements.

Thus, an updating of the data acquisition equipment is required, by means of new modules of hardware (if it is possible) or new complete systems. In this sense, one of the biggest drawbacks is the high price of these seismic exploration systems. Although it is not necessary to obtain new geophones, the data acquisition and recording equipment is very expensive by its own. Due to such economic constraints, it is not possible for all research groups and universities to maintain modern seismic exploration equipment for only education purposes.

On the other side, laboratories around the world face the need to build custom-made experimental systems to acquire data through sensors. In this way, they can configure the data acquisition system according to their needs.

In the recent years, some research groups have tried to develop their own equipment. For instance, we can find the work of Picozzi et al. (2010), where they developed a dedicated system for seismic arrays, and the proposal of Khan et al. (2012), where the PC sound card is used to digitize seismic signals for educational purposes.

Concerning this issue, the increasing accessibility to microcontrollers and to other electronic components have helped to develop home-made systems at low prices. In this way, one of the most common electronic devices used for electronic prototyping in the recent years is Arduino. Arduino constitutes an open-source electronic platform that allows monitoring and controlling different analog and digital signals, as well as other specific electronic circuits (Koenka et al., 2014). Many studies carried out on this topic (e.g. Savazzi, 2011; Agudo et al., 2014; Koenka et al., 2014; Fuentes et al., 2014) show the suitability of the Arduino platform to solve

* Corresponding author.
E-mail addresses: jl.soler@ua.es (J.L. Soler-Llorens), juanjo@dfists.ua.es (J.J. Galiana-Merino), jj.giner@ua.es (J. Giner-Caturla), pedro.jauregui@ua.es (P. Jauregui-Eslava), sergio.rosacintas@ua.es (S. Rosa-Cintas), julio.rosaherranz@ua.es (J. Rosa-Herranz).

http://dx.doi.org/10.1016/j.cageo.2016.05.014
0098-3004/

specific needs in different research areas.

In this work we have developed a low-cost Arduino-based seismic recorder for vertical geophones. The system is controlled through a user interface developed ad hoc, which includes the parameters needed for the data acquisition and recording. The equipment has been tested using sine waves of known characteristics and compared with other commercial equipments: the DT321 data acquisition card and the RAS-24 exploration seismograph. These systems are well known and have been widely used in previous research projects (e.g. Giner et al., 2012; Rosa-Cintas et al., 2013).

The obtained results demonstrate the suitability of the developed system to acquire and record seismic signals, with the same reliability as the commercial equipment compared in this study. Thus, the presented system provides a low-cost alternative to the commercial systems. Moreover, this is an open-source system, which allows other users to accomplish hardware and software changes, in order to adapt the proposed prototype to their particular requirements. It is appropriate for different educational and research seismic experiments, including the seismic monitoring for long periods of time (e.g. ambient noise measurements).

The article is organized as follows. In Section 2, the different components and equipment used in the present study are described. In Sections 3 and 4, the developed hardware design and the associated software are explained in detail. Finally, in Section 5, different test experiments and their results are shown.

2. Technology overview

The developed prototype is basically formed by four components: an instrumentation amplifier, an anti-aliasing filter, an Arduino board microcontroller, and a SD card shield. Two commercial systems, the RAS-24 exploration seismograph (Seistronix Inc., 2005) and the DT321 data acquisition card (Data Translation Inc., 2010), together with two different types of geophones, have been used to evaluate the prototype.

2.1. Arduino Due microcontroller

Arduino Due is an inexpensive multipurpose open-source hardware platform based on the Atmel SAM3 $\times$ 8E ARM Cortex-M3 CPU. It is the first Arduino board based on a 32-bit ARM core microcontroller. The Arduino Due can be programmed using the C++ language. Moreover, a lot of libraries are available to connect the main board with other systems and extend its functionality. The open-source nature of the platform and its huge user community provide access to a large collection of software code; making it relatively simple to build new prototypes, even with little previous programming experience. The Arduino Due is equipped with 68 inputs/outputs and can be programmed to work with different sampling frequencies. The analog-to-digital converter (ADC) integrated in the Arduino Due board provides a maximum resolution of 12 bits and a dynamic range from ground to the ADVREF pin voltage, which is connected to +3.3 V by default (Atmel Corporation, 2012). Its 32-bit 85 MHz processor with 96 Kb of memory provides enough computing power for real-time data processing. The Arduino Due supports multiple communication protocols, such as 2 USB-ports and 4 UARTs (hardware serial ports), which can be used to communicate with other instruments (Ruytenberg et al., 2014). Additionally, the modularity of Arduino boards is guaranteed through a wide variety of modules (shields) that extend the functionality of the system and adapt it to the desired requirements. Besides, we have the possibility to develop new modules with specific purposes (e.g. Fuentes et al., 2014).

Ardunio Due was selected from the existing range of Arduino boards due to the following reasons: 1) It presents the higher frequency clock from the available Arduino boards at the present moment, i.e. 84 MHz, which allows a theoretical maximum sampling frequency of 1 MHz, as it is pointed out in the application note published by Atmel Corporation. (2011). 2) It has two analog outputs, what allows controlling different types of chips by software, e.g. voltage controlled gain amplifiers. 3) Finally, this board is implemented with a 512 Kb Flash memory and 96 Kb SRAM memory, which ensures enough memory to store larger buffer data during the acquisition. Its characteristics allow implementing a highly scalable system, in terms of the type and number of sensors and in terms of the software development.

2.2. Geophones and data acquisition equipment (Testing equipment)

Two types of geophones have been used to test the performance of the developed prototype. One of the sensors used, the SN4-10V, is an electromagnetic 10 Hz vertical-geophone, which is commonly used for seismic prospecting. The other sensor used in this work is a Mark L4-C. This geophone presents, according to the manufacturer, a sensitivity of 280 V/m/s, a mass of 1 Kg, a natural frequency of 1 Hz, and a coil resistance of 5500 Ω.

In order to test the workability of the designed device, we have compared the recorded data with the ones obtained through two commercial data acquisition systems. First one is the RAS-24 system, an excellent and contrasted equipment, specially used for refraction and reflection surveys. However, for other kind of experiments, based on seismic noise recordings, it presents some drawbacks. The available RAS-24 system uses the RS-232 port for the data transfer with the computer, which implies a very low-speed connection.

Moreover, it has a limited buffer for data acquisition that limits the maximum recording time to 64 s, for the minimum sample rate of 250 Hz. It is not a problem for refraction or reflection seismic data acquisition but it becomes a serious handicap when we want to obtain larger registers, as it is the case of the seismic noise recordings.

The other commercial data acquisition system employed with the geophones is the DT321 card, which is a PCI data acquisition card that can be connected to any computer tower and used for different general purposes. It provides a 16-bit analog-to-digital conversion. The data acquisition and recording is controlled through the Data Acquisition Toolbox developed for Matlab.

3. Hardware implementation

The hardware scheme of the developed prototype is shown in Fig. 1. Basically, it is composed of four main blocks: 1) the instrumentation amplifier (i.e. the INA155 chip); 2) the anti-aliasing filter (i.e. the MAX7404 chip); 3) the Arduino Due, which controls the acquisition, processing and recording processes and; 4) the SD (Secure Digital) card shield that stores the seismic data. One important characteristic of these four blocks is that they can all work with the same single power supply, i.e. 3.3 V. This allows the use of a common battery for all blocks and makes the prototype suitable for portable applications. The wiring scheme of the system is shown in Fig. 2.

In the first stage, an instrumentation amplifier is extremely recommended for the proposed prototype. The differential signal provided by the geophone is directly connected to the amplifier, which provides an amplified single-mode output signal, referenced to the ground of the complete system. In addition, the amplification stage allows adapting the amplitude of the signal to cover the dynamic range of the analog-to-digital converter of the Arduino Due, i.e. the 3.3 V.

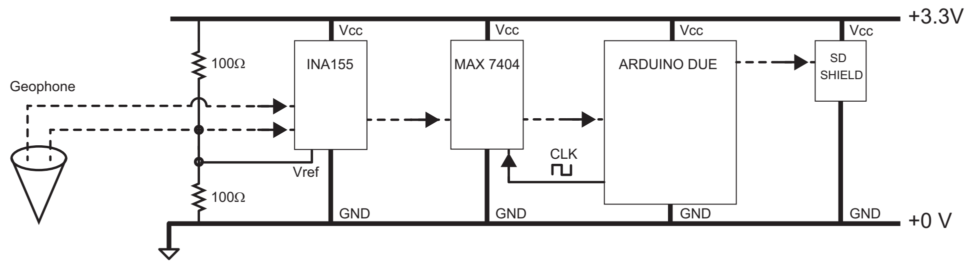

Fig. 1. Hardware scheme.

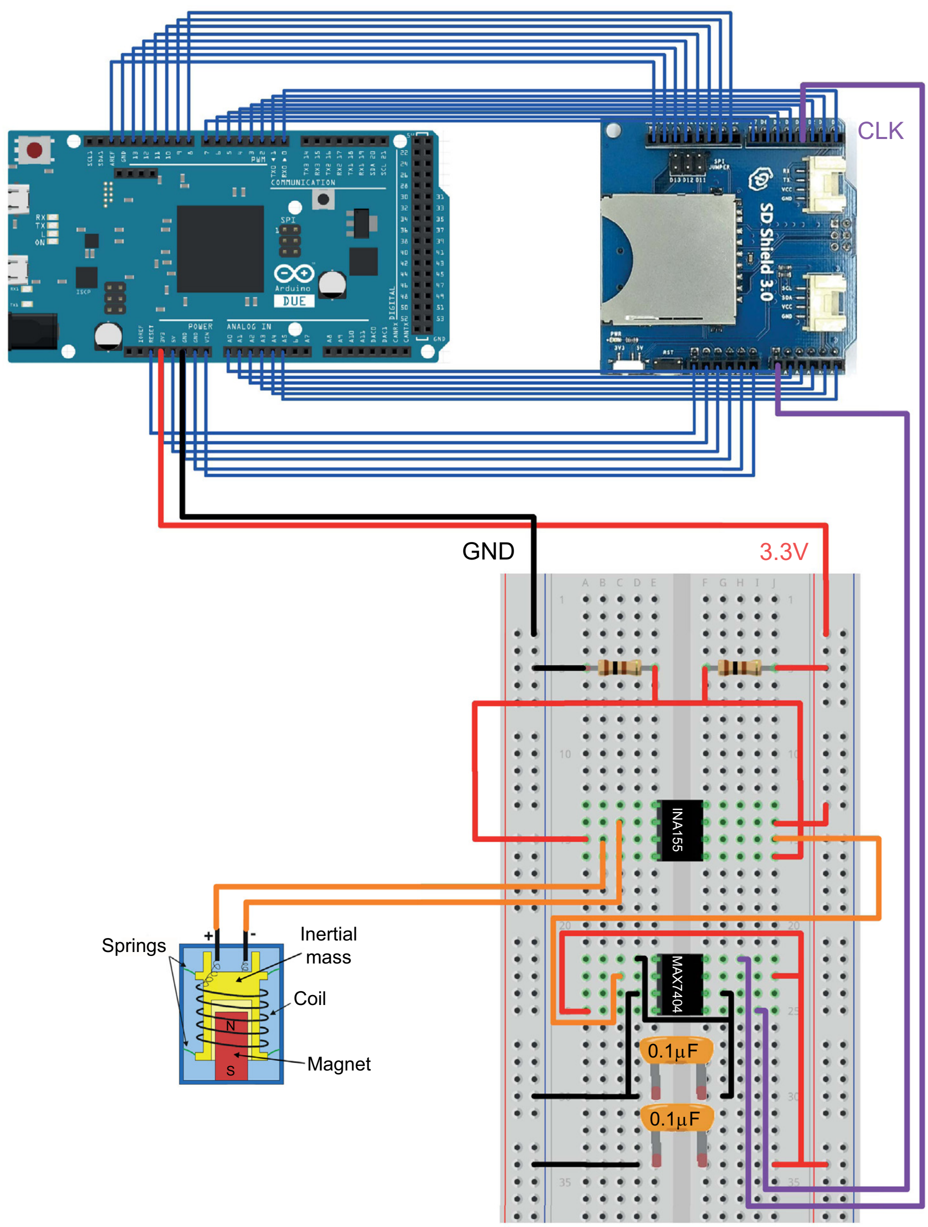

Fig. 2. Wiring scheme.

In this case, we have selected the INA155 amplifier (Burr Brown Corp., 2000), basically due to the following characteristics: 1) It is a low-cost CMOS instrumentation amplifier with rail-to-rail output swing optimized for low voltages, as it is the case of the registered seismic noise. 2) It operates with a single power supply between 2.7 V and 5.5 V, thus it can be supplied through the 3.3 V pin of the Arduino Due. 3) The voltage gain is set with the addition of a single resistor, R_G, connected between pins 1 and 8; obtaining a minimum gain of 10 V/V, when $R_G \rightarrow \infty$ (open circuit), and a maximum gain of 50 V/V, when $R_G = 0$ (short circuit).

We have used a voltage divider, formed with two resistors of 100 Ω, to bias the input voltage to the half of the power supply, i.e. 1.65 V. This bias voltage is also used as a reference voltage for the INA155.

The next stage consists of an anti-aliasing filter that is located previous to the ADC of the Arduino Due. For this task, we have used the MAX7404 integrated circuit (https://datasheets.maximintegrated.com/en/ds/MAX7400-MAX7407.pdf), which is an 8th-order, lowpass, elliptic, switched-capacitor filter. We have chosen this chip due to the following characteristics: 1) it operates

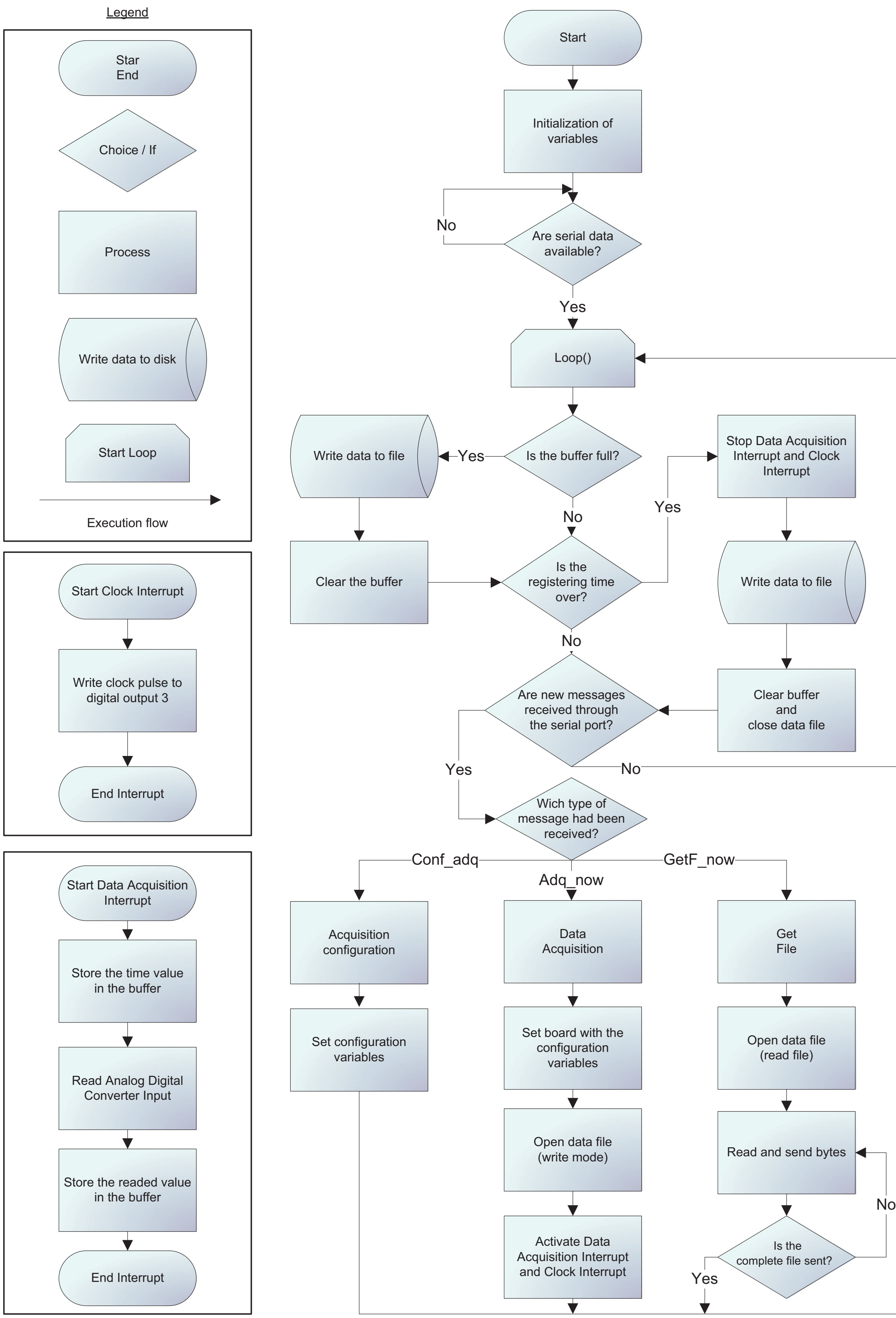

Fig. 3. Program flowchart of Geophonino.

with a single power supply, between 2.7 V and 3.6 V, which allows the supplying through the 3.3 V pin of the Arduino Due and; 2) the corner frequency can be tuned from an external clock, thus it can be controlled by a clock signal provided through the Arduino Due.

The MAX7404 filter permits a corner frequency between 1 Hz and 10 kHz, which is suitable for the seismic experiments. The specific frequency is controlled through an external clock signal, with a clock-to-corner ratio of 100:1. It means that a corner frequency of 10 Hz can be obtained by using a clock signal of 1 kHz (100 times the corner frequency). In our case, the appropriated clock signal is provided through one pin of the Arduino Due.

In the third stage, the single amplified and filtered signal is connected to the ADC of the Arduino Due. In this case, the input pin has to be set up in single-ended mode. The Arduino Due controls the data acquisition (sampling frequency, amplification, corner frequency and registering time) and the data recording.

Finally, an SD (Secure Digital) card shield V.3.0, developed by Seed Technology Inc., has been connected to the Arduino Due. This shield allows storing data easily into different kind of storage devices, e.g. SD card, Micro SD card or SDHC (Secure Digital High Capacity). The SD card shield can operate at 3.3 V or 5 V. Thus, in this case, it has been setup at 3.3 V, which is also the voltage required by Arduino Due.

4. Software implementation

The Arduino microcontroller has been programmed in the Arduino programming language, which is based on Wiring, an open-source framework for microcontrollers programming.

The user interface has been developed using Processing, an open-source language/development tool available on the Arduino web site (http://playground.arduino.cc/Interfacing/Processing).

4.1. Geophonino sketch

The sketch (i.e. the name that Arduino uses for a program) has been compiled with the Arduino 1.5.6-r2 version and stored as a single file, named 'geophonino.ino'. In Fig. 3, the program flowchart is shown.

The sketch is divided in two parts:

1) Setup block: This part of the code is executed once, at the beginning of the program, to initialize the required variables. After that, the program remains waiting to receive any data from the serial port, which constitutes the communication link between the computer and the user interface developed for Geophonino. Thus, the acquisition parameters, as well as other kind of instructions, can be sent to Arduino through the user interface. The serial port has been configured to the maximum available baud rate, i.e. 115,200 bauds, to save time in the data transfer between Arduino and the computer. Once any data is received from the serial port, the setup block ends and the sketch continues with the second block.
2) Loop block: This block is executed continuously in a loop and handles the data acquisition.

For each cycle of the loop, the sketch tests the three following points:

a) The number of data stored in the buffer: If the buffer is full, then the data will be written to a file and the buffer will be cleared.
b) The data acquisition time: If the recording time reaches the time established by the user, then data acquisition will end. The data contained in the buffer will be written in the file and the file will be closed. Finally, the buffer will be cleared.
c) The messages received through the serial port: If any message is received from the serial port, the sketch will carry out the action accordingly. Three different messages have been configured for communication between the user interface and the Arduino. These messages are identified as *Conf_adq*, *Adq_now* and *GetF_now*.
   _ **Conf_adq** contains the acquisition parameters, which are designated as:
      nomF: assigns the name of the file where the data will be stored.
      overW: establishes whether the file will be overwritten or not.
      Gain: defines the gain used by the Arduino board.
      tR: establishes the length of the record (in seconds).
      sR: defines the sampling rate according to the four possible values established by the interface (i.e. 1→100 Hz; 2→250 Hz; 3→500 Hz, and 4→1000 Hz).

It uses the symbol '&' as a separator between parameters and the symbol '=' to assign the values to the corresponding parameters. Hence, an example of the structure of this type of message is:

"*conf_adq&nomF=name &overW=N&Gain=2&tR=15&sR=1*"

– **Adq_now** starts the data acquisition.
– **GetF_now** requests the data file transfer from Arduino to the computer. This action can only be carried out when the data acquisition has ended.
  The instructions in the setup or loop blocks are executed in order of appearance. However, there are some instructions that have to be executed in a precise moment, e.g. the data acquisition at the times determined by the sampling frequency. The execution of this kind of instructions is carried out by interrupts. Arduino has two types of interrupts:
– **Hardware interrupts**: They occur when an external device sends a signal to the Arduino processor changing the logic state on some specific input pin.
– **Software interrupts**: They take place because some instruction sends a request, as for example, the timer clock.

The handler interrupt, also known as Interrupt Service Routine (ISR), is a callback function with the code to execute when the interruption is triggered. Thus, when the trigger occurs, the code in execution is stopped and the program jumps to the ISR. While the ISR is being executed, any other request is ignored. When the ISR ends, the execution returns to the main code, continuing at the same point where it was stopped. Therefore, the ISR should be as short and fast as possible, because the main program is blocked during its execution.

If any variable of the ISR is also used by the setup or loop blocks, then it must be declared as volatile, in order to assure that the variable is constantly updated. In our case, the ISR records the acquired data in a volatile variable (the buffer). When the buffer is full, the loop block stores the complete buffer in a file and empty the buffer. If the variable associated to the buffer was not volatile, then the recorded file could contain erroneous values, because the buffer might not be updated yet.

The data acquisition is carried out by means of a timer interruption, which allows executing the ISR at the times required by the user, i.e. the selected sampling frequency. Its configuration requires only the appropriate modification of the counter register associate to the internal clock. Arduino Due has nine general purpose 32-bit timer/counters. In order to simplify the configuration and use of these interrupts, we have used the DueTimer library developed by Seidel et al. (2015).

We have also used another timer interruption to generate the clock signal required to control the anti-aliasing filter. Therefore, in this case, the ISR provides alternating high and low levels through an output digital pin of the Arduino Due. For the four sampling frequencies available through the user interface (100 Hz, 250 Hz, 500 Hz and 1000 Hz), we have set up the corner frequencies to 40 Hz, 100 Hz, 200 Hz, and 400 Hz, respectively, which corresponds accordingly to clock signal frequencies of 4 kHz, 10 kHz, 20 kHz and 40 kHz.

4.2. Acquisitions settings

The ADC integrated in the Arduino board allows two different resolution modes (with 10 or 12 bits), which can be selected through the command: "*analogReadResolution(Number of bits);*". For the present prototype, we have selected the maximum available resolution (i.e. 12-bits), which provides 4096 (2^{12}) voltage levels. Considering the dynamic range of 3.3 V, each voltage level corresponds to 0.806 mV.

Arduino Due supports two input operation modes: single-ended and fully differential. The ADC is set up in single-ended mode by default, but this mode can be configured independently for each channel. In order to apply different configurations to each channel, the ANACH value in the ADC_MR register must be first set to 1. After that, the ADC Channel Offset Register (ADC_COR) must be modified as it is shown in the SAM3 × 8E datasheet (Atmel Corporation, 2012). In Fig. 4, the ADC_COR register values are shown.

Thus, if we want to connect directly the geophone to the Arduino board, then we will have to configure one input pin, e.g. A0, in differential mode. In this way, the differential signal provided by the geophone will be connected to the inputs A0–A1. However, if we use the differential amplifier INA155 to improve the amplitude of the signal provided by the geophone, then we will have to configure one input pin, e.g. A0, in single-ended mode. Therefore, the amplified and filtered output signal (pin 5, MAX7404) will be connected to the input pin A0 (Fig. 2).

4.3. Graphical user interface

The graphical user interface (GUI) has been developed with Processing and the source code has been attached to the file GuiGeophonino.pde. Processing has been selected because it is an open-source language/development tool that can be used in many operating systems such as Mac OS X, Windows, Linux or Android. Furthermore, Processing has a lot of open-source libraries available and a huge amount of documentation that are very helpful to learn how to use it.

The Geophonino GUI has been developed using the open-source library, ControlP5 (Schlegel, 2015), because it has implemented a wide range of controls, as well as many examples and tutorials that help programmers learning quickly and efficiently about its use.

The developed GUI (Fig. 5) is divided in three blocks or steps that guide the user through the acquisition configuration and the recording process as given below.

1° Configure Connection. In this block, the Arduino COM port connection is selected. The '*Refresh port list*' button allows updating the available serial ports.

2° Configure Data Acquisition. The different parameters associated with the acquisition process are configured in this step. These parameters are: data file name (maximum 8 characters long, due to the FAT16 file system), duration (in seconds), sampling rate and Arduino Due amplification. The available sampling rate values are 100 Hz, 250 Hz, 500 Hz, 1000 Hz, which are the most common values used in seismic exploration. However, other values might be also possible. Respecting the Arduino Due amplification, it provides three different amplification values: 0.5, 1 and 2 (in

44.7.17 ADC Channel Offset Register

Name: ADC_COR

Address: 0x400C004C

Access: Read-write

31	30	29	28	27	26	25	24
DIFF15	DIFF14	DIFF13	DIFF12	DIFF11	DIFF10	DIFF9	DIFF8
23	22	21	20	19	18	17	16
DIFF7	DIFF6	DIFF5	DIFF4	DIFF3	DIFF2	DIFF1	DIFF0
15	14	13	12	11	10	9	8
OFF15	OFF14	OFF13	OFF12	OFF11	OFF10	OFF9	OFF8
7	6	5	4	3	2	1	0
OFF7	OFF6	OFF5	OFF4	OFF3	OFF2	OFF1	OFF0

This register can only be written if the WPEN bit is cleared in "ADC Write Protect Mode Register" on page 1365.

- **OFFx: Offset for channel x**

0 = No Offset.

1 = center the analog signal on Vrefin/2 before the gain scaling. The Offset applied is: (G-1)Vrefin/2

where G is the gain applied (see description of ADC_CGR register).

- **DIFFx: Differential inputs for channel x**

0 = Single Ended Mode.

1 = Fully Differential Mode.

Fig. 4. ADC_COR register values (Atmel Corporation, 2012).

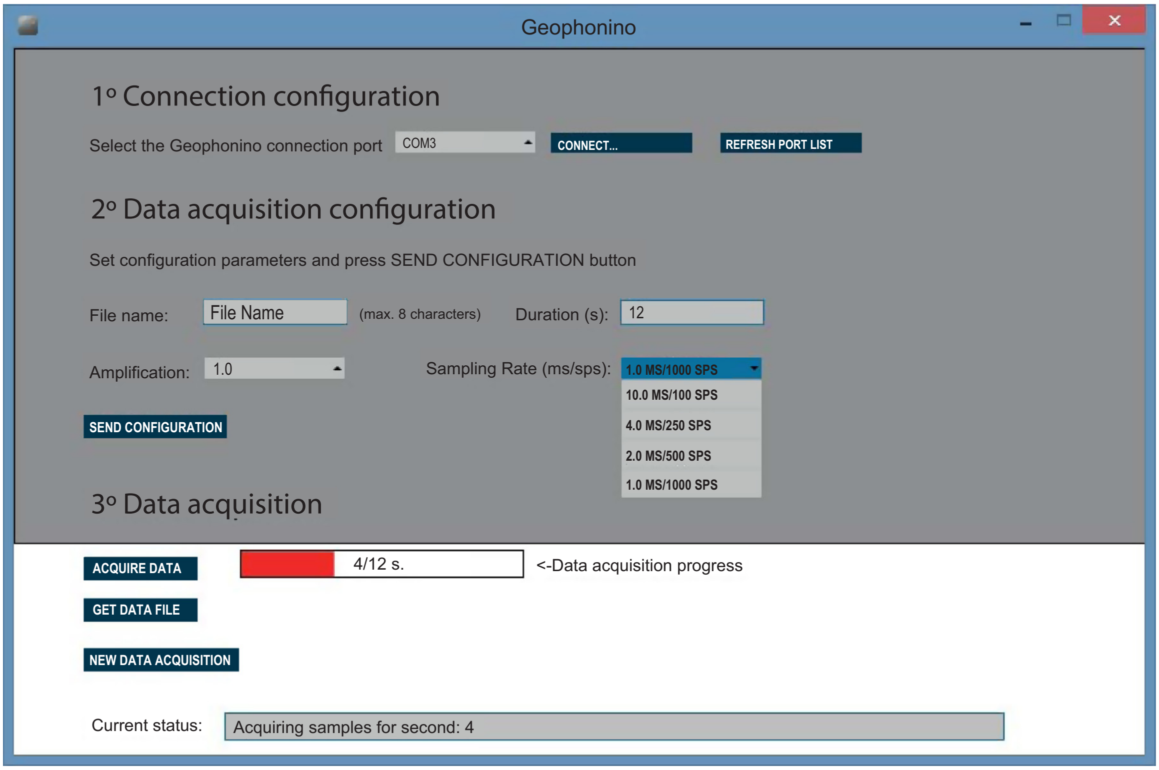

Fig. 5. Screenshot of the graphical user interface.

differential mode) or 1, 2 and 4 (in single-ended mode).

3° Data acquisition. In this step, the acquisition and recording processes are executed. This block includes the '*Acquire Data*' and '*Get Data File*' buttons, which execute the data acquisition and the data transfer to the computer, respectively.

The data are stored in a plain text format. The first line contains the acquisition parameters (time, amplitude and sampling rate), separated by commas. The following lines contain the recording time in milliseconds (first column) and the corresponding amplitude of the recorded signal (second column).

The data are only stored in the SD card by default. Nevertheless, the user can send the resulting data file to the computer at the end of the acquisition through the '*Get Data File*' button. When the recorded file is very big, the data transfer through the serial port can take very long. Therefore, in this case, it could be more convenient to store the data only in the SD card, as this type of card can be directly read in most of the computers.

5. Experiments and analysis of the results

5.1. Geophonino testing

In order to test the performance of Geophonino, we have first evaluated the developed prototype by using a known sine wave as input signal. The Tektronix AFG310 function generator has been used to provide a sine wave of amplitude 0.1 V and different frequency values, in the range 1–15 Hz. We have used small amplitudes and frequencies to simulate signals of similar characteristics to the possible real ones. The analyzed frequency range covers the expected frequency bandwidth of local seismic events. After that, the provided signal has been amplified by the instrumentation amplifier INA155 and acquired with the Arduino Due. The selected gains for the INA155 and the Arduino Due have been 20 dB (10 ×) and 0 dB (1 ×), respectively. The sampling frequency has been selected as 1 kHz, and the recording duration has been set up to 10 s.

For this experiment, the input and the amplified signals have been acquired and recorded with the Arduino Due simultaneously (Fig. 6). We have checked that both recorded signals are identical in shape and the only difference found is related with the amplitude, which is ten times higher for the amplified signal. We have also studied the frequency response of the proposed prototype by doing a frequency sweep in the typical frequency range of the seismic signals. The results have shown that the voltage gain remains constant for the frequency band of 1–15 Hz, as it was expected from the characteristics indicated in the INA155 datasheet.

5.2. Comparison results between Geophonino and the DT321 data acquisition card

For the second test, we have used a calibration pulse and we have compared the data recorded by Geophonino with the ones recorded by the commercial data acquisition card, DT321. In this case, the input signal has been provided through the 1-Hz Mark L4-C sensor, which includes a calibration coil that can be excited by an external current (Bowden, 2003). The application of a specific current causes the displacement of the main mass from its equilibrium point and swing around another position. When this current is suddenly disconnected, the main mass returns to the equilibrium point after some oscillations around it, providing the corresponding calibration pulse. In this way, the application of the same current assures the repetitiveness of the calibration pulse. In this case, we have applied a voltage of 0.6 V for our experiments.

Another important characteristic of the L4-C sensor is that it provides two differential outputs, which makes it suitable for our comparison purposes. Thus, both outputs have been connected simultaneously to the Geophonino and the DT321 card in order to

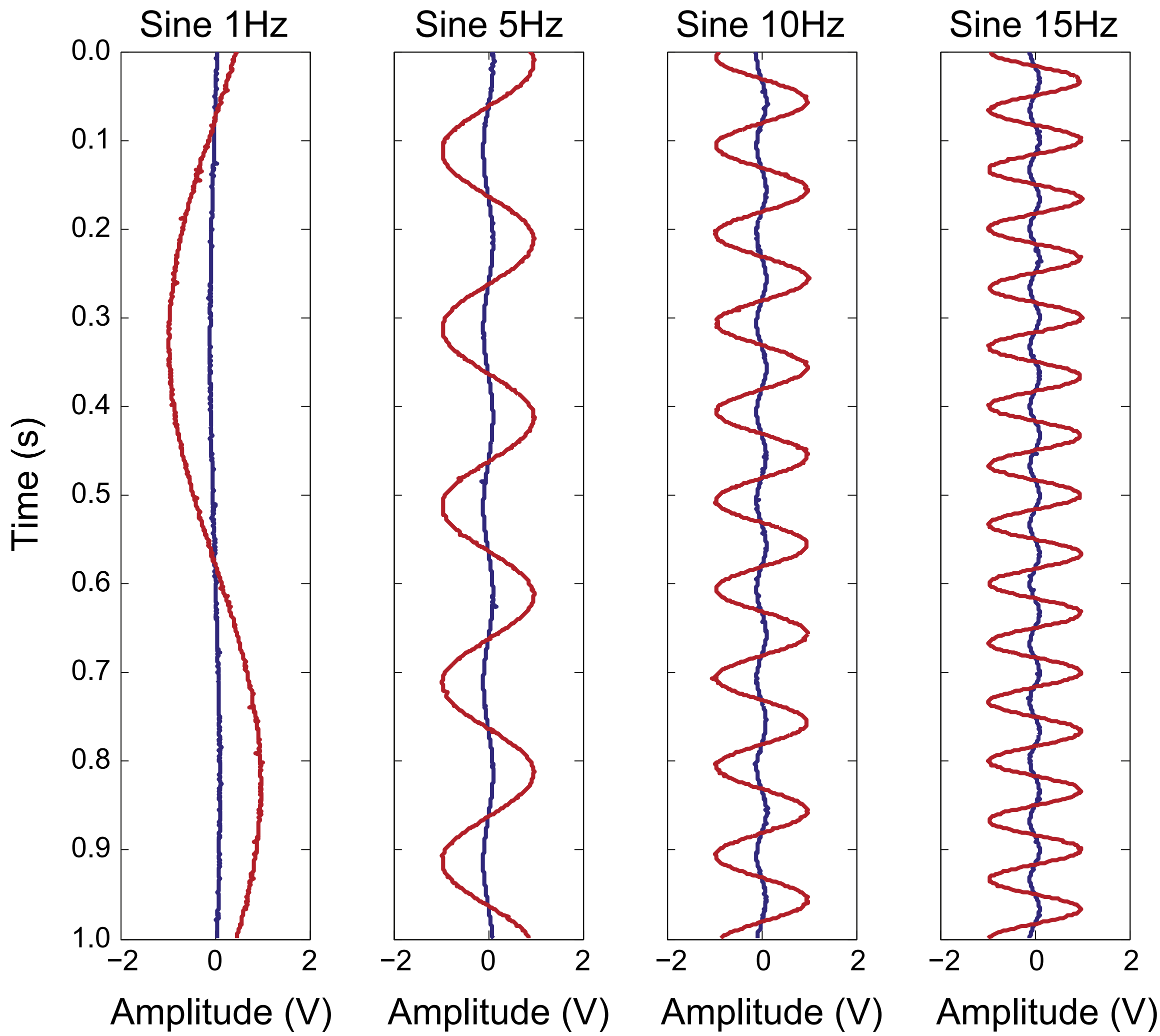

Fig. 6. Registering of sine waves with different frequencies. The input signal (blue) and the amplified signal (red) are shown. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

compare the recorded signals. The sampling rate has been selected the same (1 kHz) for both acquisition systems. In this experiment, the signal provided by the sensor has been first amplified and filtered through the INA155 and MAX7404 chips, respectively. After that, the signal has been connected simultaneously to Arduino Due and the DT321 card. The selected gains for the INA155 and the Arduino Due have been 20 dB (10 ×) and 0 dB (1 ×), respectively.

In Fig. 7, we show the signals recorded by Geophonino (red line) and the DT321 card (blue line) for one calibration pulse. The first pulse corresponds to the mass displacement due to the initial voltage input. The second pulse occurs when the voltage is switched off and the mass returns to the initial equilibrium point. From the obtained results, we can check the good agreement between both recorded signals and the suitability of Geophonino as a low cost data acquisition system.

5.3. Comparison results between Geophonino and the RAS-24 seismograph

In the third test, a real experiment has been carried out in the field, comparing the results obtained by Geophonino with the ones obtained by the commercial RAS-24 seismograph. In this case, each data acquisition system has been connected to one 10-Hz geophone, which have been disposed very close each other. The source was offset at 1 m from both geophones. As active source, a sledgehammer of 3 kg has been released from a height of approximately 2 m, to hit a metal plate (31*23 cm and 2 cm thick) placed at the same distance from the two geophones.

Both data acquisition equipment have been set up with a sampling frequency of 1 kHz. Regarding the voltage gain, it has been configured to 20 and 12 dB for Geophonino and RAS-24, respectively. It is important to note that both equipment incorporate an anti-aliasing filter. The Seistronix RAS-24 seismograph incorporates a bandpass filter that reduces the frequency band to 2–412 Hz for the sampling frequency of 1 kHz. Geophonino has a low-pass filter with a corner frequency of 400 Hz for the selected sampling frequency.

In Fig. 8a, we show one example of the signals recorded simultaneously by Geophonino and the RAS-24. The seismic waves were generated by hitting 3 times with the hammer along a period of around 10 s. We observe that the signal recorded by Geophonino is contaminated with a higher level of noise. It may be due to the quantification noise, since the ADC resolution of the RAS-24 is twice the one corresponding to Geophonino, i.e. 24 against 12 bits, respectively. According to Gray et al. (1998), we can consider a signal-to-noise (SNR) improvement of approximately 6 dB per bit. A possible solution for reducing the quantization noise would be the application of oversampling techniques, as it is explained in Atmel Corporation (2005).

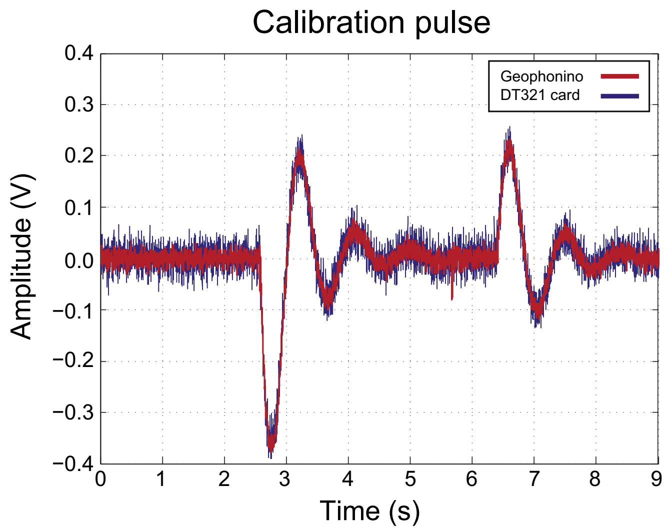

Fig. 7. Calibration pulse recorded with Geophonino (red) and the DT321 card (blue). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

In Fig. 8b, one of the pulses (around the 7 s) has been zoomed to compare them in detail. We can remark the good correlation between both pulses. This good agreement is also observed in the respective spectra of the pulses (Fig. 8c).

6. Conclusions

In this work we have developed a low-cost data acquisition system, Geophonino, which is suitable for seismic signals recordings. The proposed prototype is composed of four main parts: the instrumentation amplifier INA155, the anti-aliasing filter MAX7404, the Arduino Due and the SD card shield. This system is significantly cheaper than any available commercial recorder.

A user interface, based on Proccessing, has been also developed to control the hardware through the USB port. The GUI allows users to configure the main acquisition parameters: duration, sampling rate, gain and file name.

Several experiments have been carried out in order to test the performance of the system and their suitability to record seismic signals. Sine waves of known characteristics have been first used to evaluate the complete system and the acquisition parameters, e.g. gain, sampling frequency, etc. After that, Geophonino has been compared with other commercial data acquisition systems, the DT321 card and the RAS-24 exploration seismograph, registering calibration pulses and pulses generated by active sources, respectively.

From the obtained results, we can observe the suitability of Geophonino to record seismic signals. Thus, it constitutes a low-cost data acquisition system, which overcomes the limitations of the old equipment used by a lot of research groups for scientific or educational purposes.

Besides, the complete prototype is an open source and open hardware system. Thus, it is an easy scalable system, which makes possible the development of future versions. This flexibility makes the system more suitable to any research task, as it can be used as a base system to develop more complex or specific purpose systems.

As future studies, we plan to work in different related issues. The first one concerns to the number of channels. As the Arduino Due board has 12 analog inputs, it would be possible to implement a 12-channel acquisition system with just adding the corresponding conditioning circuits for each channel. Respecting the

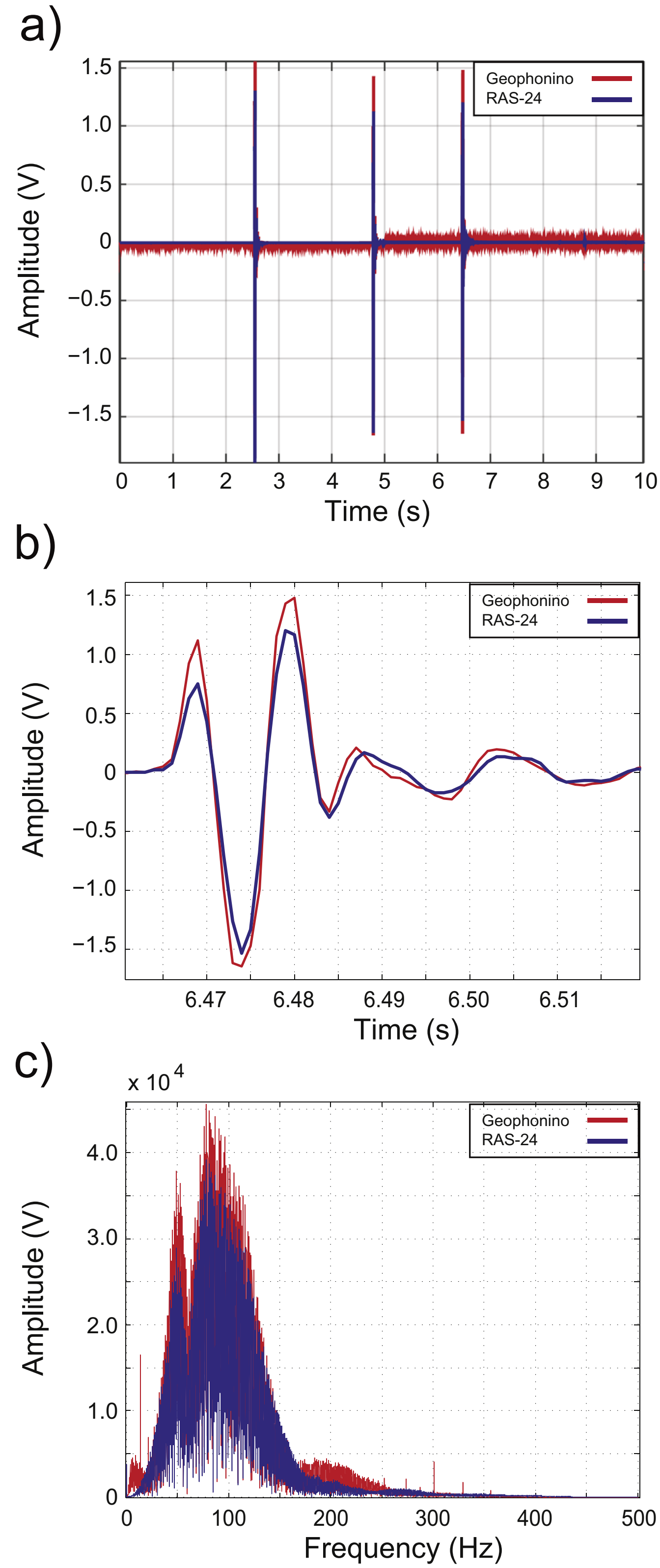

Fig. 8. Real pulses recorded with Geophonino (red) and the RAS-24 exploration seismograph (blue). a) Ten-seconds record with three pulses, b) zoomed third pulse, c) frequency response of the pulse represented in b). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

ADC resolution, it could be improved by means of oversampling and decimation techniques. Finally, other future possible improvements involve also the addition of a voltage-controller amplifier that will be managed through the analog outputs of the Arduino Due; and the addition of a GPS shield to provide the system with positioning data and GPS time.

Acknowledgments

This research has been supported by the Conselleria d'Educació, Investigació, Cultura i Sport de la Generalitat Valenciana (project AICO/2016/098). We thank to the Local Seismic Network of the University of Alicante (supported by Consorcio Provincial de Bomberos de Alicante) that provided us the geophones and the acquisition systems for the test experiments. We are also very thankful to the staff of the Electronic Laboratory of the Research Technical Services of the University of Alicante for their help with the socket adapters. Finally, we thank also to the editor-in-chief, Jef Caers, and the anonymous reviewers for their comments, which helped us to clarify and improve this paper. The identification of the name of the instruments' manufacturers does not mean any endorsement of their products.

Appendix A. Supplementary material

Supplementary data associated with this article can be found in the online version at http://dx.doi.org/10.1016/j.cageo.2016.05.014.

References

Agudo, J.E., Pardo, P.J., Sánchez, H., Pérez, Á.L., Suero, M.I., 2014. A low-cost real color picker based on arduino. Sensors 14, 11943–11956.

Atmel Corporation., 2005. AVR121: Enhancing ADC resolution by oversampling 8003A-AVR-09/05.

Atmel Corporation., 2011. Analog-to-Digital Converter in the SAM3S4 11106A–ATARM–01-Jul-11.

Atmel Corporation., 2012. AT91SAM ARM-based Flash MCU. SAM3X, SAM3A Series. 11057B–ATARM Datasheet.

Bowden, G.B., 2003. Calibration of Geophone Microseismic Sensors. Stanford Linear Accelerator Center, Stanford University. LCLS-TN-O3-6, October 2003, 9 pp.

Burr Brown Corporation, 2000. INA155. Single-Supply, Rail-to-Rail Output, CMOS Instrumentation Amplifier. Technical Datasheet Printed February, 2000. 11 pp.

Data Translation Inc., 2010. DT300 Series Low Cost, Multifunction PCI Data Acquisition Boards. Technical Datasheet. 4 pp.

Fuentes, M., Vivar, M., Burgos, J.M., Aguilera, J., Vacas, J.A., 2014. Design of an accurate, low-cost autonomous data logger for PV system monitoring using Arduino™ that complies with IEC standards. Sol. Energy Mater. Sol. Cells 130, 529–543.

Giner-Caturla, J.J., Jauregui-Eslava, P.J., Delgado-Marchal, J., Rosa-Herranz, J., Soler-Llorens, J.L., 2012. The urban seismic zonation of Lorca, made with values of parameters Vs30 and C obtained through refraction microtremores. In: Proceedings of the 7th Asamblea Hispano-Portuguesa de Geodesia y Geofísica, San Sebastian, España.

Gray, R.M., Neuhoff, D.L., 1998. Quantization. IEEE Trans. Inf. Theory 44, 2325–2383. http://dx.doi.org/10.1109/18.720541.

Khan, K.A., Akhter, G., Ahmad, Z., 2012. DigiSeis—a software component for digitizing seismic signals using the PC sound card. Comput. Geosci. 43, 217–220.

Koenka, I.J., Sáiz, J., Hauser, P.C., 2014. Instrumentino: an open-source modular Python framework for controlling Arduino based experimental instruments. Comput. Phys. Commun. 185 (10), 2724–2729.

Picozzi, M., Milkereit, C., Parolai, S., Jaeckel, K.H., Veit, I., Fischer, J., Zschau, J., 2010. GFZ wireless seismic array (GFZ-WISE), a wireless mesh network of seismic sensors: new perspectives for seismic noise array investigations and site monitoring. Sensors 10, 3280–3304.

Rosa-Cintas, S., Galiana-Merino, J.J., Rosa-Herranz, J., Molina, S., Giner-Caturla, J., 2013. Suitability of 10 Hz vertical geophones for seismic noise array measurements based on frequency-wavenumber and extended spatial autocorrelation analyses. Geophys. Prospect. 61, 183–198.

Ruytenberg, T., Webb, A.G., Beenakker, J.W.M., 2014. A multi-purpose open-source triggering platform for magnetic resonance. J. Magn. Reson. 247, 15–21.

Savazzi, E., 2011. Construction and programming of an autonomous focus stacker. Comput. Geosci. 37, 1670–1676.

Schlegel, A., 2015. ControlP5, a GUI library for the programming environment Processing. Zenodo . http://dx.doi.org/10.5281/zenodo.16290.

Seidel, I., Klaus, P., Feilb, Colombod, Suppa, M., Alexander, M., Van Gael, J., Murphy, J., 2015. DueTimer. Senodo . http://dx.doi.org/10.5281/zenodo.16039.

Seistronix Inc., 2005. Data Sheet Ras-24 Digital Seismograph Rev. 1.04. 3299D Monier Cr., Rancho Cordova, CA 95742, USA.