Improving wireless sensor behavior by means of generic strategies

Vincent Le Cam, Laurent Lemarchand, William Martin, Nicolas Bonnec

Cover page

Title: **Improving wireless sensor behavior by means of generic strategies**.

Principal author: Vincent Le Cam [1]
Co-authors: Laurent Lemarchand [1]
William Martin [1]
Nicolas Bonnec [1]

Abstract

LCPC instrumentation division [1] has developed a generic wireless platform that can be considered as the a result of redundant needs in wireless monitoring especially applied to civil engineering monitoring applications. This platform includes software and hardware bricks and aims at being generic by its native implementation of sober components, the worldwide TCP/IP protocole (802.11g), a signal processor, a small GPS receiver, and a micro embedded operating system (uClinux).

Since 2009, this platform -named PEGASE - is subject of an industrial transfer that has generated some tens of indivual sales. A set of pluggable daugthers boards (that integrate the application specific sensing operation) offers a ready-to-use panel of wireless sensing solutions for developping specific applications as well as they can be seen as prototyping boards for further electronic developments.

As PEGASE platform reached a mature level of dissemination, LCPC recent efforts are now leaded with the goal of improving its wireless capacities. Those works concern energy saving while keeping a high level of embedded processing, of sampling rate or time-synchronization.

After a quick summary of PEGASE properties, this article aims at presenting the last algorithms and evolutions that have been developped and added to the system. As software layers are mainly written in standard C language under Linux OS, those pragmatic solutions could easily be re-used by even radically different systems. The focus will specifically be pointed on: an algorithm that allows PEGASE wireless boards to be synchronized up to some uS using a GPS technique while keeping the GPS receiver OFF most of the time; a description of how the use of an operating system such as uClinux allows a full and remotely uptade of wireless sensors; the hardware and software strategies that have been developped to make PEGASE fully autonomous using solar cells.

[1]LCPC Instrumentation Division, Laboratoire Central des Ponts et Chaussées, route de Bouaye, BP 4129, 44341 Bouguenais Cedex, France.

INTRODUCTION

Considering the exponential technological developments especially in electronic domain, a question could be: *how to pose the concept of a generic wireless platform in a world in constant evolution ?* Moreover, electronic components lifespan appears as an *epsilon* part of the expected structures lifespan (bridges, railway, industrial machines, etc.). So how to get pragmatic solutions that could reach the challenges that SHM ask ?

The ideas and the concepts defended in this article try to answer those questions by separating the solution in two part. One part of the answer resides in the most intelligent choice possible of an electronic solution. This means sober and resisting components, a relevant processor, etc. But no eternal electronic devices such as processors, memories, etc. have been designed for good yet. Consequently, thinking new hardware solutions while keeping an active technological survey in electronic and WSN domains will remain necessary for years. But a part of the answer is to extract, as much as possible, the generic concepts that can be reused independently from hardware solutions.

In a previous article [1] -as other laboratories or companies did [3,8,9]- LCPC strongly supported the idea of generic wireless sensor platform using the *best of* the available technology. Considering the heavy cost of any electronic development (especially in the wireless world), developping generic and easily reusable hardware platforms is considered as relevant. From the first successes that LCPC wireless platform (PEGASE) met, the idea has been to factorize and to capitalize improvement strategies that can be applied to PEGASE and that could also be applied to other wireless platforms.

LCPC WIRELESS PLATFORM PROPERTIES

First of all, as many other entities deciced [5, 7], LCPC wireless platform have been made modular to fit the need of beeing reusable without any expensive and long developments. This modularity resides in a sensor made of a "mother board" that integrates redundant needs (computation, communication, data storage) and a set of pluggable "daughter board" that integrates application specific needs such as the sensitive element (accelerometer, temperature, pressure,...), its specific conditioning design including hardware filters, analog to digital converters.

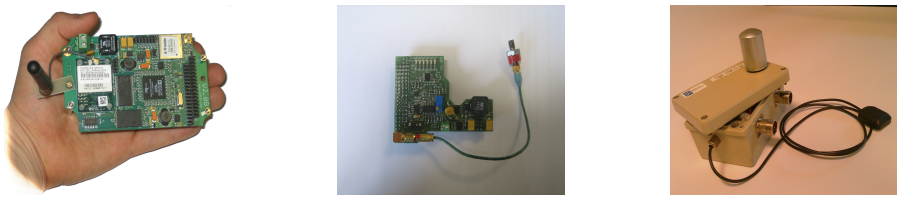


Figure 1: PEGASE mother board, a daughter board, a specific solution

As justified in [1], the main characteristics of PEGASE feature are the following:

- Use of TCP/IP/WiFi as the wireless protocol [4,10]: reliable, low-cost, scalable (IP is the worldwide protocol). Turned OFF when PEGASE doesn't communicate.

- Use of the Analog Device low-power Blackfin BF537 as core processor (Digital Signal Processor): 16 bits processor able of complex operations.
- Implementation of a small and low-power GPS receiver to ensure localization and, first of all, absolute time synchronization up to few μS GMT.
- uClinux as the embedded operating system: allows high level of abstraction while PEGASE algorithms are then programmed using standard ANSI C language.

According to application needs, PEGASE power consumption could vary between 400 mW (full mode ON) to some 80 mW (in sleep mode). As described below, even if this energy consumption is not the lowest in WSN world, this level allows viable configurations based on batteries coupled to solar cells.

PEGASE APPLICATIONS

Since its first version on january 2008, PEGASE has been used in various configurations where its properties fitted specific needs. To illustrate that dissemination, three applications are described below.

Bridge inspection by UAV In order to simplify structures inspections and to make them less expensive, one of LCPC projects refers to the instrumentation of an Unmanned Aerial Vehicle. A military U.A.V. has been bought and its instrumentation by PEGASE platform has the goal to interface many devices that could help human subjectivity. Those devices are: GPS receivers, a camera controled by a 3-axial robot, a laser-meter.

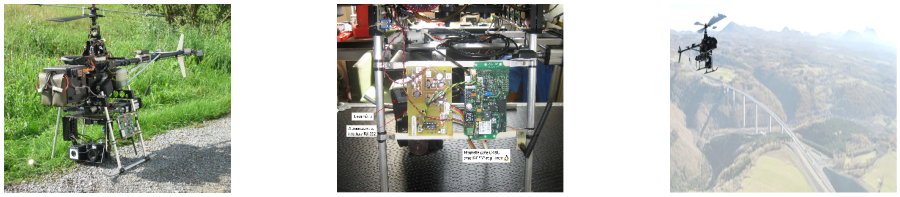


Figure 2: PEGASE platform for U.A.V. instrumentation

Cable health monitoring As already described [1], cable health monitoring is a LCPC major preoccupation. One of the favorite method used to detect and to localize wire-breaks in cables (due to corrosion or mechanical fatigue) is based on the acoustic emission of those breaks and the elastic wave thus generated. In that case, PEGASE is customized in a sensor whose mission consists in permanently *listening* using an accelerometer. For each detected wave (comparison to a threshold), an asbolute time-stamping is made and information is wirelessly transmitted to a remote supervisor. Knowing wireless sensors location, waves spectrum and dating, supervisor is then able to localize the wire-break and to warn an operator via a GSM modem.

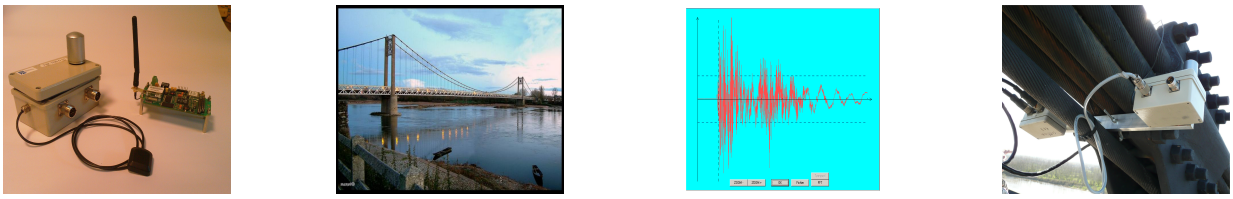


Figure 3: PEGASE platform customized for acoustic monitoring of cables

Continous meteorologic monitoring PEGASE has been used and customized in order to implement specific needs relative to meteorologic measurements close to structures such as bridges or dams. In that application, the goal is the validation of models measuring the influence of wind gradient on noise propagation. Real-time monitoring processes associated to accurate time stamping and DSP capacities of PEGASE are involved. As figure 4 shows, a specific daugther board emulating 5 RS232 ports have been designed in order to be connected to 5 anemometers. The final product takes place into a small box.

Figure 4: PEGASE involved in meteorological measures

PLATFORM IMPROVEMENTS

Using GPS for time-synchronisation and quartz drift correction

Wireless Sensor Network time synchronization is a real challenge [2]. Because of quartzes drift (natural error, aging, t^0c), very few determinist methods propose an accuracy better than mS and, if any, they often suppose numerous and energy-expensive communications with a supervisor sending beacons. A solution implemented in PEGASE concept, uses the propensity of any GPS receiver to deliver a digital signal named PPS (Pulse Per Second). Each PPS from each GPS receiver is synchronised with an accuracy of 1 S +/- 20 nS (i.e. $20*10^{-9}$S). The PPS output of the GPS receiver is routed to the RESET entry of DSP time-counter. In parallel, the standard RS232 output of the GPS receiver that contains absolute GPS informations (localization, date, hour, minutes, seconds) is routed to a RS232 input port of the DSP. A real-time algorithm continuously decode those informations. Then, on PPS rising edge, a Linux real-time interruption resets microseconds counter and updates latest GPS information decoded (refreshed until next second).

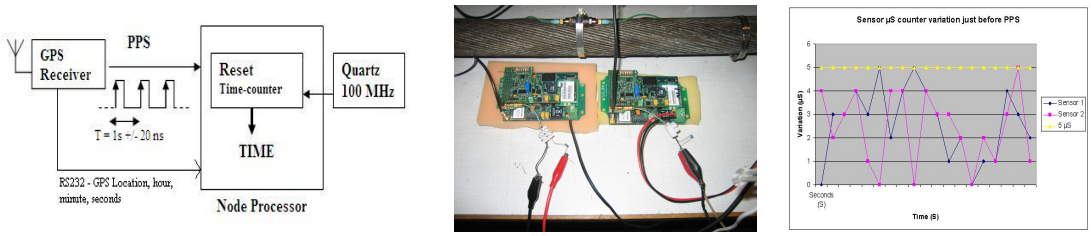


Figure 5: GPS synchronization principle, test-bench and results.

The result is that, between two PPS edges, any function, any process, any event such as a digital to analog conversion can be time-stamped by a C code instruction that gets the actual value of the time-counter. Testing campaigns and measurements gave, for a same event, **a time stamping difference between two PEGASE wireless sensors doesnt exceed 15 μS**, while keeping a GMT referential thanks to GPS standard.

Increasing synchronization accuracy by correcting quartz drift. Once previous PPS synchronization method was perfectly under LCPC control and knowledge, a second level of synchronization has been implemented in PEGASE algorithms. Considering that PPS accuracy (about 20 nS) is much more sufficient than whised accuracy in typical SHM applications (where synchronization needs vary from mS to some μS), time-counter values (cf. figure 5) have been analysed on PPS rising edge. As far as its value on PPS edge is supposed to be equal to 1000000 μS a linear difference can be observed from the begining of a second (theorical time-counter = 0μS) until the end of that second (theorical time-counter = 10^6 μS). This linear difference nammed δ is due to PEGASE quartzes drift (natural factory error) and, consequently, very constant.

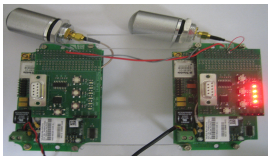
Then a specific algorithm has been developped and added to the previous one to complete PEGASE synchronization principle: that algorithm is called by the uClinux low-level interrupt mechanism. On each PPS interrupt, δ is evaluated as the difference between the theorical value of 10^6 μS and time-counter real value (influenced by quartz error). This value is added to a 16 places table that is averaged, this gives the $\Delta 16$ average drift of the quartz. On each PPS event (i.e. each second):

$$\delta = 1000000 - time\_counter$$

$$\Delta 16 = \frac{\Delta 16 + \delta}{16}$$

Then, any event that occurs in PEGASE system is time-stamped reading the time-counter and applying a linear correction that takes $\Delta 16$ drift into account:

$$Event\_uS = \frac{time\_counter * 10^6}{10^6 - \Delta 16}$$



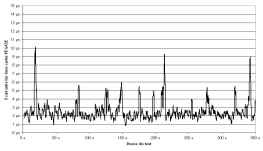


Figure 6: GPS second test-bench and improved results.

Testing campaigns measurements gave, for a same event, including the previous time correction **an absolute time stamping differencebetween two PEGASE wireless sensors doesnt exceed 4 μS**, while keeping a GMT referential thanks to GPS standard.

Turning GPS OFF for power saving while keeping synchronised

The GPS based synchronization principle could be subject of a criticism: even a low-power GPS receiver needs some 25 mW. The present improvement proposes to periodically turn OFF the GPS receiver in order to cancel its power consumption. The postulate resides in the fact that, after short periods (around 50 seconds), the Δ16 average drift of the quartz is well known by the system, and, aging and temperature influences on quartz are null during short periods.

The idea is, once Δ16 is evaluated, a new time-counter is immediately start; its period is fixed to $\{ 10^6 - \Delta 16\}$ μS. This period is the real value of what represent the delay of 1s view by PEGASE board using a specific quartz at a specific moment. Then the GPS receiver could be stopped and time-stamping operations are then made by the system using this new timer that acts as a relay while GPS receiver is OFF.

This algorithm has also been implemented and works. First results give a linear response in the possibility to stop the GPS receiver while keeping a quite good accuracy in sensor synchronization (the final drift is due to the uncertainty of the OS and the residual errors in Δ16 evaluation). While keeping the GPS ON during 60 S and then OFF during 600 S, the sensor has drifted of 270 μS when GPS has been rearmed. Other developments in that way are under works.

A collection of reusable software bricks

As mentionned, uClinux implementation as the embedded operating system in PEGASE allows high level of abstraction while PEGASE algorithms are then programmed using standard ANSI C language. No specific knowledge about the BF537 processor registries is necessary since all its peripherals can be driven through uClinux libraries. More than the simple use of standard C language, LCPC engineers have developped a collection of classes (i.e. software bricks) written in C-object language.

This library named UFC (for Uclinux Foundation Classes) offers a usefull dozen of directly reusable classes (.c & .h files) to easily manage the main PEGASE peripherals: CGPS, CAbsoltuTime, CSPI (to interface the SPI port and then Analog-to-Digital converters), CTcpIPClient, CTcpIPServer, CWiFiMngt, CRS232port, etc.

Those generic classes associated to the PEGASE hardware natural genericity accelerate the customization of PEGASE to develop a quick answer to a specific application. Moreover, the C knowledge is quite universally diffused and makes the concept easily adopted by designers.

Using solar cells for energy harvesting

According to the application needs, a strategy must be defined in terms of energy. For short term applications (from few hours to few weeks) effectives solutions could be found by means of batteries (Lead or NiMh technologies). In those cases, for uncontinuous monitoring (low sampling and processing), hibernate modes can be applied. PEGASE platform can reduce its power consumption to 80 mW. On the other hand, for long term monitoring (months, years) or applications that require continous processing or numerous wireless communications, solutions using solar cells could be defined

For PEGASE wireless platform, a specific *smart* solar cell solution has been defined. Many solutions that use solar cell as an auxiliar source: while solar cell voltage is less or equal than battery voltage, wireless system uses energy from battery, when battery voltage is exceeded, solar cell energy is used. The optimization consists in the integration of a boost circuit. If the boost circuit is correctly designed, each voltage value that output from the solar cell is widely amplified. All the energy that the solar cell is able to deliver is amplified, the battery threshold is exceeded without time loss waiting for greater voltage. Such a boost circuit is integrated to PEGASE concept.

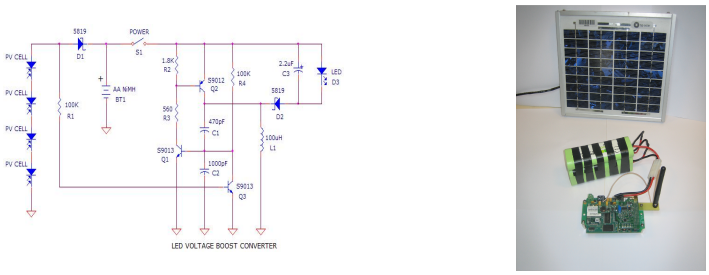


Figure 7: A boost circuit added to the PEGASE concept for an optimal solar cells source

This hardware optimization is driven by a software that implements a Maximum Power Point Tracking algorithm (MPPT [6]). Continuously and automaticaly, PEGASE finds and maintain the optimum level of energy that can be extracted from the solar cell source. First results gave an acceptable size of 30cm * 30cm solar cell (cf figure 7) for an application of continuous monitoring that requires 250 mW.

CONCLUSION AND ROADMAP

Since a third-party partner (A3IP company) has been licencied by LCPC, PEGASE has been sold in hundreds of specimens and implemented in various configurations. This dissemination prooved the capacity of wireless systems to really answer a large spectrum of applications. Developments in progress have the goal to increase this panoply. Even if

uClinux and WiFi integration could be considered as *heavy*, the result is a great ability for developpers or customers to acheive their own applications. The genericity of C language and the worlwide IP protocol make them ubiquitous.

A quite expert job has been leaded to develop specific embedded drivers under uClinux OS in order to get specific behaviors for time synchronization, quartz drift auto-training and correction. This specific and dynamic correction takes temperature effects into account and the result is an absolute time synchronisation better that 4 μS.

Even if technologies evolve (components, processor, batteries...), generic principle could be extracted independently from technological choices. Those main principles are: daughter/mother boards, Linux integration, a ready to use c-object library, a boost circuit linked to a MPPT algorithm, GPS synchronization and quartz correction. Most of the improvements can be reused and applied to other wireless platforms even using drastically different electronic implementations.

References

[1] Vincent Le Cam Laurent Lemarchand Louis-Marie Cottineau Frédéric Bourquin. Design of a generic smart and wireless sensors network – benefits of emerging technologies. *Structural Health Monitoring 2008*, 1(1):598–605, 2008.

[2] Vincent Le Cam Laurent Lemarchand Louis-Marie Cottineau. Synchronization of wireless sensors : review of methodologies. experience feedback of the very precise gps solution. *Structural Health Monitoring 2006*, 1(1):1339–1347, 2006.

[3] M. David Culler; Estrin, D.; Srivastava. Overview of wireless sensor networks. *Smart Environments: Technologies, Protocols, and applications*, 37(8):41–49, 2004.

[4] Adam Dunkels, Thiemo Voigt, and Juan Alonso. Making tcp/ip viable for wireless sensor networks. *Proceedings of the First European Workshop on Wireless Sensor Networks (EWSN 2004)*, 2004.

[5] A. Vecchio H. Van Der Auweraer E. Moya T. Torfs, B. Peeters. Wireless sensor developments for structural health monitoring. *Structural Health Monitoring 2008*, 1(1):606–613, 2008.

[6] Lehman Brad Florent Boico. Single sensor mppt algorithm for multiple solar panels configurations.

[7] R.A. Swartz J.P. Lynch. Dsitributed data processing architectures for structural health monitoring systems implemented on wireless sensor networks. *Structural Health Monitoring 2008*, 1(1):21–32, 2008.

[8] F. L. Lewis. Wireless sensor networks. *Smart Environments: Technologies, Protocols, and applications*, 2004.

[9] Henry O. Marcy. Wireless sensor networks for area monitoring and integrated vehicle health management applications. *Rockwell Science Centre 2003*, 2003.

[10] Ariz Scottsdale. Wifi moves into the sensor networking. *Business Wire*, 2008.