Using a single-board microcontroller and ADC to perform real-time sonar signal processing

Ben Travaglione

Defence Science and Technology Group, Australia

Email: ben.travaglione@defence.gov.au

ABSTRACT

Over recent years, the processing power of single-board microcontrollers has increased significantly. These microcontrollers are low cost, have a small footprint, use relatively small amounts of power and are able to interface with a multitude of different devices. One such microcontroller is the Beaglebone Black. In this work we test the current limits of a Beaglebone Black acting as a portable, real-time, multi-channel, sonar signal processor. By interfacing analogue-to-digital converter chips to the open-source microcontroller, running open-source software, we show that it is possible to process high resolution, high sample rate, hydro-acoustic data in real-time.

1. INTRODUCTION

There are many situations in environmental monitoring, defence, and the study of marine life were it is necessary to obtain information about sound levels in the underwater environment over extended periods of time. Sound sources of interest in the undersea environment can have large variations in sound intensity level and can also be of a higher frequency than in air. These features of the sound sources of interest place heavy demands on an underwater acquisition system. Large variations in sound intensity level are often dealt with by using an acquisition system with high precision. To access the high frequencies of interest, an acquisition system requires a correspondingly high sample rate. There are a variety of commercial devices available for taking underwater measurements, however these devices are generally costly and have been geared towards a specific task. There is a demand for more compact, low cost, light weight and versatile alternatives (Martinez et al., 2011). The current crop of open-source single-board microcontrollers offer a viable path to such alternatives. These microcontrollers have a number of attributes which make them attractive for use in an acquisition system, including low cost, portability, and a high degree of configurability.

In previous work we have looked at a number of low-cost single-board microcontrollers for building underwater acquisition systems, including the Raspberry Pi and the Arduino (Travaglione, Munyard, and Matthews, 2014). However, the Beaglebone Black has a number of advantages over the other single-board microcontrollers. We have shown that a Beaglebone Black can be combined with a commercially available analogue-to-digital converter (ADC) chip to create an acquisition system capable of simultaneously sampling up to eight channels at 130 kSPS (kilo samples per second) with a precision of 24 bits (Travaglione, Munyard, and Matthews, 2015, Travaglione, 2016) at a cost of less than AU$ 500, as compared to a commerical system with similar characterisics, which would cost around AU$ 10 000. Others have investigated the Beaglebone Black's potential for recording multi-channel audio (Moro et al., 2016,Langer and Manzke, 2015), and there is a keen interest in using single-board microcontrollers as acquisition systems (Moure et al., 2015).

If an undersea environment is to be monitored over an extended period of time, a direct consequence of the high sample rate and high precision of an acquisition system is the requirement for a large storage capacity. A common task in acoustics is to perform frequency analysis of recorded sounds. Often, in post-processing, the frequency data will be reduced into bands whose width is proportional to their frequency, drastically reducing the size of the stored data. Third octaves are a commonly used compromise between frequency fidelity and stored data size. It would be highly advantageous if an underwater acquisition system was able to process the sound in real time and reduce it to third octave levels.

In this paper, we show that a single Beaglebone Black combined with analogue-to-digital converter chips can simultaneously acquire data from a number of hydrophones (dependent upon the sample rate) at 24-bit precision and process the input in real time to produce, store and transmit third octave levels. (Although the device will not give 24-bit accuracy, by using a 24-bit ADC we can ensure that the accuracy of the acquisition system will be limited by electronic noise levels and the limitations of the attached sensors rather than the quantization process of the ADC.) In Section 2 we give a summary of the features of the Beaglebone Black, highlighting those which aid in the construction of an acquisition system. In Section 3 we discuss the characteristics of the particular analogue-to-digital converter chips used in our systems. Section 4 looks at the Beaglebone Black Acquisition System (BBBAS), outlining the processing bottleneck associated with rapidly storing the large data sets created by the high precision and the high sample rate.

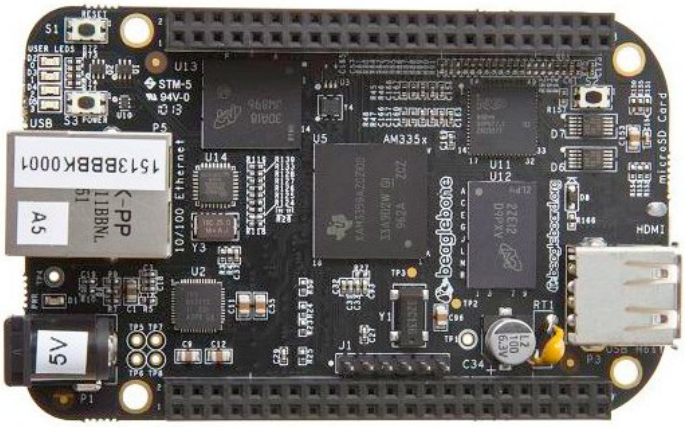


Figure 1: The Beaglebone Black. This microcontroller is shown to scale when document is printed at A4 size. [Image taken from the Beaglebone Black website (BeagleBoard.org Foundation, n.d.)]

Section 5 describes how a Beaglebone Black can be used to overcome some of the problems of large data sets by doing real-time processing to produce and transmit third octave levels. Finally, in Section 6 we draw some conclusions and discuss some potential applications of a portable, undersea, real-time sonar processor.

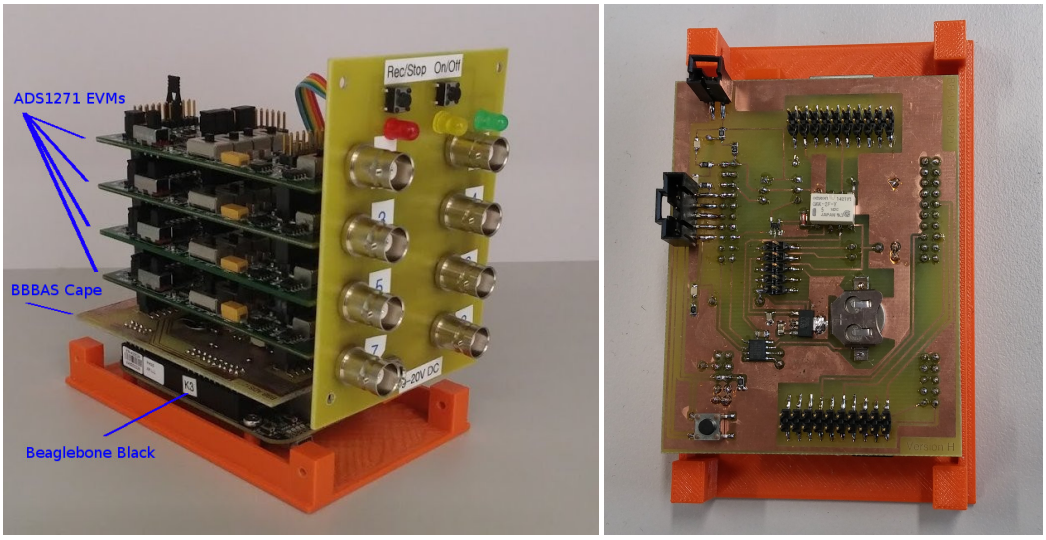
2. BEAGLEBONE BLACK

The Beaglebone Black is a single-board microcontroller, which at its heart contains a Sitara AM335x ARM Cortex-A8 chip (BeagleBoard.org Foundation, n.d.,Texas Instruments, n.d.(c)). There are a number of features which make the Beaglebone Black an attractive option in the development of an underwater sonar processing unit including its small footprint (see Figure 1), low power consumption and low cost. The Beaglebone Black has 4 GB of on-board memory and 512 MB of RAM, and the Sitara chip runs at 1 GHz. It has 92 General Programmable Input/Output (GPIO) pins. However, for the purposes of our acquisition system, the unique feature of the Beaglebone Black is the two Programmable Real-time Units (PRUs) contained within the Sitara chip (*TI AM33XX PRUSSv2* n.d.). These programmable units are 32-bit processors, running at a clock speed of 200 MHz. They each have 8 KB of program memory, and access to a shared memory space which can be configured to hold up to 8 MB. Each of the PRUs also has direct access to a subset of the 92 GPIO pins. These PRUs enable the Beaglebone Black to drive peripheral devices and access data from these devices in real time, as will be shown in Section 4.

The Beaglebone Black can run a variety of different operating systems, however the most common, and the one which we use, is the Debian variety of Linux. Having access to a extensive and well developed operating system on the microcontroller confers a number of significant advantages. It is possible to quickly and easily install a wide range of applications to enhance the functionality of an embedded system such as the BBBAS discussed in Section 4. For example, the device can be given a web interface by installing a web-server application such as Apache with a script interpreter such as PHP. The PRUs need to be programmed in ARM assembly language, and the Linux-side application is programmed in C. Having the relevant compilers on-board the microcontroller obviates the need for complicated and time-consuming cross-compilation. The extensive open-source on-line community also makes trouble-shooting and debugging much easier than for a closed-source project.

3. ANALOGUE-TO-DIGITAL CONVERSION

A key component of an acquisition system is the chip which takes in a voltage signal, and translates the analogue voltage level into a digitised value. For our acquisition system, we chose to use the ADS1271, which is a 24-bit, delta-sigma analogue-to-digital converter (ADC) with a recommended maximum data rate of 105 kSPS (Texas Instruments, n.d.(a)). (Although the data-sheet and manual give a maximum speed of 105 kSPS, in our testing, the device performed well at a speed of 130 kSPS.) The ADS1271 has three modes of operation, allowing for the optimisation of speed, resolution, or power. It also has two selectable methods of communication; either an SPI protocol or a frame-sync protocol. To maximise the number of channels a single Beaglebone Black is able to process, we used the SPI protocol for communication between the ADS1271 chips and the Beaglebone Black. Another important feature of the ADS1271 is the ability to 'daisy-chain' the chips. The output from one ADS1271 chip can be fed into the next chip, which sends its output, plus the output of the previous chip onto the next chip and so on, through a series of chips, until the final chip sends all the data on to the Beaglebone Black. This allows the development of an acquisition system capable of



(a) (b)

Figure 2: (a) The BBBAS without a cover, showing the ADS1271 EVM boards and the Beaglebone Black, connected via the BBBAS Cape. (b) Top view of the BBBAS Cape.

simultaneous high precision multi-channel recording, and while the Beaglebone Black needs to send a clock signal to every ADS1271 chip, it only needs to receive data from one chip (the last in the line).

The manufacturers of the ADS1271 also produce an evaluation module (ADS1271 EVM) for the ADS1271 (Texas Instruments, n.d.(b)). The ADS1271 EVM contains two ADS1271 chips, has a footprint similar to the the Beaglebone Black and a number of 2.54 mm pin headers. Thus, the ADS1271 EVM greatly simplifies the circuitry needed to both daisy-chain a number of ADS1271 chips together, and to interface with the Beaglebone Black. The evaluation modules also contain unity-gain amplifiers for each ADC, which can act as over-voltage protection circuits for the relatively sensitive ADS1271 chips.

4. BEAGLEBONE BLACK ACQUISITION SYSTEM

The Beaglebone Black Acquisition System (BBBAS) is an open-source acquisition system (based on open-source hardware, running open-source software) capable of simultaneously recording eight input channels at 24-bit precision, with a sample rate as high as 130 kSPS. The acquisition system is described in greater detail in (Travaglione, 2016). Here we simply outline the main specifications of the BBBAS device, and give enough detail of the internal workings of the device to enable discussion of the modifications which allow the device to also perform as a real-time processor.

It is possible to power the BBBAS with a DC source anywhere in the range of 9 to 20 Volts, however it was specifically designed for, and tested using, a 12 V lead-acid battery. All subsequent values for electric current are at the nominal input of 12 V. When idle (not recording or processing) the BBBAS draws approximately 100 mA, with an additional current of approximately 50 mA being consumed if a Wi-Fi dongle is attached. When recording or processing inputs, the system powers the ADS1271 EVMs, each of which consume approximately 80 mA, hence, when recording on all eight channels, the BBBAS system draws approximately 470 mA.

The Beaglebone Black is connected to the ADS1271 EVM boards via the BBBAS Cape (In the beaglebone community, a cape is a circuit board designed to connect to a Beaglebone Black). Figure 2 (a) shows the BBBAS, composed of a Beaglebone Black, the BBBAS Cape, four ADS1271 EVM boards and a front panel with BNC connectors for the eight input channels. Figure 2 (b) shows a top-view of the BBBAS Cape. The primary role of the BBBAS Cape is to connect the PRUs on the Beaglebone Black's Sitara chip to the ADS1271 input pins via various GPIO pins on the Beaglebone Black. As well as interfacing the microcontroller to the ADC chips, the BBBAS Cape fulfils a number of other roles, including providing power to the microcontroller and ADC boards, providing a relay to control the power to the ADC boards, and providing matched positive and negative voltage sources to the unity-gain op-amps on the ADS1271 EVM boards. The BBBAS Cape also provides the Beaglebone Black with a real-time clock, which allows recordings to be assigned accurate timestamps.

The BBBAS has a browser-based web interface, with network connections over Ethernet, USB, or Wi-Fi. The BBBAS web interface allows the user to choose the number of channels to be recorded as well as the length and sample rate of the recording. The interface also allows the user to download the recordings, or analyse the recording

Figure 3: Screen-shot of the BBBAS web interface, showing three test signals being analysed by the device's open-source software.

using the web interface. An example of a recording analysis is depicted in Figure 3. The code running on the PRUs is written in ARM assembly language. The code which stores the data on a microSD card attached to the Beaglebone Black is written in C. The web interface is written in php and javascript. The analysis software is written in python using the Sci-Py package (Jones, Oliphant, Peterson, et al., 2001–), and graphics is provided by the Bokeh package (Bokeh Development Team, 2014). The analysis software is highly configurable, and in fact can be altered for each data set using a Jupyter notebook (Pérez and Granger, 2007). The data can be transferred over the network, either as raw binary or as a compressed, lossless flac file. The data can also be accessed at a later date by powering down the device and extracting the attached microSD card.

4.1. Real-time data acquisition

In order to explain the real-time processing capabilities of the BBBAS we must first discuss the roles of the various processors contained within the AM335x chip. As mentioned in Section 2, there are three separate processors within the Beaglebone Black which are used during the acquisition of data. First, we explain the roles of the two PRUs, which both have a clock period of 5 ns. In high speed mode, the ADS1271 requires 256 SPI clock cycles to complete a digitisation cycle. Using one of the PRUs as an SPI clock, the very minimum period of the SPI clock will be 10 ns (one cycle to set an output pin to high, and one cycle to set it to low). However, the other PRU also has a 5 ns clock cycle, and this PRU has more work to do. Within a single SPI clock cycle, it must be able to decrement a bit counter, read a bit of data from the ADS1271 chip, write the bit of data to an internal register, check whether it has finished measuring a channel, and then prepare for the next bit. This set of operations takes 5 PRU clock cycles, and therefore, if we wish for a symmetric clock (a requirement of the ADS1271 chip), the minimum period for the SPI clock is 6 PRU clock cycles. This gives a sample period for the system of $256 \times 6 \times 5$ ns $= 7680$ ns, corresponding to a sample rate of approximately 130208 samples per second. This sample rate is above the specified maximum for the ADS1271, but in all our testing, the chip appears stable and able to maintain its digitisation at this speed. Lower sample rates are made possible by extending the amount of time that the clock GPIO pin is kept low and high. Any integer multiple (greater than 2) of $256 \times 2 \times 5$ ns $= 2.56$ μs will be a valid sample rate period. In the BBBAS device, we chose multiples of 3, 4, 5, 6, and 8. Giving sample rates of 130 208 Hz, 97 656 Hz, 78 125 Hz, 65 104 Hz, and 48 828 Hz (rounding to the nearest hertz).

Each ADC produces 24 bits of data per digitisation, and each bit takes one SPI clock cycle to transfer from the ADS1271 to the PRU, so reading all the data from up to eight channels takes 192 SPI clock cycles. Remembering that the ADS1271 starts a new digitisation cycle every 256 clock cycles, the recording PRU has 64 SPI clock cycles (which at the maximum sample rate is 1920 ns) to copy the data from the PRU memory to the shared RAM. At this point, the acquired

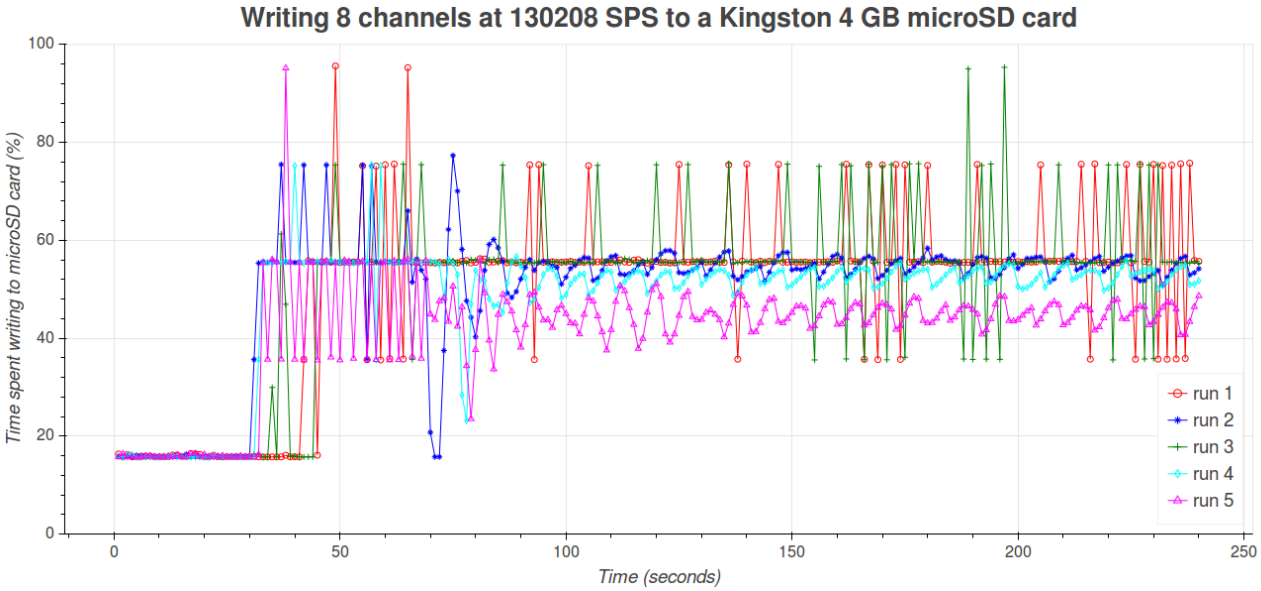


Figure 4: The write-time to the microSD card is unpredictable. Each second, the system needs to write 3 MB of data to the microSD card. This figure depicts the percentage of each second spent writing data to the microSD card for 4 different 4 minute recordings. The variation in the write-time for different runs is indicative of the different amounts of wear for different sectors of memory on the microSD card. The initial fast write-times in the first 30 - 40 seconds is due to file buffering to RAM by the Linux system.

data is in RAM which is accessible to the main processor on the Sitara chip, however the size of this RAM is limited to 8 MB. It is therefore necessary to copy the data from the RAM to a more permanent storage space. This is accomplished with the use of two buffers in the shared RAM. The acquisition algorithm has been written such that the PRU stores a chunk of data into one of the buffers before signalling to the main processor, and then begins storing the next chunk of data into the second buffer. If the system is in acquisition mode, the main processor (which is not a real-time system) has a limited amount of time to write the data from each buffer to file storage before the PRU will begin to over-write the memory with new data. This process of writing the data to file storage is relatively slow. Remembering that the Beaglebone Black has only 4 GB of onboard storage, which contains the operating system for the microcontroller, the digitised data is instead stored on an attached microSD card. In order to optimise the write speed, the code has been designed to write the output directly to the microSD device, bypassing some of the time-consuming Linux file system processes. Even with these optimisations, writing to file constitutes a significant bottleneck in the acquisition process. At its maximum capacity, the system needs to write approximately 3 MB of data to the microSD card every second, which corresponds to a data transfer rate of 24 Mbps. Writing to the microSD card is also highly dependent upon the quality of the card. There are a number of different classes of microSD card, with associated minimum specifications for write speeds, however even for cards with the same class, there are large differences between different brands. The specification which is most important for an acquisition system is the sequential write speed. Unfortunately this speed is not constant, even for a single microSD card, and varies over time, as the internal components in the microSD card carry out wear levelling, in order to get the maximum lifetime out of the card. The Linux operating system is able to overcome the variation in write times to the card by buffering the data to RAM, however this only works until the RAM buffer is filled.

Figure 4 shows the percentage of time spent writing to the microSD card for several different runs, recording eight channels at a sample rate of 130 208 SPS when writing to a Kingston 4 GB Class 10 microSD card. The variation in the write-time for different runs is indicative of the different amounts of wear for different sectors of memory on the microSD card. The initial fast write-times in the first 30 - 40 seconds is due to file buffering to RAM by the Linux system.

An important point to remember is that for the acquisition system to store a recording without any data loss, it is the worst-case write time that is critical. If the time spent writing exceeds the time available before the PRU over-writes with new data, then there will be data losses in the stored recordings. As is evident in Figure 4, recording eight channels at 130 kSPS takes the BBBAS to the limit of its write speed for this particular microSD card. However, it is often the case that the raw data is not needed, rather it is the processed data which is required by the end user. If we can process the data in real-time, rather than store all the acquired data, we can substantially reduce the amount of data that needs to be written to file. The following section looks at a specific example of real-time processing.

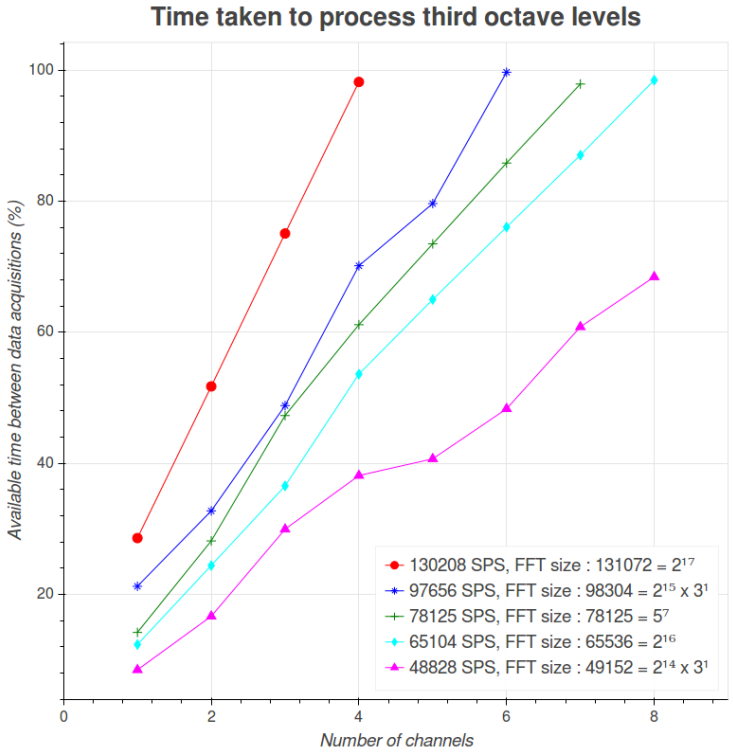


Figure 5: The time taken to produce third octave levels as a percentage of the available time between buffer over-writes and as a function of the number of channels.

5. REAL-TIME PROCESSING OF THIRD OCTAVE LEVELS

As explained in the previous section, one of the PRUs on the Beaglebone Black takes in the digitised data from each ADS1271 chip and stores these values to one of two shared RAM buffers, which are then available to the 1 GHz main processor on the Beaglebone Black. The question then becomes; how much processing is the Beaglebone Black able to do with the data contained in a buffer before the acquired data is overwritten with new data? To test this we added the ability to generate third octave levels for the acquired data. Third octave levels were generated by first transforming the acquired data from the time domain into the frequency domain. We wished for a resolution of at least 1 Hz in the frequency domain. Setting the size of each buffer, and hence the input vector length of the Discrete Fourier Transform (DFT), equal to the sample rate provides a frequency resolution of exactly 1 Hz. However, this will generally result in poor performance, because the DFT is implemented via a Fast Fourier Transform (FFT) algorithm, and the speed of such algorithms are dependent upon the size of the factors of the input vector length. We chose to use the FFTW package (Frigo and Johnson, 2005) to implement our DFT. To increase the speed of processing, FFT sizes were chosen such that they were a product of small prime factors, whilst keeping frequency resolution as close as possible to 1 Hz.

The BBBAS calculates third octaves by first separating the acquired data in each buffer into separate channels, transforming each 24-bit integer into a 64-bit float representing the voltage measure by the ADC, applying a spatial windowing function, Fourier transforming the resultant data set for each channel, summing the resultant powers over each third octave bin, before finally transforming the data into decibels.

We took a number of other steps to optimise the processing code. This included writing the code in a single C program, which meant that we could set the priority of this single program to the maximum level permissible within the Linux operating system. We also shut down all unnecessary programs on the Linux system whilst it is in sonar processing mode.

Figure 5 shows the time taken to process the inputs as a percentage of the available time between buffer over-writes and as a function of the number of channels. The time taken to produce the third octaves is shown for a number of different sample rates. As the sample rate increases, the size of the input to the FFT increases, increasing the time taken to calculate the third octaves. It is only possible to generate real-time third octave levels for all eight channels if the sample rate is less than 65 kSPS, however, even at the maximum sample rate of 130 kSPS the BBBAS is able to consistently calculate, transmit and store third octave levels for four channels in real time.

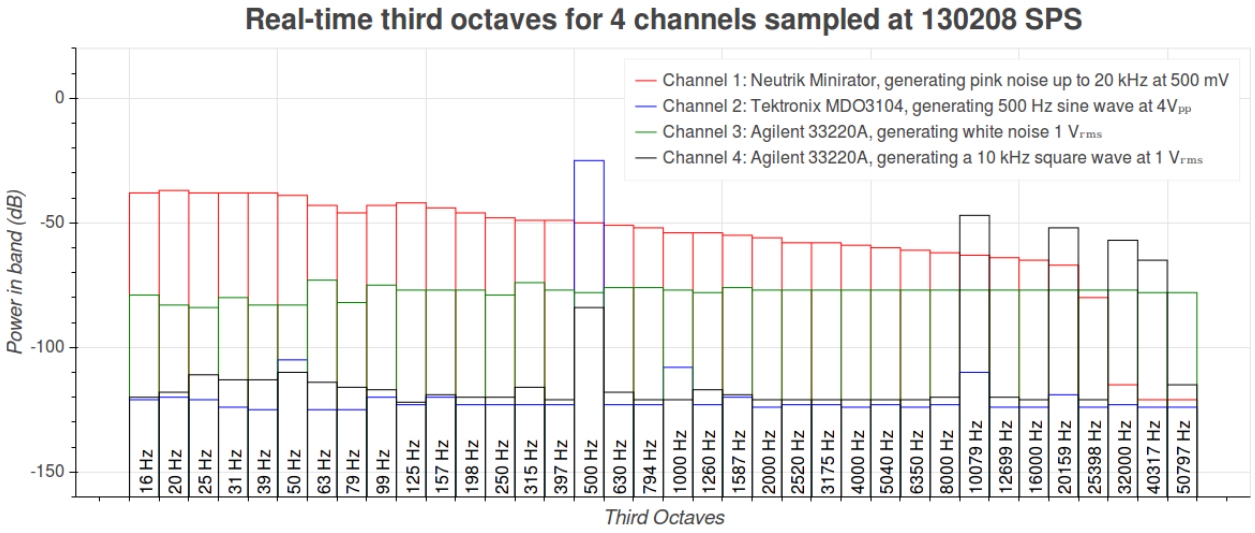


Figure 6: An example of the third octave levels calculated in real-time for four simultaneous signals at 130 kSPS.

Unlike the storing of the raw acquired data, which had inconsistent timing due to memory algorithms on the microSD card, the real-time processing happens entirely within the RAM of the Beaglebone Black, and therefore is consistent over time. For the maximum sample rate, and maximum number of channels, the amount of data being stored each second is reduced from approximately 3 MB down to less than 1 kB. Because this data rate is now so small, it is possible to transmit the third octave levels in real time over a communications channel (Wi-Fi, Ethernet, USB) using a simple User Datagram Protocol (UDP).

Figure 6 shows an example of the third octave levels calculated in real-time for four simultaneous signals at 130 kSPS. Channel one was attached to a Neutrik Minirator, generating pink noise up to 20 kHz at 0.5 V_{rms}. Channel two was attached to Tektronix MDO3104, generating a 500 Hz sine wave at 4 V_{pp}. Channel three and four where each attached to separate Agilent 33220A signal generators. The signal generator attached to channel three was generating white noise at 1 V_{rms}, while the signal generator attached to channel four was generating a 10 kHz square wave at 1 V_{rms}.

6. CONCLUSIONS AND FURTHER RESEARCH

In this paper we have shown that a Beaglebone Black attached to a number of ADS1271 chips can process high-speed, high-precision data in real-time to produce third octave levels for a number of input channels. We have previously shown that a Beaglebone Black can form the basis of an eight channel, 130 kSPS, 24-bit acquisition system, however the high precision and sample rate place high demands on the data storage capability of the Beaglebone Black, especially when writing to a microSD card. By processing the data in real-time, we can obviate the need to store such large raw data sets. We have shown that all eight 24-bit channels can be transformed into the frequency domain and summed into third octave bins for sample rates up to 65 kSPS. If a sample rate as high as 130 kSPS is required, then it is possible to generate real-time third octave levels for four simultaneous 24-bit channels.

The lifetime of an undersea acquisition system is commonly limited by either the energy requirements of the device or the amount of storage space available for recorded data sets. The BBBAS attempts to address both of these concerns. The energy requirements of the device are relatively modest, thanks to the relative efficiency of the Beaglebone Black microcontroller. The problems with limited storage can be addressed by allowing the device to generate third octave levels for the input channels, and then use these third octaves to trigger recording only when signals of interest are detected.

Having shown that real-time processing is possible with an open-source system based on a Beaglebone Black, a natural next step would be to investigate the possibility of finding correlations between the different channels, and thus perform some form of array processing.

7. ACKNOWLEDGEMENTS

The author would like to acknowledge the assistance and advice of David Matthews, Joshua King and Andrew Munyard.

REFERENCES

BeagleBoard.org Foundation. *"Beagleboard Black Homepage"*. Viewed 15 July 2016, http://beagleboard.org/black.

Bokeh Development Team (2014). *"Bokeh: Python library for interactive visualization"*. URL: http://www.bokeh.pydata.org.

Frigo, M. and S. Johnson (2005). "The Design and Implementation of FFTW3". In: *Proceedings of the IEEE* 93.2, pp. 216–231.

Jones, E., T. Oliphant, P. Peterson, et al. (2001–). *"SciPy: Open source scientific tools for Python"*. Viewed 20 July 2016, http://www.scipy.org/.

Langer, H. and R. Manzke (2015). *"Linux-Based Low-Latency Multichannel Audio System"*. Viewed 15 July 2016, http://www.creative-technologies.de/linux-based-low-latency-multichannel-audio-system-2/.

Martinez, J. et al. (2011). *Design and Implementation of an Underwater Sound Recording Device*. Tech. rep. Pacific Northwest National Laboratory.

Moro, G. et al. (2016). "Making high-performance embedded instruments with Bela and Pure Data". In: *International Conference on Live Interfaces*. (June 29–July 3, 2016). URL: http://www.eecs.qmul.ac.uk/~andrewm/moro_icli2016.pdf.

Moure, D. et al. (2015). "Use of Low-Cost Acquisition Systems with an Embedded Linux Device for Volcanic Monitoring". In: *Sensors* 15.8, p. 20436. URL: http://www.mdpi.com/1424-8220/15/8/20436.

Pérez, F. and B. E. Granger (2007). "IPython: a System for Interactive Scientific Computing". In: *Computing in Science and Engineering* 9.3, pp. 21–29. URL: http://ipython.org.

Texas Instruments. *"ADS1271 24-Bit, 105kSPS Industrial Delta-Sigma ADC"*. Viewed 20 July 2016, http://www.ti.com/product/ads1271.

— *"ADS1271 Evaluation Module (EVM)"*. Viewed 20 July 2016, http://www.ti.com/tool/ads1271evm.

— *"AM3358 Sitara Processor"*. Viewed 15 July 2016, http://www.ti.com/product/am3358.

TI AM33XX PRUSSv2. Viewed 15 July 2016, http://elinux.org/Ti_AM33XX_PRUSSv2.

Travaglione, B. (2016). "In preparation".

Travaglione, B., A. Munyard, and D. Matthews (2014). "Using low cost single-board microcontrollers to record underwater acoustical data". In: *43rd International Congress on Noise Control Engineering*. URL: http://www.acoustics.asn.au/conference_proceedings/INTERNOISE2014/papers/p236.pdf.

— (2015). "High resolution undersea acoustic data acquisition using single-board microcontrollers". In: *Australian Acoustical Society Acoustics 2015 Hunter Valley*. URL: http://www.acoustics.asn.au/conference_proceedings/AAS2015/papers/p63.pdf.