An Original Smart Data Sampling for Wireless Sensor: Application to Bridge Cable Monitoring

VINCENT LE CAM, LAURENT LEMARCHAND,
ARTHUR BOUCHE, DAVID PALLIER and FRANÇOIS ILLIEN

ABSTRACT

Structural health monitoring (SHM) systems often ask for a robust, flexible and cost-effective solution. In that domain, since years, the technological development of Wireless Sensor Network try to be an answer. Between many other questions, one of the key-point in wireless sensing resides in the time synchronization (e.g. how to ensure the same time base between electronic systems that doesn't know each others ?). At Gustave Eiffel University, robust and deterministic solutions based on GNSS modules have already been demonstrated [1], the goal of the work presented in this paper is to go deeper into turn-key solutions by implementing and coupling this GNSS-synchronization principle into a low-power FPGA to an Analog-To-Digital converter. This hardware and software association represents a generic solution for signal sampling in a wireless manner. This work is illustrated and demonstrated by an application on the acoustic monitoring of wire-breaks in bridges cables.

INTRODUCTION

State of the art, open questions

As introduced above, the development of wireless sensor networks entails numerous practical and technological considerations that must be addressed individually. While the energy efficiency of wireless nodes is a crucial aspect, involving questions such as power consumption reduction, energy storage, and energy harvesting, other factors often go unnoticed. For instance, it is essential to account for the metrological aging of electronic systems deployed on structures over extended periods. Additionally, optimizing the placement of wireless sensors on structures (considering energy efficiency and radio reception) is another overlooked aspect. Gustave Eiffel laboratories have dedicated

Vincent Le Cam, Research Engineer, co-head of CoSys department, head of SII Laboratory at Gustave Eiffel University, France.
Laurent Lemarchand, Arthur Bouche, Research Engineers, SII Laboratory at Gustave Eiffel University, France.
David Pallier, Research Engineer, CTO of Nav4You, Alle´e des ponts et chausse´es 44340 Bouguenais. François Illien, PhD position, LS2N Laboratory, 44322 Nantes Cedex 3

extensive research to wireless synchronization, aiming to provide a robust and deterministic solution to address the practical question: *How can we ensure a consistent time base in a network of wireless electronic nodes that are totally independent?*

Use-case illustrations

Wireless synchronization of nodes poses a significant challenge in structural health monitoring (SHM) scenarios due to the following factors: continuous monitoring requirements, deterministic accuracy (with no tolerance for approximations or variations in data sampling quality or time-stamping accuracy), and the potential for long distances between nodes ranging from tens of meters to several kilometers. In addition, except for certain physical quantities such as temperature or water level, many SHM use-cases require accurate and high-rate data sampling. To illustrate, let's consider three examples along with their respective order of magnitude:

- In structural vibration monitoring, typical frequencies range from a few Hertz to several tens of Hertz. Wireless nodes must maintain a time-base accuracy of a few milliseconds [3].
- In acoustic wave propagation (mechanical waves, ultrasonic waves, etc.), typical data sampling rates are in the range of 100 kHz to 1 MHz. Consequently, wireless acoustic sensors need to ensure a time-base accuracy of a few microseconds.
- In optical or electrical wave propagation, the required accuracy typically reaches the order of a few nanoseconds.

Objective

In this article, we first review the principle of node synchronization using GNSS receivers. We then demonstrate the integration of this solution with a 12-bit ADC, illustrating the design of a complete turn-key electronic solution. The resulting system provides timestamped samples at frequencies of several hundred kilohertz, with a timestamping accuracy of a few microseconds.

GNSS SYNCHRONIZATION PRINCIPLE

Wireless node's synchronization through a GNSS module (typically using a GPS receiver) is a well-known principle [1] [2]. Referring to the figure 1, this solution can be summarized as following:

- A GPS receiver is connected to a microprocessor that could be as well an ARM processor or an FPGA...
- Two main signals are important from the GPS receiver as inputs to the processor:
  - First: The PPS (Pulse Per Second) is a binary signal that marks the beginning of a new GPS second. It exhibits a typical accuracy of 20 to 50 nanoseconds relative to the GPS time.

  - Second : a serial port is utilized to transmit NMEA frames between two PPS signals, which carry essential information such as position and GPS time corresponding to the next PPS rising edge.

- Inside the processor/FPGA, the PPS signal is directly connected to the *reset* input of a time counter.
- Inside the processor, a software (or hardware on FPGA) block is dedicated to standard NMEA frames decoding [1] [5]. Those ASCII informations (position, GPS time...) are updated each second, on PPS edge.

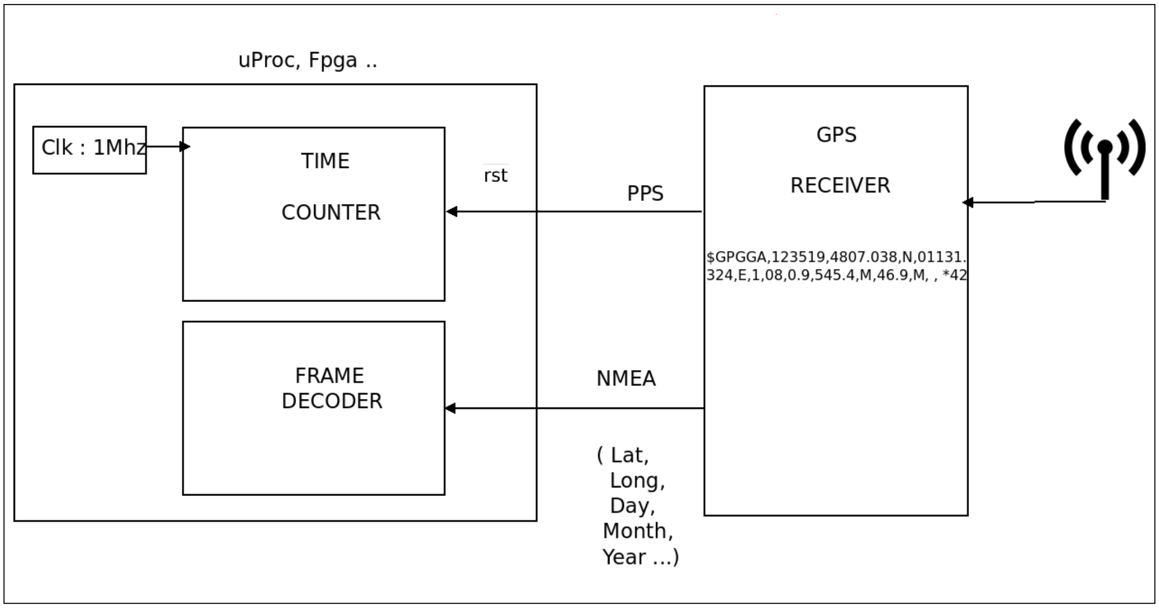


Figure 1. Node architecture for a typical GPS synchronization

What is important to retain is the fact that the time-counter, reset on each PPS every second, acts as the time reference of the wireless node. Even if this time-counter is clocked by a clock based on a physical quartz component which isn't perfect [4], the drift cannot influence our timing for more than one second. In other words, the time-counter is enslaved to the PPS accuracy. For instance, if using a quartz at 100 MHz with a given accuracy of +/-50 ppm, thus after 1s, the drift of the time-counter remains in the interval of [99 995 000 ; 100 005 000] strokes. As illustrated in the figure 2 in red lines: the drift cannot influence on more than 1 second. Thus, the maximum time-stamping error of a same event between two wireless sensors S_i and S_j will be:

$$||Ti–Tj|| \leq 100\mu s \tag{1}$$

Finally, for any event that can occurs (a rising edge on a GPIO port, a sample delivered on an SPI port by an Analog-to-Digital Converter...), on the corresponding interruption, the microprocessor/FPGA just has to copy and register the value of the time-counter in a register. It represents the absolute time with the resolution of the timer.

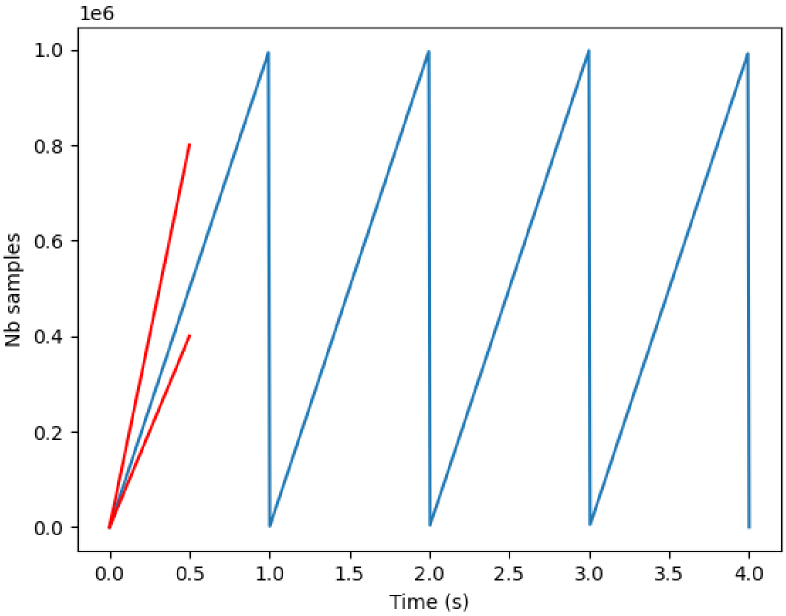


Figure 2. Time-counter drift reset on each PPS edge

This principle is implemented and evaluated on various architectures in the referenced PhD work [1]. Additionally, the work explores further advancements achieved by integrating the following optimizations on FPGA:

- In addition to NMEA standard frames, the decoding block also takes into account the quantitative error (Q_{Err}) provided in the proprietary UBX frame on u-blox GNSS receivers. These quantitative errors enable the correction of the PPS signal resulting in an accuracy of approximately 10 ns relative to the GPS time.
- With the highly accurate PPS signal serving as a time reference, the FPGA can determine the practical ”real” frequency of its clock. As most quartz-based clocks exhibit relatively stable behavior over short-term periods (ranging from tens of seconds to minutes), it is feasible to maintain good timing accuracy during these intervals without relying on the PPS signal. Consequently, a mechanism can be implemented to periodically activate and deactivate the GPS receiver, thereby reducing its power consumption.

COUPLING A 12 BITS ADC TO A LOW POWER FPGA

Theory and implementation

The figure 3 illustrates the concept. The previous synchronization design is implemented on a TinyFPGA ICE40 [7] from Lattice completed with:

- a 12 bits ADC continuously providing samples on an SPI port. Here the low-power AD7450 from Analog Device [6] can sample up to 1 MegaSamples/s
- a 16 bits SPI port implementation in Verilog language where final data will be output (see format below)

The core of the concept resides in a periodic association of a sample of 12 bits to its time-stamping (e.g. its corresponding value in the time-counter). The time-counter value can reach 16000000 counts/s as the FPGA internal clock is running at 16 MHz. Thus, any time-stamping of any sample coming from the AD7450 needs 24 bits to be encoded (as 24 bits allow counting up to 16777216 beats per second).

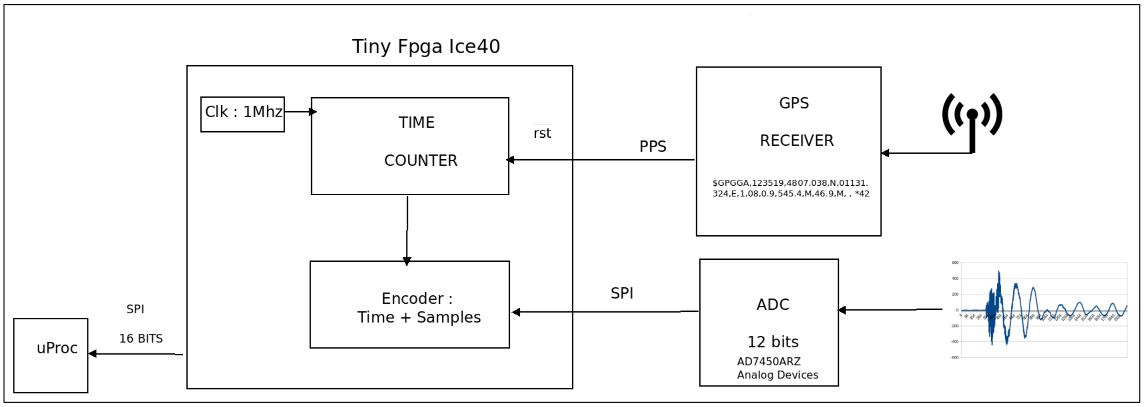


Figure 3. Logical schematic integrating the 12 bits ADC

The *Time + Samples* encoder block inside FPGA (figure 3) applies the following principle: each sample from the AD7450 ADC is registered at the desired sampling-frequency, but, only one-on-eight samples time stamping is delivered as it ask for 8 × 3 bits to output the timestamping. One bit is necessary to encode if (1) a new timestamping is beginning or (0) for a completion of it. The format of the output of the FPGA is described in figure 4 and table I. The timestamping bits are noted b_x^t and the data bits from the ADC are noted b_x^d.

TABLE I. FPGA data output format

New time-stamping 1 bit	Time-stamping 3 bits	12 bits data from ADC
1	$b_0^t b_1^t b_2^t$ of n	b_{11}^d ... b_0^d of n
0	$b_3^t b_4^t b_5^t$ of n	b_{11}^d ... b_0^d of $n+1$
0	$b_6^t b_7^t b_8^t$ of n	b_{11}^d ... b_0^d of $n+2$
0	$b_9^t b_{10}^t b_{11}^t$ of n	b_{11}^d ... b_0^d of $n+3$
0	$b_{12}^t b_{13}^t b_{14}^t$ of n	b_{11}^d ... b_0^d of $n+4$
0	$b_{15}^t b_{16}^t b_{17}^t$ of n	b_{11}^d ... b_0^d of $n+5$
0	$b_{18}^t b_{19}^t b_{20}^t$ of n	b_{11}^d ... b_0^d of $n+6$
0	$b_{21}^t b_{22}^t b_{23}^t$ of n	b_{11}^d ... b_0^d of $n+7$
1	$b_0^t b_1^t b_2^t$ of $n+8$	b_{11}^d ... b_0^d of $n+8$

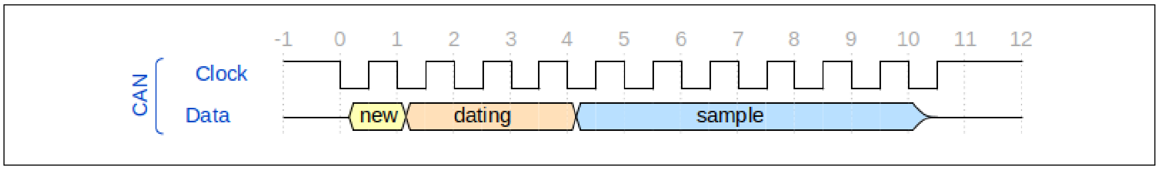


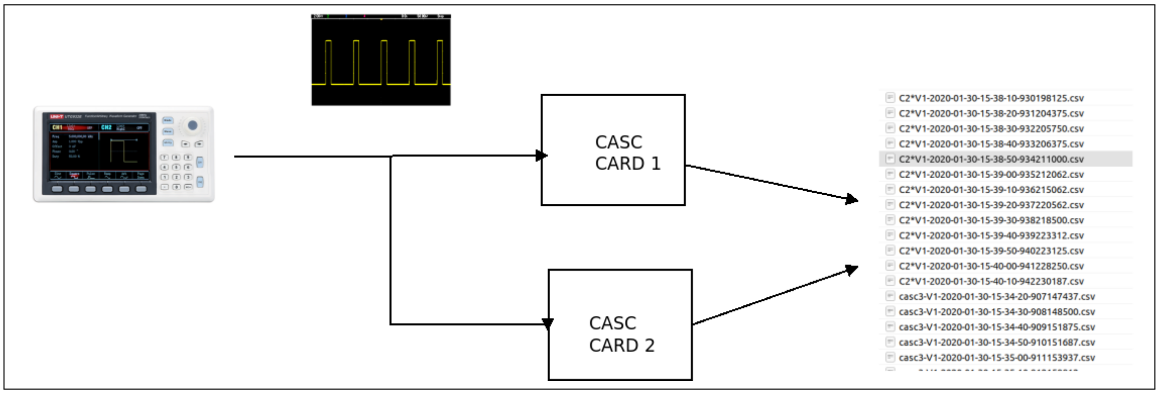
Figure 4. Chronogram of one sample on the 16 bits SPI port

Time-stamping bit to gps-time

The time-stamping bits are derived from the time counter inside the FPGA. To convert these bits into GPS time, a calculation is required. Firstly, to timestamp samples on the scale of a second, the processor utilizes its current GPS time. Secondly, for sub-second timestamping, the FPGA's clock theoretical frequency of 16 MHz is used. Consequently, a counter value of 0 corresponds to the start of the second, while a counter value of approximately 16 million corresponds to the end of the second. Linearization of the counters is then performed (for example: a counter value of 4 million represents approximately a quarter of a second). Since only one out of every eight samples is timestamped, a linear interpolation is conducted to determine the timestamps of the seven remaining untimed samples. This interpolation is possible due to the relative stability of the quartz and SPI frequencies over short periods.

Results

In order to validate the *Smart Converter* principle, the following test configuration has been applied : directly on the analog inputs of two separate wireless sensors, a step signal is applied. Each wireless sensor records a frame at a sampling frequency of 185 kHz. For each step signal, both wireless sensors transmit their frames to a Supervisor where the timestamps of the same event, as recorded by **two independent wireless sensors**, are compared. The event corresponds to the detection of a rising edge on the signal using a threshold. This setup is illustrated in figures 5 and 6.



```
C2*V1-2020-01-30-15-38-10-930198125.csv
C2*V1-2020-01-30-15-38-20-931204375.csv
C2*V1-2020-01-30-15-38-30-932205750.csv
C2*V1-2020-01-30-15-38-40-933206375.csv
C2*V1-2020-01-30-15-38-50-934211000.csv
C2*V1-2020-01-30-15-39-00-935212062.csv
C2*V1-2020-01-30-15-39-10-936215062.csv
C2*V1-2020-01-30-15-39-20-937220562.csv
C2*V1-2020-01-30-15-39-30-938218500.csv
C2*V1-2020-01-30-15-39-40-939223312.csv
C2*V1-2020-01-30-15-39-50-940223125.csv
C2*V1-2020-01-30-15-40-00-941228250.csv
C2*V1-2020-01-30-15-40-10-942230187.csv
casc3-V1-2020-01-30-15-34-20-907147437.csv
casc3-V1-2020-01-30-15-34-30-908148500.csv
casc3-V1-2020-01-30-15-34-40-909151875.csv
casc3-V1-2020-01-30-15-34-50-910151687.csv
casc3-V1-2020-01-30-15-35-00-911153937.csv
```

Figure 5. Testbench

The timestamping differences between the two nodes for a thousand events are summed in table II. The absolute maximum timestamping difference is around 6 μs which pro-

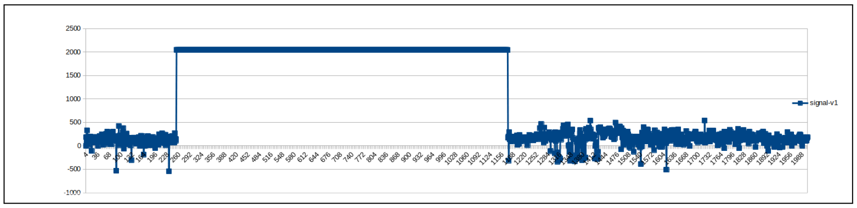


Figure 6. A typical frame from the applied test. Step signal

vides sufficient accuracy for acoustic wave monitoring.

TABLE II. Experimental results

Average	Absolute Maximum	Absolute Minimum	Standard deviation
3.225 μs	6.187 μs	0.5 μs	2.06 μs

Future testbench

To further enhance the demonstration, a dedicated testbench for real structural health monitoring (SHM) with a focus on acoustic wave monitoring has been developed (Figure 7). Wireless sensors are installed at each end of a piece of bridge cable. By simulating the breakage of a single steel rope using a hammer-induced shock, the Time Difference of Arrival (TDOA) algorithm is employed to determine the source of the break. Our solution will be tested on this testbench in future work.

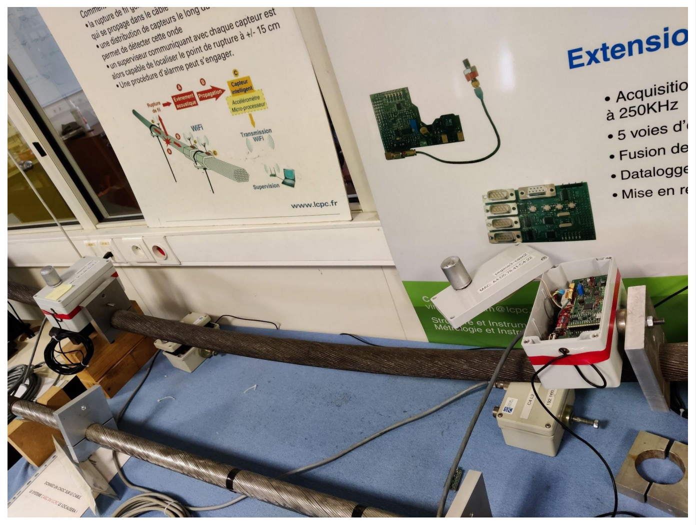

• la rupture de fil
qui se propage dans le câble
• une distribution de capteurs le long
permet de détecter cette onde
• un superviseur communiquant avec chaque capteur est
alors capable de localiser le point de rupture à +/- 15 cm
• Une procédure d'alarme peut s'engager.
Rupture de fil
Evènement acoustique
Propagation
Capteur intelligent
Accéléromètre Micro-processeur
WiFi
Transmission WiFi
Supervision
www.lcpc.fr
Extensio
• Acquisitio à 250KHz
• 5 voies d'
• Fusion de
• Datalogge
• Mise en r

Figure 7. The acoustic waves testbench using two wireless sensors

CONCLUSIONS AND PERSPECTIVES

The present article aims to provide a comprehensive implementation of the well-known GNSS synchronization principle in conjunction with a 12-bit ADC. The proposed solution entails a unified Smart Data Sampler package that incorporates an FPGA, a GPS receiver, and an ADC. For seamless integration, any microcontroller platform compatible with this solution only needs to offer a 16-bit SPI port and decode the samples according to the specifications outlined in Table I. An initial implementation of the SHM technique is presented for acoustic wave time sampling, where two wireless sensors sample the acoustic waves within a bridge cable, yielding an average error of 3.22 µs. In terms of future prospects, there are two main objectives: improving time-stamping accuracy and conducting an energy balance analysis of the system. The improvement of time-stamping can be achieved through two approaches. Firstly, by utilizing the actual frequency of the FPGA, considering that the theoretical frequency of 16 MHz may not be accurate. Secondly, by leveraging the quantitative error (Q_{Err}) to correct any inherent inaccuracies in the PPS signal itself. Regarding energy considerations, the Lattice Ice40 FPGA and the ADC converter are low-power components, while the GNSS receiver consumes a higher amount of power. Previous works [1][2] have demonstrated the feasibility of maintaining highly accurate time synchronization even when the GPS receiver is turned off. Therefore, future research will focus on assessing the energy balance of this solution by implementing a periodic on/off mode for the GPS module.

REFERENCES

1. David Pallier. PhD. 2021. ”Sensor Enhancement to Augmented Usage and Reliability”
2. D. Pallier, V. Le Cam, S. Pillement, ”Energy-efficient GPS synchronization for wireless nodes”, IEEE Sensors Journal, 2020
3. V. Le Cam, M. Döhler, M. Le Pen, I. Guéguen et L. Mevel, “Embedded subspace-based modal analysis and uncertainty quantification on wireless sensor platform PEGASE”, in Proc. EWSHM, p. 1705-1715 (cf. p. 3, 26, 121).
4. F. L. Walls et J. R. Vig, “Fundamental limits on the frequency stabilities of crystal oscillators”, IEEE Trans. Ultrason., Ferroelectr., Freq. Control, t. 42, no 4, p.576-589, juil. 1995. doi : 10.1109/58.393101 (cf. p. 12, 13).
5. V. Le Cam, A. Bouche et D. Pallier, “Wireless Sensors Synchronization : an accurate and deterministic GPS-based algorithm”, in Proc. IWSHM (cf. p. 26, 29, 121).
6. Reference to the ADC7450 of Analog DEvice. https://www.analog.com/media/en/technical-documentation/data-sheets/ad7450.pdf
7. Reference to the Tiny FPGA ICE40 from Lattice.
   https://www.latticesemi.com/Products/FPGAandCPLD/iCE40Ultra