

GPS-Disciplined Analog-to-Digital Converter for Phasor Measurement Applications

Xiaodong Zhao, *Member, IEEE*, David M. Lavery, *Member, IEEE*, Adrian McKernan, D. John Morrow, *Member, IEEE*, Kieran McLaughlin, and Sakir Sezer

Abstract—This paper presents a data acquisition unit which synchronously samples multiple channels in a manner such that the time of day at which each sample is taken is known. This allows measurements taken at multiple locations to be compared with confidence. The intended application is wide area electrical power system measurements, in particular phasor measurement units (PMUs). The novelty of the authors' design is the application of an open hardware development platform to discipline a commodity analog-to-digital converter (ADC) to a broadcast time signal, usually but not exclusively GPS. The methodology used creates a driver layer for the ADC to achieve real-time sampling in a nonpreemptive Linux environment. The use of open hardware and software addresses the need for a transparent instrument for use in research and development of PMU technology. Through a choice of either a software or hardware phase-locked loop, the ADC is controlled to acquire exactly 256 samples per nominal power system cycle (i.e., 50/60 Hz), precisely time synchronized to GPS, at 16-b resolution and 94.2-dB SNR. The design of a printed circuit board expansion board featuring all necessary components is provided. The performance of the system is evaluated. Interoperability and data exchange with other systems is achieved by use of open schemas and communication protocols. This allows rapid integration with popular numerical simulation environments.

Index Terms—Data acquisition (DAQ), phase estimation, power system measurements, synchronization, timing.

I. INTRODUCTION

THE phasor measurement unit (PMU) is quickly becoming a standard piece of equipment found in transmission substations for the purposes of fault analysis, and has been popular in the literature for numerous years as the cornerstone of many real-time protection and control concepts [1]–[4]. In simple terms, a PMU samples voltage and current waveforms in synchronism with a broadcast time signal, allowing the computed waveform parameters at one location to be compared with those from other locations across a wide geographical area. The waveform parameters, including amplitude, phase,

and frequency, are assembled into a phasor for storage or telecommunication. By definition, a phasor is only valid in a stationary system; that is to say that the frequency is constant. Naturally, much of the interest in phasor analysis is around fault conditions, or transient conditions, during which the system frequency will vary. Such interest is addressed in the recent versions of IEEE Std. C37.118.1, which describes the requirements for PMUs [5].

Best *et al.* [3] and Lavery *et al.* [4] have interests in the use of PMU for real-time protection and real-time control applications. Evaluation of commercially available PMUs has shown that many are validated against the earlier IEEE C37.118-2005, prior to the dynamic conditions requirements, and thus the performance of these units is unsatisfactory for analysis of transient events. Consequently, the authors have been working toward an open source PMU, allowing end user control and visibility over the entire function of the device. The authors' motivations are described in [6].

In previous work, the OpenPMU [6] used a commercially available data acquisition (DAQ) unit. Although the overall performance of the OpenPMU was satisfactory, its future development in the form described could not be further improved upon. One of the principal limitations was that the sampling clock of the DAQ employed utilized a local crystal oscillator and provided no means to discipline this to an external source. Instead, a trigger interrupt was used to window the sample data according to a time signal. The impact of sampling time error is well described in [7], which notes that a “saw tooth” error is observed on the outputs of susceptible PMUs. While DAQs which do allow for externally-disciplined sampling clocks are commercially available, these tend to be expensive, complex and proprietary. The latter, namely, that the means by which the sampling is performed is not known to the end user, is an important motivation for this paper. An approach in [8] uses similar equipment to the authors' technique, however, the methodology is not open sourced and thus is not transparent to the end user.

This paper describes a GPS-disciplined analog-to-digital converter (ADC) that has been developed by the authors to provide a low cost, yet accurate and precise, alternative to commercial offerings. The main benefit of this approach is that by carefully disciplining the sampling clock to an external time signal; the timing related errors described in [7] are eliminated. Furthermore, this approach will reduce development time for instruments requiring precision timing and wide area coordination. An added benefit is the use of an open source

Manuscript received November 25, 2016; revised January 31, 2017; accepted March 22, 2017. Date of publication May 18, 2017; date of current version August 9, 2017. This work was supported in part by the EPSRC CAPRICA (EP/M002837/1), in part by EPSRC SuppleHeat (EP/M000656/1), and in part by EPSRC i-Give (EP/L001063/1) Projects. The Associate Editor coordinating the review process was Dr. Wendy Van Moer. (*Corresponding author: David M. Lavery.*)

The authors are with the School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, Belfast BT7 1NN, U.K. (e-mail: xzhao06@qub.ac.uk; david.lavery@qub.ac.uk; a.mckernan@qub.ac.uk; dj.morrow@qub.ac.uk; kieran.mclaughlin@qub.ac.uk; s.sezer@qub.ac.uk).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIM.2017.2700158

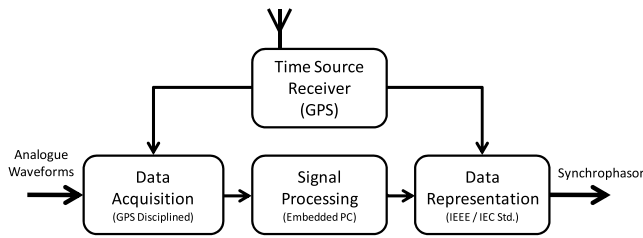


Fig. 1. Modules of the OpenPMU system.

development platform, the BeagleBone Black. The architecture of the device is described and the performance of the time keeping is evaluated.

II. GPS DISCIPLINED PHASOR ACQUISITION

The open source phasor measurement system planned by the authors, OpenPMU, modularizes the functions of a PMU into four distinct components. These are shown in Fig. 1, and are based upon functional descriptions of PMU described in [9]. The main functions are DAQ, signal processing and data representation. These are governed by a time signal, disciplining the system to Coordinated Universal Time (UTC). This paper is primarily concerned with the DAQ module. In the OpenPMU system, information is exchanged between modules by user datagram protocol (UDP) datagrams, with the data arranged in an XML schema.

The correct function of the PMU is predicated on the availability of good quality time synchronized waveform sample data, from which phasor parameters can be estimated. PMUs are normally synchronized to a broadcast time signal that is itself disciplined to the UTC time base. Due to the widespread availability of the NAVSTAR GPS time signals distributed by satellites, many commercial PMUs synchronize to this time signal by means of an integrated GPS receiver. Often an input compatible with IRIG-B is fit for use with substation master clocks and other time services.

Assuming that a good quality time signal is available at the PMU device, one of two main approaches is taken when acquiring the waveform sample data. The first approach is that an ADC is employed to convert the waveform into numerical values sampled at intervals controlled by an independent oscillator local to the ADC. This method does not intrinsically lock the sampling clock to the reference time signal, so a windowing function is employed on the microprocessor controlling the ADC to adjust the sampling data so that it appears in synchronism with the reference time signal. This might involve interpolation of sample points to fit specified time values, or the time data of the raw samples may be recorded and utilized by the phasor estimation algorithm to correct for sampling clock drift.

The second approach is to construct the DAQ system such that the sampling clock is synchronized to the time signal, thus ensuring that the ADC samples the waveform at known instances in time. When this method functions correctly there is no need for further processing of the waveform sample data to correct for sampling clock drift. This approach is favored by the authors.

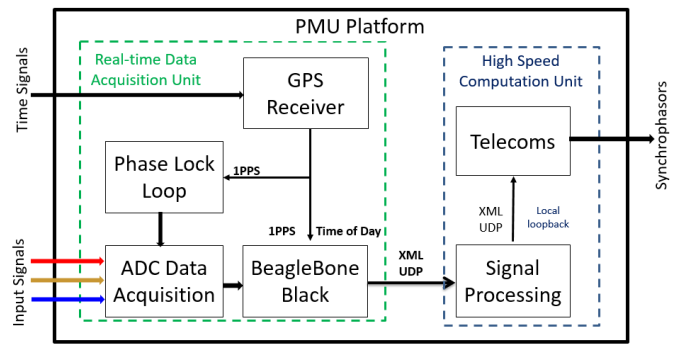


Fig. 2. Schematic of PMU platform.

The authors have surveyed the function of a number of commercial PMUs in their possession, and it is apparent that many utilized the former method described above by observation of the timestamps on the point-on-wave sample data. For these particular devices, the PMU functionality was added to an existing mature product family of disturbance fault recorders. It is possible that the vendor may lock the sampling process to the observed line frequency, but this functionality varies from device to device.

Commercial laboratory equipment may be employed to construct a GPS-disciplined ADC, however, in the authors' experience the market leading solutions proved to be highly closed source and proprietary, considerably expensive, and require extensive training in a specialist programming environment.

The authors identified the need for a simplified GPS-disciplined ADC which allows for total visibility of signal paths and information flow while maintaining a modular platform, as shown in Fig. 1. The DAQ module developed by the authors makes use of widely available, cost effective, and highly documented components. These are constructed around an open source hardware development platform, the BeagleBone Black [10]. A schematic is shown in Fig. 2. The subject of this paper is those components belonging to the DAQ unit. This unit is considered to be "real time," in that the hardware and software must work to hard real-time goals. The computation unit performs the signal processing and telecoms functions. These are more computationally taxing functions, and although must be performed at high speed, do not have hard real-time deadlines. Traditional x86 PCs are well suited to this task, and modern low-power microprocessors such as found on the "Raspberry Pi" family of computers may be used if the algorithms are suitably optimized. The authors found that the Beaglebone Black lacked the resources to operate the functions of the computation unit in real time. The following sections will describe the DAQ unit in depth.

III. HARDWARE ARCHITECTURE

The overall architecture of the GPS-disciplined DAQ unit is shown in Fig. 3. It consists of four main components: the "BeagleBone Black" which is the main controller of the unit, the ADC, the phase-locked loop (PLL), and the GPS receiver. These components are described in turn.

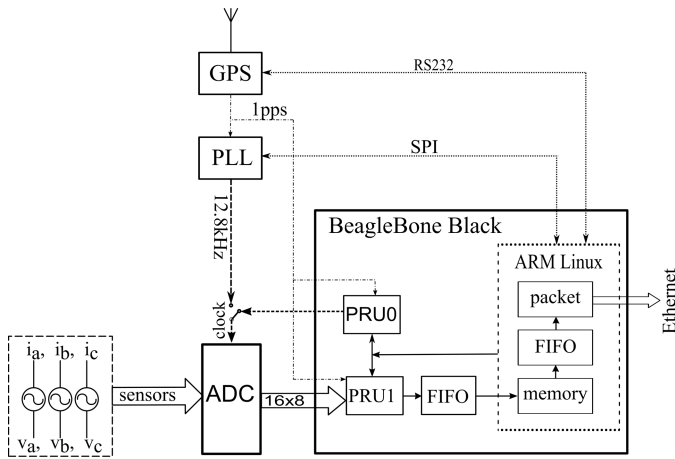


Fig. 3. Architecture of the GPS-disciplined ADC.

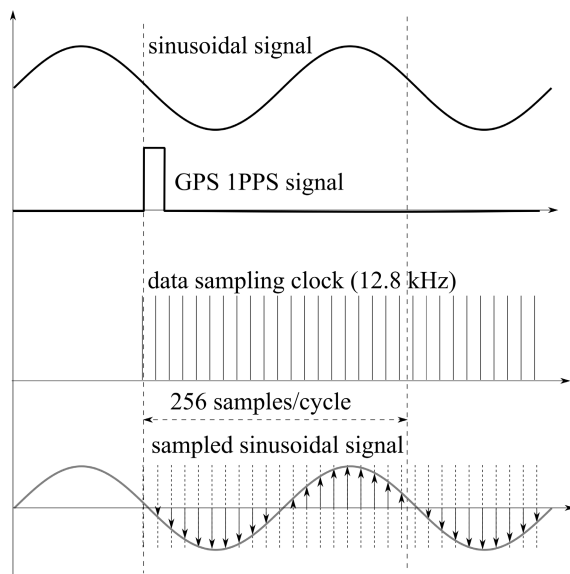


Fig. 4. GPS-disciplined signal sampling.

The general concept is to use the 1-pulse-per-second (1PPS) time signal from the GPS receiver to discipline a sampling clock which causes the ADC to acquire waveform sample data at the correct instant in time. Fig. 4 illustrates the sampling of a sinusoidal signal to a sampling clock disciplined to 1PPS. That is, the 1PPS provides the phase reference for the PLL which is producing the sampling clock. Due to the universal availability of the GPS 1PPS time signal, the exact same sampling time instants can be guaranteed at different geographical locations. The accuracy of the sampling time will be limited by the intrinsic accuracy of GPS, the quality of the receiver and the quality of the PLL. A sampling rate of 12.8 kHz is used for 50-Hz systems, yielding nominally 256 samples per cycle at 50 Hz. On a 60-Hz system, 15.36-kHz clock would be used.

The challenge is how to ensure that sample data is transferred to the higher level operating system with the time at which the data was acquired intact. This is achieved using shared memory, as described in the following section. The ADC communicates by either a serial or parallel interface (SPI). For speed, the parallel byte interface on the

ADC is used in preference to the serial interface. This transfers 8 b at once, thus reads a 16-b sample in two CPU cycles.

The sampling clock is generated by a PLL. The authors have provisioned the DAQ unit with two PLL options. One of these is a commercial IC, which shall be called the “external PLL,” the other is a software PLL programmed on the BeagleBone Black. The user may select which PLL is in use from the operating system. The performance of both PLLs is evaluated later in this paper.

The GPS module provides timestamps for the sampling. A host program running on ARM Linux manages and gathers all the data and information, then packages the measured sample data along with the time at which it was taken to a phase estimation algorithm on another CPU via Ethernet.

A. BeagleBone Black

The DAQ unit is constructed around the “BeagleBone Black.” This is a low cost development board featuring the AM3358 ARM Cortex-A8 processor from Texas Instruments, clocked at 1 GHz. It can run a variety of operating systems, including Android, Debian, Ubuntu, and other Linux variants, affording the benefits of a high level operating system with low level access to input–output registers for connectivity to external components.

The particular advantage of the BeagleBone Black for this application is the existence of two “programmable real-time units” (PRUs). The PRUs are two separate reduced instruction set computing CPUs on the same silicon die as the main ARM CPU, with separate data and instruction memories while sharing the same data bus. The PRUs are clocked at 200 MHz and have access to pins, events and hardware resources on the system-on-chip, so they can be tasked with hard real-time functions. That is to say, they can be given a function to do that operates independently of the operating system on the main CPU, thus the typical causes of delays that would interfere with a real-time process are eliminated, while data can be shared based on the same memory map between ARM Linux and PRUs.

The functional diagram of the BeagleBone Black in this application is illustrated in Fig. 5, where PRU1 is configured to read the ADC data in real time, and PRU0 is programmed to generate a software-based sampling clock, which will be detailed in the PLL section. The host program in ARM Linux has access to the PRU data via the internal shared memory bus, which will allow the ADC data to be read asynchronously whilst preserving accurate timing data.

Each PRU has its dedicated data and code RAM with a size of 8 kB. A shared data RAM of 12 kB can also be utilized. For a sampling rate of 12.8 kHz, 16-b data width, eight channels, the memory needed for a sampling period of 10 ms would be

$$12.8 \text{ k} \times 2 \times 8 \times 0.01 = 2.048 \text{ kB.}$$

Thus a queue with a maximum size 5×10 ms periods can be used to buffer the shared data between ARM Linux host program and PRU code. That is, up to 50 ms of acquired waveform data can be buffered before the host program must retrieve it. In practice, the buffer will not be allowed to reach this size as it is read nominally every 10 ms.

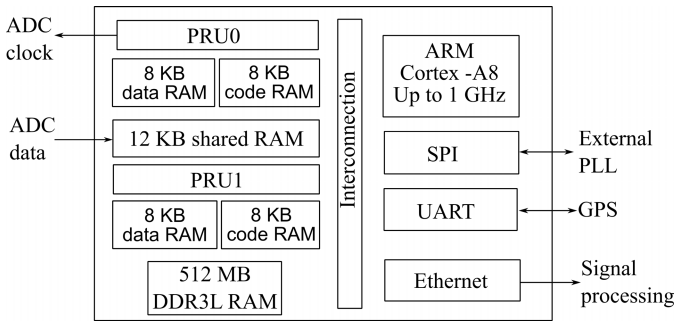


Fig. 5. BeagleBone Black function diagram.

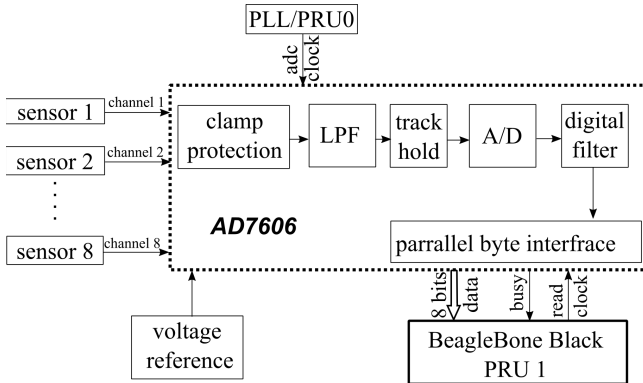


Fig. 6. ADC function diagram.

B. Analog-to-Digital Converter

The ADC employed is the AD7606 from analog devices. This is an 8-channel, 16 b, 200 kSPS (kilo-samples per second) ADC IC with SPIs. The key feature of interest to this application is that all eight channels are simultaneously sampled by means of sample and hold circuitry which may be externally triggered via a TTL input. This ADC is well suited to the acquisition of power system waveforms, as described in [11]. This application may employ either the external PLL or the software timer (PRU0) of the BeagleBone Black, as shown in Fig. 6. It is thus possible to “clock” the sampling process accurately and then read the data asynchronously from the main CPU which will acquire, package and transmit all the data.

With 12.8 kHz as the selected sampling frequency for the PMU application, which is relatively slow compared to the maximum sampling rate of the ADC, a digital filter with oversampling ratio of 8 on the ADC is utilized to further improve the SNR from 89 to 94.2 dB [12], in addition to the low-pass filter on chip.

The eight input channels with voltage ranges of ± 10 V are sampled simultaneously to guarantee that all the signals have the same time stamp. The ADC may be coupled to standard instrument voltages and currents by means of suitable transducers. The authors have used Hall effect transducers when testing the ADC.

C. GPS Module

The GPS unit employed by the authors is a Garmin GPS-18x LVC, a relatively common GPS unit featuring a 1PPS output providing time signal, and additional data including

time-of-day and navigational information via NMEA protocol over RS232. The accuracy of the 1PPS quoted by Garmin on the GPS-18x product datasheet is $\pm 1 \mu\text{s}$ at the rising edge of the pulse, but in practice the unit is shown to perform much better. Pedersen demonstrates the GPS-18x to have a ± 94 ns “wander” around a consistent 126-ns offset compared to a precision GPS receiver [13]. This is with a weak indoor signal. Given that the manufacturer’s datasheet claims a 30-m navigational accuracy, an internal temporal accuracy of ± 100 ns is implied, consistent with Pedersen’s observations. At this order of magnitude, the properties of the cable used for the time signal, not least its length, becomes important. A 200-ns time error in a PMU observing a 50-Hz system would lead to only a 0.0036° error in phase angle estimation. Other sources of estimation error will be of greater significance. The authors note that the Garmin module is comparable to the quality of modules found inside many commercial PMU appliances, but a more temporally precise GPS module (or other time signal) could be substituted if the end user required this.

Time-of-day information is communicated from the GPS receiver to the BeagleBone Black operating system using universal asynchronous receiver/transmitter (UART)/RS232, employing the NMEA protocol at a baud rate of 19 200 (1920 characters per second). Following the rising edge of the 1PPS signal, the GPS module transmits the GPRMC sentence of the NMEA protocol, which contains within it the date and time of the preceding 1PPS. The GPRMC sentence can be 74 characters long at most, leading to a delay of around $74/1920 = 38.54$ ms. Accurate time-of-day can be obtained by the operating system at any instant in time by extrapolating from the rising edge of the 1PPS.

D. Phase-Locked Loop

1) *External PLL*: PLLs are a mature technology with myriad devices available in the marketplace. However, most such devices are targeted toward professional and commercial radio applications, thus are designed to work with both input reference and output frequencies in the MHz / GHz ranges. The output frequency is limited to some multiple of the reference frequency, typically 1 to 2 orders of magnitude. Some PLLs are available for the audio range in the 10/100 kHz ranges, but less common are PLLs which will work down as low as 1 Hz which is required when using a 1PPS signal as a phase reference. Since a sampling clock of 12.8 kHz is desired, the 1-Hz pulse needs to be multiplied by some four orders of magnitude. One such device that fulfills this need is the AD9548 from analog devices.

AD9548 contains a digital PLL (DPLL) core with four pairs of input reference pins with frequencies from 1 to 750 MHz, and four pairs of output pins up to 450 MHz, making it ideal for the application of GPS 1PPS synchronization. The core DPLL has a similar structure to an analog PLL, except that a digital loop filter is used after the phase-frequency detector (PFD), as well as the direct-digital synthesizer (DDS) instead of an analog voltage-controlled oscillator.

Shown in Fig. 7, the reference input is the GPS 1PPS, which is reduced by a prescaler R , where $R = 1$ in this application.

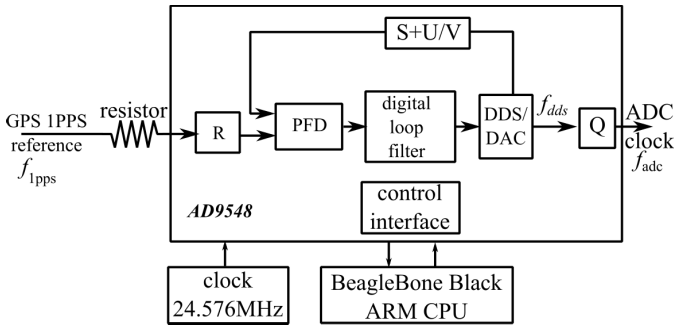


Fig. 7. Simplified diagram of DPLL.

The PFD delivers the difference between reference and feedback as digital words to the loop filter. The feedback divider provides the inter-plus-fractional multiples of reference input, then the output frequency of the digital-to-analog converter (DAC) is reduced again using a second scale Q , after which the ADC sampling clock can be obtained as

$$f_{\text{adc}} = \frac{f_{1\text{PPS}}}{R} \left(S + \frac{U}{V} \right) \frac{1}{Q}$$

where in this application, for the 12.8kHz sampling clock, those parameters are chosen to be

$$R = 1, \quad U = V = 0, \quad S = 1.28 \times 10^8, \quad Q = 10000.$$

In the laboratory setup, ringing was observed on the 1PPS signal, which could cause it undesirably retrigger the DPLL, thus a series resistor for impedance matching is used before the reference input [14].

The DPLL requires a clock source to drive its internal functions and DDS/DAC, which will depend on a precise clock source such as temperature-controlled oscillator (TCXO) or oven-controlled oscillator [15]. For 1PPS operation in this application, the maximum loop bandwidth of the digital loop filter is 0.05 Hz. Due to the fact that the 1PPS signal has a time error around ~ 100 ns [13], the “phase-locked threshold” detection for the DPLL is set to its maximum value of 65 ns, which is the time difference between the 1PPS and divided feedback at the PFD. To reduce the total cost of the application and avoid the usage of an extremely stable clock source, a TCXO with frequency tolerance of 0.1 ppm/°C (parts per million per degree Celsius) is adopted. Based on the module of the DPLL with the third-order digital loop filter in [16], the maximum allowed change of frequency for the TCXO can be obtained as

$$\Delta f = 0.014 \text{ Hz/sec}$$

which means for a TCXO of 24.576 MHz, the maximum allowed temperature change in one minute would be

$$\Delta T = \frac{\Delta f \times 60}{0.1 \text{ ppm} \times 24.576 \text{ MHz}} = 0.34 \text{ }^\circ\text{C/min.}$$

As long as the temperature change sits inside ΔT , the PLL can lock on the 1PPS signal. For a PMU enclosed in a plastic enclosure at a fixed location, it is unlikely to see such temperature changing rates, thus it is exceptional for the PLL

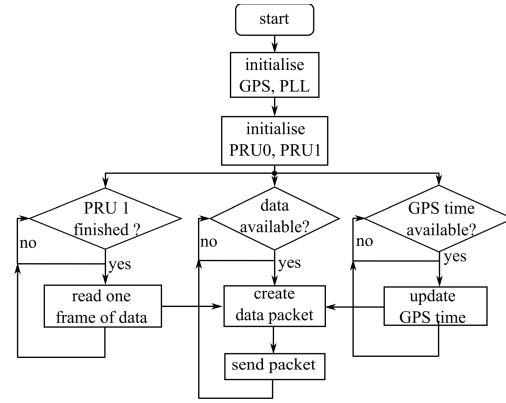


Fig. 8. ARM Linux host program flowchart. Sampled data from PRU1 is exchanged with the Linux host via shared memory. The memory addresses are organized to indicate the sequence of the samples from the first sample of the UTC second (indicated by 1PPS), such that the time of day may be transferred from the GPS module.

to lose lock due to temperature variations. Otherwise, the PLL would run in hold-over mode based on the reference history, or the software approach described below can be used.

2) *Software PLL*: An alternative approach that is appropriate given the low frequencies involved is to use a software defined PLL, in this case implemented on a PRU on the BeagleBone Black (PRU0). The advantage of this method is that it eliminates the requirement for a discrete PLL. The disadvantage is that the accuracy of this method is reduced, with the granularity of the counter determined by the clock speed of the PRU. In this case a clock speed of nominally 200 MHz, with one instruction executed per clock cycle, gives a counter granularity of nominally 5 ns. PRU0 features a 32-b CPU counter which is read on each rising edge of the 1PPS to determine the frequency of the crystal. The sampling period is decreased or increased depending whether the PRU0 counter indicates if the crystal is running fast or slow relative to the 1PPS. Fine granularity may be achieved by adjusting the sampling period on a per cycle basis, although a deadband is required to accommodate jitter in the 1PPS from the GPS (~ 100 ns). The 1PPS resets the PLL to ensure that the sampling frequency is in phase with the 1PPS.

IV. SOFTWARE DESIGN

Because of the unique structure of BeagleBone Black, the software involved can be divided into three parts: the host program running on ARM Linux and the two PRU codes. The PRU codes are loaded and started by the host program, while exchanging data through shared memory map. Due to the continuous sampling and sending process, it is critical to synchronize each operation in the right order.

A. Host Program Design

The ARM Linux host program is the center of operations, including the selection and control of the PLL, and gathering information from GPS and ADC via PRU1. Timing is a critical issue in the program, so it is split into three parallel tasks to avoid waiting on I/O operations as shown in Fig. 8. The



Fig. 9. XML data format used for communication of sample data between DAQ unit and signal processing unit. Sample data is expressed in base64 so that it may be represented as ASCII characters.

first task is responsible for gathering the sample data acquired by PRU1 at regular intervals of 10 ms. This forms one data frame, which includes sample data from all eight analog input channels. The second task reads the UART every second to acquire GPS time. The third task converts all the binary sample data into base64, so it can be represented as ASCII characters. The time information and other appropriate metadata relating to the samples are gathered and packaged into a datagram structured with XML markup, and transmitted to the signal processing unit using UDP/IP. An example datagram is shown in Fig. 9.

The phase estimation block will decode the XML datagram and recover the raw ADC data allowing it to compute phasor parameters using an appropriate phase estimation algorithm, as shown in Fig. 2. It is separated from the DAQ unit since its functional requirements are different. It requires a powerful CPU to calculate complex mathematical functions at a high speed, but provided the sample data was labeled with the correct time stamp, there is no need for this to be a real-time process. The authors usually employ a typical desktop PC, although have had success in using a Raspberry Pi 2. The use of XML/UDP makes the DAQ unit agnostic as to the architecture of the signal processing unit.

B. PRU Codes

The two PRUs are dedicated processors running in parallel with the ARM Linux host. Flowcharts for both PRU cores are shown in Fig. 10.

PRU0 is designed to generate the sampling clock for the ADC in case the external PLL is unavailable. To achieve the best timing performance, it is coded in Assembly language. The program starts by checking the rising edge of the 1PPS signal, on which it will loop to generate a 12.8-kHz sampling clock. The accuracy of this method depends on the execution time of each instruction which is nominally 5 ns. Since the

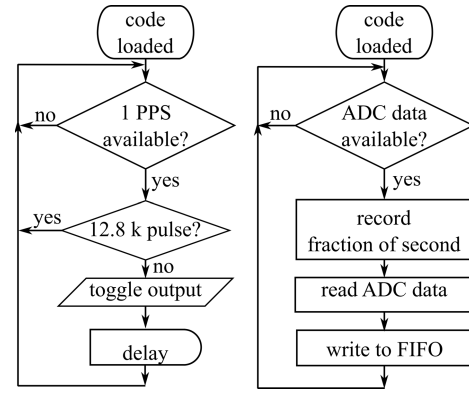


Fig. 10. Code flowcharts. PRU0 flowchart (left). PRU1 flowchart (right).

200-MHz PRU clock is not disciplined to a time signal, it will drift with respect to the 1PPS but this is compensated for by comparison with the PRU0 CPU counter.

The function of PRU1 is to acquire sample data and make this available in blocks to the host program. PRU1 is evoked to acquire sample data by the ADC when the “data available” TTL signal rises. PRU1 acquires one sample in turn from each of the eight channels and appends the new data to the FIFO buffer shared with the ARM Linux host. It continues to acquire sample data, with each period of 128 samples (10 ms) forming one frame. The fraction of second from the rising edge of the 1PPS signal is also recorded in this buffer. In this way, PRU1 provides a driver layer for the ADC to achieve real-time sampling in a nonpreemptive Linux environment.

V. PERFORMANCE EVALUATION

A. PLL Error Analysis

Besides the inherent ~ 100 ns error from the GPS module, there will be errors caused by the PLL as well. With regards to the external PLL, the PLL provides indications of frequency lock and phase lock, which can be tested and visualized by an oscilloscope. Even when the PLL is locked to the reference, there will still be error caused by the distribution block Q . This time error between rising edges of the reference signal (1PPS) and the locked output is determined by two parts: the first is the deterministic delay by the output divider Q as shown in Fig. 7, plus an integer m , and the second part is the propagation delay t_{prop} inside the PLL IC

$$t_{pll} = (Q + m) \times \frac{1}{f_{dds}} + t_{prop}$$

where m is 4 or 5 and f_{dds} is the output frequency of DDS as shown in Fig. 7. For 12.8-kHz output, f_{dds} is 128 MHz.

To compensate for the time error t_{pll} , the phase angle of the PLL output is shifted after the frequency and phase lock have finished. This can minimize the time error at the event of lock, however, it cannot track the error changes over temperature or power supply.

The software triggered sampling clock is determined by the PRU oscillator speed. As the PRU is running at nominally 200 MHz, it will take ~ 5 ns for each instruction and at least

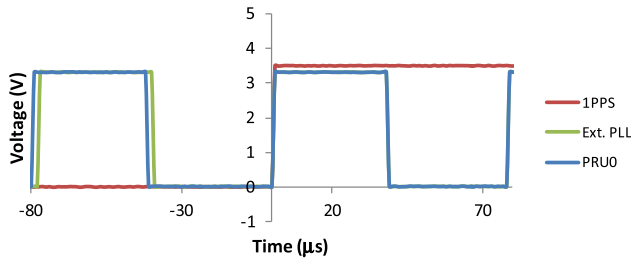


Fig. 11. Snapshot of PLL outputs pre- and post-1PPS.

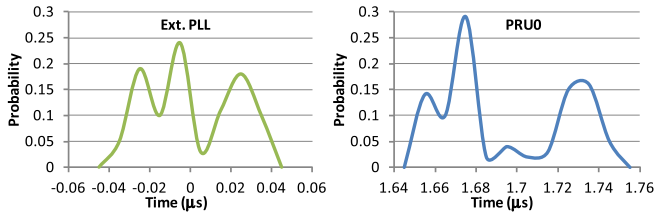


Fig. 12. Probability distribution of the error for each PLL method over 100 measurements, with respect to an ideal rise time of 78.125 s.

two instructions for a rising edge detection. So the rising edge difference will be nominally 10 ns.

Observed on an oscilloscope, it is apparent that both PLL methods are in phase with each other immediately following the 1PPS rising edge from the GPS ($t > 0$ s). This would be anticipated, given that the two PLLs have just been resynchronized. However, approaching the 1PPS ($t < 0$ s), there is an observable phase error between the two PLL methods (see Fig. 11).

To characterize the error observed in Fig. 11, a storage oscilloscope was configured to capture the output of each PLL in the period leading up to the 1PPS. The sampling rate is 250 MHz, yielding 4-ns granularity.

The correct time for the rising edge is 78.125 μ s prior to the 1PPS occurring. Over the course of 2 h, 100 measurements were stored by the oscilloscope. The probability distribution of the errors observed for each PLL method is shown in Fig. 12. The mean rise time of the hardware PLL was calculated to be 78.126 μ s prior to the 1PPS, with standard deviation of 23 ns, and maximum error of 30 ns. This is nearly ideal. Note that the jitter in the hardware PLL output may be a result of jitter in the 1PPS itself.

The software PLL was found to have a mean rise time of 79.82 μ s, standard deviation 33 ns, and maximum error of 1.75 μ s. This would equate to a phase angle estimation error of 0.031°, which is less than a tenth the permissible error in [5]. By inspection of the oscilloscope data, the drift is spaced equally over the 1PPS interval and thus would manifest as a “saw tooth” phase error as described in [7]. The authors consider that further refinement of the software PLL algorithm may improve this result. In practical applications where the measured waveforms are acquired from a live system, the error introduced by this sampling time error is sufficiently small as to be indistinguishable from noise.

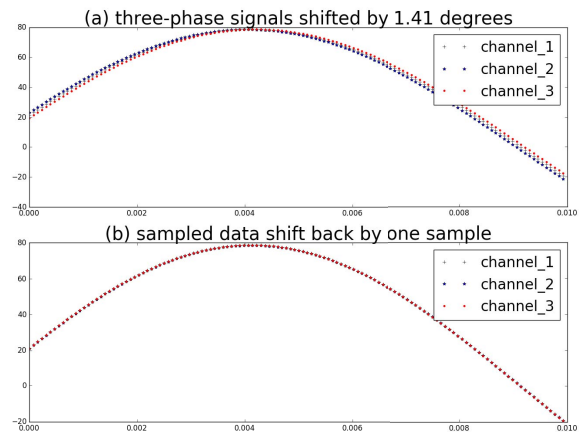


Fig. 13. Three-phase test wave with one sample shifted by 1.41°. (a) Three-phase signals shifted by 1.41°. (b) Sampled data shift back by one sample.

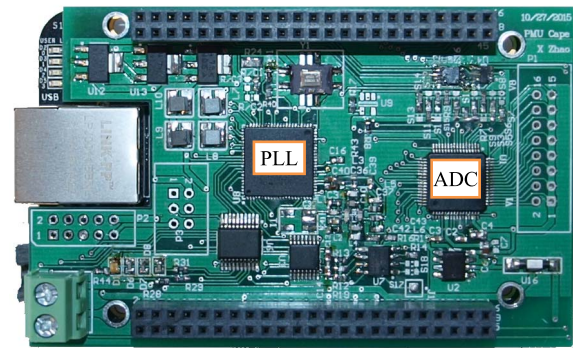


Fig. 14. DAQ board in the form of a BeagleBone Black “cape.”

B. Sampling Performance Evaluation

The authors have subjected the ADC to the compliance tests described in [5] using a recently calibrated Omicron CMC256 + GPS disciplined relay test set. Since the results of these compliance tests are heavily dependent on the phase estimation algorithm employed, they cannot be used to directly assess the performance of the ADC. However, there are two notable observations. The phase estimation algorithm requires packets of 128 samples per channel from the ADC. A sample over or under will cause the phase estimation to fail. This did not occur during testing. Likewise, if the ADC sampling clock was running fast or slow, the estimated phase angle would drift or exhibit a “saw tooth” pattern [7] with respect to the theoretical phase angle during compliance tests. This was not observed, indicating that the sampling clock is stable.

A 50-Hz three-phase test waveform was generated using the relay test set, with each phase shifted by 1.41° in advance of its predecessor, equivalent to one sample difference at a sampling rate of 12.8 kHz. This is shown in Fig. 13(a). If the ADC sampling rate is exactly 12.8 kHz, the three-phase signals will then overlap when shifted by one sampling period. This has been successfully verified in the testing illustrated in Fig. 13(b).

C. Physical Construction

The final design of the DAQ unit, excluding the voltage/current transducers, is constructed into a “cape,” which

can be fitted on the top of the BeagleBone Black, as shown in Fig. 14. This provides a small and neat module for the overall PMU platform, and can also be used as a data sampling component for other power system applications.

VI. CONCLUSION

It is anticipated that PMU adoption will increase across the electrical utility sector in both traditional monitoring, and novel protection and control applications. Myriad commercial devices are in existence but many lack certifications against dynamic testing requirements. The “closed” nature of the hardware and software of these devices makes them of limited use for research and development work.

This paper has presented a method to discipline the sampling clock of an ADC to a broadcast time signal for the purpose of waveform DAQ for phasor estimation. Of particular importance to this application is that the sampled data is taken at a consistent sampling rate, in synchronism with the time signal. This allows synchrophasors to be computed and compared over a wide area.

The DAQ unit is capable of synchronously acquiring eight analog waveforms at a precise rate of 12.8 kHz (for 50-Hz systems) or 15.36 kHz (for 60 Hz systems), with 16-b resolution and 94.2-dB SNR. The sampling clock is achieved by PLL, using either a commercially available IC, or a software PLL of the authors’ design. Both have been evaluated and are shown to be suited to the application of phasor measurement. The software PLL described by the authors is implemented on the popular open source Beaglebone Black development board. The external components for the DAQ unit have been constructed into a “cape” for the BeagleBone Black, yielding a highly affordable instrument.

Data is expressed in base64 and communicated along with the time at which it was acquired and other metadata in an XML formatted datagram using UDP/IP. This allows the data to be exchanged with a wide variety of platforms, software languages, and environments. Due to the flexibility of the data representation, in addition to phasor estimation, this system could have hardware-in-the-loop application in providing sampled data to power system simulation environments. This system is of use to any application which requires real-time sampling on a nonpreemptive operating system. It is proposed that this design can be used as the “engine” for smart grid appliances, reducing development times, and costs for the DAQ stage.

REFERENCES

- [1] E. O. Schweitzer, D. Whitehead, G. Zweigle, K. G. Ravikumar, and G. Rzepka, “Synchrophasor-based power system protection and control applications,” in *Proc. Int. Symp. Modern Electr. Power Syst. (MEPS)*, Sep. 2010, pp. 1–10.
- [2] M. Prasad, K. N. Satish, K. S. Meena, and R. Sodhi, “A synchrophasor measurements based adaptive underfrequency load shedding scheme,” in *Proc. IEEE Innov. Smart Grid Technol. Asia (ISGT Asia)*, May 2014, pp. 424–428.
- [3] R. J. Best, D. J. Morrow, D. M. Lavery, and P. A. Crossley, “Synchrophasor broadcast over Internet protocol for distributed generator synchronization,” *IEEE Trans. Power Del.*, vol. 25, no. 4, pp. 2835–2841, Oct. 2010.

- [4] D. M. Lavery, R. J. Best, and D. J. Morrow, “Loss-of-mains protection system by application of phasor measurement unit technology with experimentally assessed threshold settings,” *IET Generat. Transmiss. Distrib.*, vol. 9, no. 2, pp. 146–153, 2015.
- [5] *IEEE Standard for Synchrophasor Measurements for Power Systems—Amendment 1: Modification of Selected Performance Requirements*, IEEE Standard C37.118.1a-2014, Amendment to IEEE Standard C37.118.1-2011, Apr. 2014, pp. 1–25.
- [6] D. M. Lavery, R. J. Best, P. Brogan, I. Al Khatib, L. Vanfretti, and D. J. Morrow, “The OpenPMU platform for open-source phasor measurements,” *IEEE Trans. Instrum. Meas.*, vol. 62, no. 4, pp. 701–709, Apr. 2013.
- [7] W. Yao *et al.*, “A novel method for phasor measurement unit sampling time error compensation,” *IEEE Trans. Smart Grid*, to be published, doi 10.1109/TSG.2016.2574946.
- [8] J. A. Culliss, “A 3rd generation frequency disturbance recorder: A secure, low cost synchrophasor measurement device,” Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Univ. Tennessee, Knoxville, TN, USA, 2015.
- [9] A. G. Phadke and J. S. Thorp, *Synchronized Phasor Measurements and Their Applications*. New York, NY, USA: Springer, 2008.
- [10] *BeagleBoard.Org—Black—What is BeagleBone Black?*, accessed on Jan. 12, 2016. [Online]. Available: <http://beagleboard.org/BLACK>
- [11] Z. He and Y. Liao, “The design of analog acquisition system in distribution automation,” in *Proc. China Int. Conf. Electr. Distrib. (CICED)*, Sep. 2012, pp. 1–4.
- [12] (2010). *AD7606 Data Sheet Analog Devices, Inc.* accessed on Jan. 12, 2016. [Online]. Available: http://www.analog.com/media/en/technical-documentation/data-sheets/AD7606_7606-6_7606-4.pdf
- [13] K. Pedersen. *GPS 18x 2.70 Test (IPv6 Access): Test of the Garmin GPS-18x With Firmware Version 3.70. How Precise, How Accurate?*, accessed on Jan. 12, 2016. [Online]. Available: <http://n1.taur.dk/gps18x/>
- [14] (2009). *AD9548 Data Sheet Analog Devices, Inc.* accessed on Jan. 12, 2016. [Online]. Available: <http://www.analog.com/media/en/technical-documentation/data-sheets/AD9548.pdf>
- [15] (2009). *AN-1002 Application Note Analog Devices, Inc.* accessed on Jan. 12, 2016. [Online]. Available: <http://www.analog.com/media/en/technical-documentation/application-notes/AN-1002.pdf>
- [16] (2010). *AN-1079 Application Note Analog Devices, Inc.* accessed on Jan. 12, 2016. [Online]. Available: <http://www.analog.com/media/en/technical-documentation/application-notes/AN-1079.pdf>



Xiaodong Zhao (M’13) received the B.Eng. and M.Sc. degrees in electrical engineering from the Harbin Institute of Technology, Harbin, China, in 2009 and 2011, respectively, and the Ph.D. degree in electrical engineering from Queen’s University Belfast, Belfast, U.K., in 2015.

He is currently a Research Fellow with Queen’s University Belfast. His current research interests include data acquisition for synchrophasor measurement, HVDC transmission, and power electronics application for wind energy integration.



David M. Lavery (M’10) was born in Belfast, U.K., in 1984. He received the M.Eng. and Ph.D. degrees from Queen’s University Belfast, Belfast, in 2006 and 2010, respectively.

He is currently a Lecturer with the Energy, Power, and Intelligent Control Cluster, Queen’s University Belfast. His current research interests include anti-landing detection, cyber-security and telecommunications, and synchrophasor measurement.



Adrian McKernan received the M.Eng. degree in electrical and electronic engineering and the Ph.D. degree from Queen's University Belfast, Belfast, U.K., in 2005 and 2010, respectively.

He is currently a Research Fellow with the Centre for Wireless Innovation, The Institute for Electronics, Communications, and Information Technology, Queen's University Belfast. His current research interests include wireless localization, experimental wireless channel measurements, and system sampling.



D. John Morrow (M'99) received the B.Sc. and Ph.D. degrees from Queen's University Belfast, Belfast, U.K., in 1982 and 1987, respectively.

Since 1987, he has been a Lecturer in Electrical Engineering with Queen's University Belfast. His current research interests include electric power systems, power system instrumentation, and embedded generation.

Dr. Morrow is a member of the IET and the PES Excitation Systems Subcommittee.



Kieran McLaughlin received the M.Eng. degree in electrical and electronic engineering and the Ph.D. degree from Queen's University Belfast, Belfast, U.K., in 2003 and 2006, respectively.

He is currently a Lecturer with the Centre for Secure Information Technologies, Queen's University Belfast, where he leads research in cyber security for smart grids, industrial control systems, supervisory control, and data acquisition networks. His current research interests include threat analysis, intrusion detection/prevention, and cyber-physical

resilience measures.



Sakir Sezer received the Dipl.Ing. degree in electrical and electronic engineering from RWTH Aachen University, Aachen, Germany, in 1994, and the Ph.D. degree from Queen's University Belfast (QUB), Belfast, U.K., in 2000.

From 1995 to 1998, he was with Nortel Networks, Northern Ireland, U.K., and Amphion Semiconductors, Northern Ireland, where he was involved in the development of ASICs, for access and high-speed transmission products. In 1998, he joined QUB to lead research and teaching in the areas of communication and digital systems. He is currently the Director and the Head of the Network and Cyber Security Research with the Centre for Secure Information Technologies, QUB, where he is also the Chair of Secure Communication Technologies.