Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2014, Article ID 473905, 8 pages
http://dx.doi.org/10.1155/2014/473905

*Research Article*

# Cognitive Wireless Sensor Network Platform for Cooperative Communications

**Agustín Tena, Guillermo Jara, Juan Domingo, Elena Romero, and Alvaro Araujo**

*Universidad Politécnica de Madrid, ETSI Telecomunicación, 28040 Madrid, Spain*

Correspondence should be addressed to Agustín Tena; agus@die.upm.es

Received 15 July 2013; Revised 22 November 2013; Accepted 6 December 2013; Published 8 January 2014

Academic Editor: Shukui Zhang

Nowadays, Wireless Ad Hoc Sensor Networks (WAHSNs), specially limited in energy and resources, are subject to development constraints and difficulties such as the increasing RF spectrum saturation at the unlicensed bands. Cognitive Wireless Sensor Networks (CWSNs), leaning on a cooperative communication model, develop new strategies to mitigate the inefficient use of the spectrum that WAHSNs face. However, few and poorly featured platforms allow their study due to their early research stage. This paper presents a versatile platform that brings together cognitive properties into WAHSNs. It combines hardware and software modules as an entire instrument to investigate CWSNs. The hardware fits WAHSN requirements in terms of size, cost, features, and energy. It allows communication over three different RF bands, becoming the only cognitive platform for WAHSNs with this capability. In addition, its modular and scalable design is widely adaptable to almost any WAHSN application. Significant features such as radio interface (RI) agility or energy consumption have been proven throughout different performance tests.

## 1. Introduction

According to the CISCO report [1], ubiquitous computing is shown as one of the most important trends with a 40-fold increase between 2010 and 2015. Typical ubiquitous applications include security and surveillance, automatic monitoring of forest fires, avalanches, hurricanes, health or vehicular networks. WAHSNs provide a technological solution to these challenges, so their growth is closely linked to these data.

This increasing demand for wireless communication presents a challenge to efficient spectrum utilization and spectral coexistence. Regarding spectrum scarcity, most WAHSN solutions operate on unlicensed frequency bands. In general, they use the industrial, scientific, and medical (ISM) bands like the 2.4 GHz band also used by Wi-Fi or IEEE 802.15.4 devices. For this reason, the unlicensed spectrum bands are becoming overcrowded [2].

To address this challenge, new techniques arise to be applied in WAHSNs, such as Cognitive Radio (CR) [3]. CR enables opportunistic access to the spectrum through cooperation and context awareness. Spectrum sensing capabilities and cooperation between devices allow for better spectrum use and better data reliability. The concept of CN was proposed as a wireless network that is aware of its environment and adapts its internal values, thanks to cooperation, to achieve reliable and efficient communication. CN has three main technical components: cognitive capabilities of devices, collaboration among terminals, and learning about the history.

WAHSNs are one of the areas with the highest demand for cognitive networking due to overcrowded unlicensed bands [4, 5]. Introducing CR capabilities in sensor networks structures provides benefits by implementing a more efficient and dynamic way to use the spectrum. However, there are also some challenges to face, specifically the study of the sensing state, collaboration among devices, energy efficiency, and decision making.

But, despite the potential of CWSNs, they are not yet deeply explored. Real or simulated scenarios scarcely exist and realistic platforms are crucial to the improvement and development of this new field. The shortage of CWSN devices or test beds contributes to the scarcity of results in this area.

WAHSN is a mature area, and platform devices and test beds for research and even real applications are common, but on the other hand, realistic devices with access to different spectrum zones for CWSNs do not exist. In order to enable and promote this new paradigm, an effort to facilitate the technology with devices and test beds is necessary. Therefore, in this work, a new platform for CWSN development is proposed. It consists of a versatile and modular WAHSN device which provides access to different spectrum bands.

When designing CWSN devices, the fact that WAHSN nodes are very limited in terms of memory, computational power, or energy consumption is crucial and must be taken into account in the design process. Hence, the architecture must be cost saving, meaning that the processing resources of the microcontroller (MCU) are finite both in raw processing as well as in memory allocation for data. As a development platform, the software architecture must provide advantages when implementing cognitive strategies in order to be a valuable tool. The design must be modular, so it will not need to be redone entirely with every particular change, and scalable, allowing flexibility in the complexity of applications.

This cognitive architecture is based on the model proposed in collaboration with the Berkeley Wireless Research Centre and the Telecommunication Networks Group at TU Berlin [6]. This model, Connectivity Brokerage, describes a scheme for cognitive networks deployment.

The organization of the paper is as follows. Section 2 is a general view of the state of the art and it describes the need for this work. A full hardware description of the platform is shown in Section 3. Sections 4 and 5 describe the software modules for hardware and cognitive algorithms control. In Section 6, results of a set of performance tests are exposed and the paper ends with some conclusions in Section 7.

## 2. Related Work

Because of the novel stage of this research field, there are no many specific devices to build applications and services over CWSN. It is natural that most works are based on Wireless Sensor Network (WSN) platforms on one side and Software Defined Radio (SDR) platforms on the other side.

There are many SDR platforms that have been developed to support individual research projects. The Japanese National Institute of Information and Communications Technology (NICT) constructed a SDR platform [7] to test next generation mobile networks. The platform has two embedded processors, four Xilinx Virtex2 FPGAs, and RF modules that could support 1.9 to 2.4 and 5.0 to 5.3 GHz. The signal processing was partitioned between the CPU and the FPGA, with the CPU taking responsibility for the higher layers. An objective of this platform is to explore selection algorithms to manage the use of different existing standards.

Berkeley Cognitive Radio Platform (CRP) [8] is based on the Berkeley Emulation Engine (BEE2) which is a platform that contains five high-powered Xilinx Virtex2 FPGAs and can connect up to eighteen daughter-boards. In this platform, daughter-boards have been designed to support up to 25 MHz of bandwidth in an 85 MHz range in the 2.4 GHZ ISM band. The RF modules have highly sensitive receivers and operate either concurrently at different frequencies (FDD) or at the same frequency in a time-division manner to avoid self-generated noise. This CRP requires only a low-bandwidth connection to a supporting PC as all signal processing is performed on the platform.

The Kansas University Agile Radio (KUAR) [9] platform was designed to be a low-cost experimental platform aimed at the 5.25 to 5.85 GHz frequency range with a tunable bandwidth of 30 MHz. The platform includes an embedded 1.4 GHz general purpose processor, a Xilinx Virtex2 FPGA, and supports gigabit Ethernet and PCI-express connections back to a host computer. This allows for almost all processing to be implemented on the platform.

The mobile communications department at EURECOM proposed an open-source hardware/software development platform and an open-forum for innovation in the area of digital radio communications. OpenAirInterface [10] implements in the software the physical and medium-access layers for wireless communications as well as providing an IPv4/IPv6/MPLS network device interface under Linux. The initiative targets 4th generation wireless systems (UMTS Longterm-evolution (LTE), 802.16e/j) and rapidly deployable MESH networks using similar RI technologies. The development can be seen as an open-source test bed for advanced algorithmic prototyping and performance evaluation.

The Universal Software Radio Peripheral (USRP) [11] is one of the most popular SDR platforms currently available and it provides the hardware platform for the GNU radio project. The second generation platform was released in September 2008 and utilizes gigabyte Ethernet to allow support for 25 MHz of bandwidth. The system includes a medium range Xilinx Spartan3 device which allows for local processing. The radio-frequency performance of the USRP was limited and is more directed towards experimentation rather than matching any communications standard.

There are many different kinds of devices for WSN platforms with similar characteristics: low power, memory and processing constraints, and ISM bands. Bean, BTnode, MANTIS Nymph, IMote, MicaZ, SenseNode, XYZ, Sentilla Mini, EyesIFX, TelosB [12], ANT [13], and Iris [14] are some of the most important WSN devices. But none of them have different RIs or radio reconfiguration capabilities.

Focusing on CWSN devices, there are no specific platforms to develop WAHSN applications and services with cognitive capabilities. There is a commercial product, WaspMote [15], based on an ATMEL microcontroller and an expansion board that makes it possible to use two different RIs.

Here, a CWSN device is proposed with three different RIs controlled by a Microchip MCU, becoming the only cognitive platform for WAHSNs with these communication capabilities. It is designed fitting WAHSN environment limitations regarding power consumption, size, cost, and resources. At the same time, its modular design, based on expansion boards, and scalability make it adaptable to a wide range of WAHSN applications.

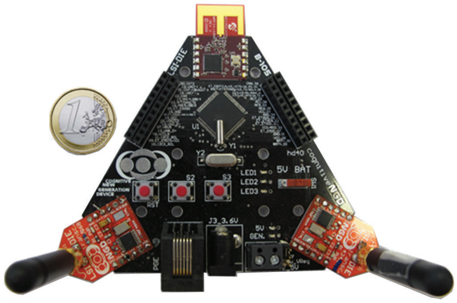


Figure 1: A picture of the cNGD.

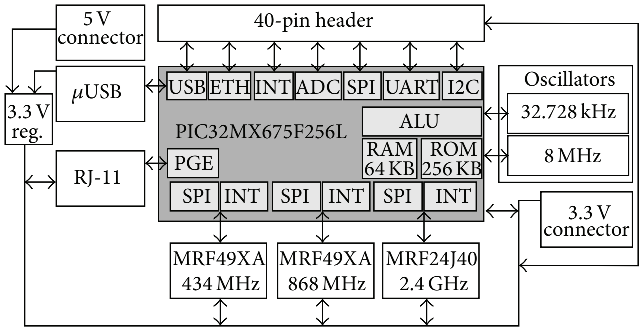


Figure 2: cNGD block diagram.

## 3. Hardware Design

The entire design of the implemented platform hardware is based on a previous experience working a CWSN node described in [16]. This node did not satisfy some of the requirements in terms of low power consumption, cost, size, and communication capabilities. Therefore, a new platform was required. The aim is a flexible scheme based on three different RF bands. As a WAHSN platform, it must have a downward trend regarding power consumption, cost, size, and resources.

This new platform, named Cognitive Next Generation Device (cNGD), is shown in Figure 1.

The objective is to work over different WAHSNs applications applying cognitive models, covering widely contrasting concepts and sorts of applications. Hence, the desired device must be modular and feature-expandable, scalable, and fully configurable.

The cNGD block diagram is explained in Figure 2.

Together with the design of the node itself, some other ad hoc devices such as proper size transceivers (Figure 3) and expansion pluggable boards were developed. These boards, called shields, aim to expand the possibilities of the main device through its generic and modular nature while meeting, at the same time, the low cost and size constraints.

The node deployed is based on a single core unit. The control core is a PIC32MX675F256L, and it is replaceable by its larger flash memory version if needed. This is a low power and low cost RTCC-including 32-bit microcontroller able to perform a throughput high enough for test-benching purposes.

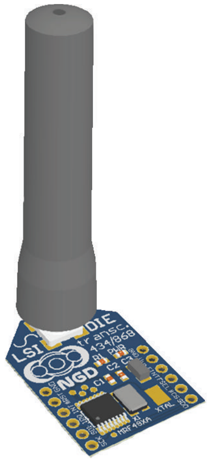


Figure 3: A 3D model of the implemented MRF49XA RF module.

cNGD executes its RF communications over MiWi protocol, an IEEE 802.15.4 proprietary standard for Wireless Personal Area Networks (WPAN). It presents great advantages for low transmission rates, short distances, and cost constrained networks, which makes it perfect for WAHSN. The physical layer of this communication stack can operate over three ISM bands: 434 MHz, 868 MHz, and 2.4 GHz. This is one of the most important features of the device and makes it the only CWSN node with these features so far.

In order to transmit in 434 MHz and 868 MHz bands, the device includes two transceivers based on the MRF49XA chip and is implemented following the design guidelines proffered by Microchip. On the other hand, an MRF24J40MA RF module is responsible for the 2.4 GHz band. All of the embedded interfaces allow enabling sleep modes and they offer quite low power consumption at transmission. Additionally, they let the applications carry out spectrum sensing, a key component of CR. There is also the possibility for the microcontroller to control the power supply of the different RIs.

The hardware includes access to different peripherals since it is not designed for a specific kind of application and must be useful as a development platform, providing the developer some extra functionalities if necessary. To accomplish this requirement, a set of pins were made accessible through a pair of 20-pin headers. These headers make the cNGD suitable for expansion with pluggable and stackable shields or even with other entire nodes. This option supposes a highly valuable feature of the device, reducing its cost but expanding its possibilities through modularity. It gives the developer the freedom to create attachable devices needed for their WAHSN applications. The list of approachable peripherals and pins covers battery connection (for charging purposes), GPIOs, MCLR pin, external interruptions, analogue inputs, USB, Ethernet module, I2C bus, UARTs, and SPI. It is important to point out that some of these functionalities are multiplexed and thus cannot be used simultaneously.

The device can work using either an external 5 V power supply, taking advantage of the rated voltage used by USB

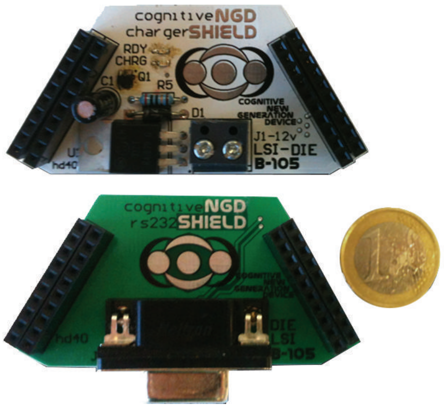


Figure 4: chargerSHIELD (top) and rs232SHIELD (bottom).

standard, or a 3.3 V battery. In case of using an external 5 V supply, the device lessens it to 3.3 V using a voltage regulator. This regulator can be deactivated when not needed, cutting down its power consumption. A switch allows commuting between batteries or wired supply mode. The battery is, in practice, essential for portability in research. A battery charger shield was designed and implemented for Ni-Mh and Ni-Cd options (Figure 4).

The hardware also provides wired serial communication using USB 2.0, which provides interoperability and facilitates data transfer from the running application to any device, normally a computer. Besides, the developer might design a custom communication method by using the different buses at the headers. For this work, a simple serial communication shield, rs232SHIELD, was designed to bridge one UART peripheral and a classic rs232 interface (Figure 4).

Apart from the reset push-button, the device contains two other general purpose buttons. It provides as well three configurable LEDs that the application might use for its own purposes. An RJ-11 connector embedded in the main board is used to program the controller, by taking advantage of the development framework provided by Microchip.

Therefore, the hardware capabilities include wireless communication over three different RF bands, the first time on a CWSN platform, controlled by a single PIC. In addition, it hosts accessible external interruptions, SPIs, Ethernet modules, and other peripherals through headers to fit the developer requirements with modular attachable devices. It also hosts wired communication through USB and portability options including an appended battery.

## 4. Firmware Design

In this section, features of the node's firmware, the RF protocol stack, and the designed Hardware Abstraction Layer (HAL) are explained.

As was mentioned, cNGD is based on a 32-bit microcontroller and three RF transceivers, each over an ISM band. As these elements are all part of Microchip's catalog, incompatibility issues were avoided and communication among them was quite straightforward.

*4.1. Adapted MiWi Protocol Stack.* The chosen transceivers can work either with Microchip's proprietary MiWi protocol stack (free of charge) or with ZigBee stack (under license). It was decided to use MiWi since this one supposes an easier and lighter solution. Its features cover firmware-specific CWSN requirements: reconfiguration of radio parameters, spectrum sensing capabilities, and enough data rate for potential applications.

The MiWi protocol stack is structured in layers according to the OSI model and the IEEE 802.15.4 standard. Its implementation ranges from physical to network layer, with three network protocols available. The protocol to be used can be chosen in regard to network topology, size, and routing requirements. Microchip also offers a top layer interface as an API for developing user applications.

Originally, MiWi stack was designed to work only with one of its compatible transceivers at a time. In order to admit the three desired devices, an exhaustive MiWi protocol stack adaptation was carried out. Thanks to the adaptation, the node firmware can handle all the transceivers' routines and protocols' tasks together as a whole.

Otherwise, the replication of the stack threefold would have been very inefficient in terms of processing and memory usage. The average program memory saved with this firmware is estimated at 31%, with variations regarding configuration parameters. This improvement would be even more significant if the HAL implementation was not taken into account.

During the implementation process, an important effort was aimed at achieving a more modular and flexible design. The firmware must be coherent with the hardware design but also must be highly reconfigurable and admit upgrades. The design decisions pursue optimal resource utilization and independence from the hardware design. For instance, if a simplified node implementation with not all the RF modules and different peripherals selection were desired, the firmware will still work after minimal configuration changes.

*4.2. Hardware Abstraction Layer.* In order to isolate the developer from the complexity of the adapted protocol stack, a HAL has been implemented. It offers the upper layer a set of functions as an API, enabling the development of application or modules without the need of understanding lots of the stack details. Despite the fact that the HAL implementation implies more RAM and flash memory usage, the benefits make up for this drawback.

All HAL functions are designed for returning an error code. Thus, the upper layer can be aware of the result of the last action carried out and, if necessary, it can undertake countermeasures regarding the identified error cause. This leads to both a reduction of applications development time and a more reliable network operation. The HAL implementation is scalable, as new error codes, functionalities, and modules can be incorporated in to it.

The HAL deals with network initialization and maintenance, power management, and debugging, among others. However, as the design is intended to be a node for CWSN, the emphasis was placed on radio communication and reconfiguration. It was an objective to keep the HAL functions simple and yet covering the most functionalities as possible. As an example, the node can do an energy scan over all the frequency bands with a single call to the proper function.

The available configuration choices have been widened due to the additional versatility provided by the HAL. One of the most remarkable features of the node's firmware is the possibility of tailoring the configuration parameters to fit the target application requirements.

All in all, the adapted protocol stack and the implemented HAL provide the cognitive module and the applications with an easy and intuitive interface for managing the node's features.

Table 1: Relationships between modules.

| Origin | Destination | | | | | | |
|---|---|---|---|---|---|---|---|
| | Disc. | Exec. | Opti. | Repo. | Poli. | Accss. | VCC |
| Disc. | | | | M | | | |
| Exec. | M | | M | | | M | |
| Opti. | M | M | | M | M | | M |
| Repo. | | | | | | | |
| Poli. | | | M | M | | | |
| Accss. | | | | M | | | |

M: Messenger.

## 5. Software Architecture

*5.1. Overview.* As was previously exposed at the Introduction, the model is based on the Connectivity Brokerage scheme. This architecture is composed of six different modules.

Repository stores information regarding the node, the network, nodes from inside or outside the network, and the environment.

Discovery module collects information considered interesting such as Received Signal Strength Indicator (RSSI), Link Quality Indicator (LQI), and different information regarding the discovered nodes.

Optimization module executes the desired cognitive strategy.

Execution module takes actions normally ordered by the Optimization module.

Access Control module is responsible to control the actions other nodes are able to take in such node.

Policy module keeps some variables representing different policies such as energy consumption, security, and others. Different weights of these variables will force Optimization module to take different actions.

Apart from these modules, two more have been added to the structure: Messenger and Virtual Control Channel (VCC).

Conceptually, VCC module already existed in Connectivity Brokerage, but not as part of the CRModule. In this implementation, part of VCC's functionality has been included within the CRModule in a module with the same name. This will be justified in the following points of this paper.

There are basically two types of messages depending on whether their destination and source modules belong to the same node or not.

*5.2. Intramessages.* These messages are the ones that involve two modules of the same node.

The Messenger module acts as a bridge for communicating the rest of the modules with each other and with the VCC module itself. However not all the modules are able to communicate with the rest, for example, the Repository module cannot request anything as its purpose is just to store data. That is the reason why the possible interactions between modules are presented. The relations describe the requests in a unidirectional way.

In Table 1, the relation between modules can be seen marked with Ms to represent that the requests are processed through the Messenger module.

When implementing cognitive strategies in a node, one of the problems to be faced is the flow of a relatively high amount of messages between modules. A Messenger module is helpful since it relieves and makes easier the communication tasks.

Suppose a strategy that is being executed in routines allocated in the Optimizer module. These routines will potentially need to interact with Repository, Executive, Discovery, and probably other nodes in the network, which implies interaction with the VCC module. The shortest way to do this is to directly call the specific functions of each module.

However, as the functionality of the modules increases, it becomes more and more complex to implement these calls. Furthermore, if a function is modified, or a new module is added, it would be a very complex task to reflect those changes in all the affected modules. So, the Messenger module provides an interface to communicate each module with any other in a generic way.

Therefore, the Messenger module reduces complexity when implementing strategies and increases the scalability of the software architecture.

*5.3. Intermessages.* These messages occur when a module from one node sends a message to a module in a different node. These types of messages are called control messages.

A simple VCC module has been included in the CRModule due to the need of sending messages among modules in different nodes. When implementing a cognitive-based strategy in CWSNs, as a cooperative network, it is common that multiple nodes will be implied in the strategy. That way the Optimizer module of a node may need information from other nodes' Repository module, execute changes in them through their Executive module, or just request that their Optimizer module carries out the same cognitive strategy of the first one. Therefore, the VCC module constitutes a tool for creating and processing messages received from and sent to other nodes in the network.

Once at the destination node, the VCC module requests will arrive to the destination module through Messenger and also upon Access Control approval. When Messenger is prompted to process a message that comes from the VCC

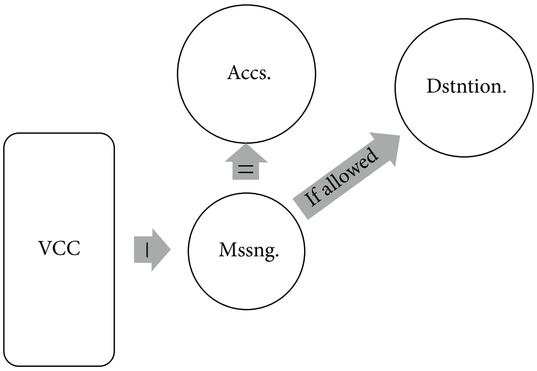


Figure 5: Receiving control message protocol.

module, it automatically knows that it is a control message coming from a different node. Therefore, prior to sending the request to the destination module, it will check the permissions for the origin node, destination module, and specific function that are requested in the Access Control module. Figure 5 illustrates this sequence.

*5.4. Communications.* As has been explained, there will be two different types of messages depending on their destination.

Consequently, the Messenger module is composed of two message processors, one to send the requests to other modules in the node and the other to send/receive requests to/from other modules of external nodes.

When making requests to another module in the same node, apart from the two modules involved in the communication, only the Messenger module intervenes, as specified previously in this section. Its aim is to transport the request from origin to destination. To do that, a primary function is defined in the Messenger module. This function is generic so it can be used to send requests between any modules (in accordance with Table 1). The function also has some generic parameters that carry any kind of module-specific variables to the destination.

Within a request targeting a module in a different node in the network, the communication can be seen as the one explained above for the transmitting node, as it is a request sent by the origin module to the VCC module. However, internally in the Messenger module, it is processed by a different message processor which also adds the proper header to the message prior to sending it to the VCC module, so it will be ready for sending it wirelessly.

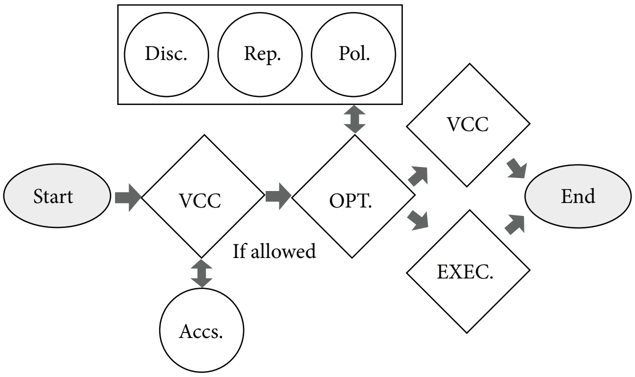


Figure 6: Generic execution logic diagram.

Table 2: Information about cNGD current consumption.

| Design element and operation mode | Supply current ($I_{DC}$) |
|---|---|
| 434/868 MHz RI (RX mode) | 11.8 mA |
| 2,4 GHz RI (RX mode) | 16.5 mA |
| Transceivers 434/868 MHz (sleep mode) | 2.2 mA |
| Transceivers 2.4 GHz (sleep mode) | 1.1 mA |
| MCU (run mode at 80 MHz) | 50.9 mA |
| MCU (sleep mode) | 0.83 mA |

*5.5. Generic Execution Example.* A general execution of the architecture over a node is illustrated in Figure 6, Messenger module is not represented but its functionality is assumed. A routine execution might be triggered by several events; here we consider the reception of a control message through the VCC. In this case, the message requests a task to be carried out at the Optimization module. After obtaining the access approval by the Access Control, Optimization will achieve the task. This module might be in need of data within Discovery, Repository or Policy modules, which will be accessed. Optimization module also might give way to a posterior action at the Executive module or an intermessage delivery at the VCC. These last tasks would end the routine execution.

The CRModule provides a platform for carrying out cognitive and cooperative strategies not only in one node but in a network, and makes their implementation less complex and more efficient in terms of memory.

## 6. Tests and Results

The platform has been tested in order to prove and characterize valuable features such as RI agility. Carried out tests measure the MCU and RIs performance and global current consumption. Results are shown throughout this section.

In Table 2, information about the power consumption of the platform is shown.

Flash memory needed by the firmware is 88.2 KB, whereas the required RAM memory is 2.1 KB at its widest configuration. On the other hand, flash memory required by the cognitive architecture is 32.6 KB. Table 3 shows the time costs of the communication protocol stack management. The results gather separately the measured times for different RIs and protocol versions. A network coordinator role and no network activity are considered.

Another test intends to measure the RIs control cost. Results in Table 4 provide the time cost for different operations at the three RIs. Results are similar for 434 MHz and 868 MHz RIs since they both use the same transceiver model. Spectrum scan task takes longer than others due to needed software delays introduced at the respective routines.

Table 3: Time costs on the communication protocol management tasks.

| Management part | P2P ($\mu$s) | Mesh ($\mu$s) |
|---|---|---|
| Common management | 134.1 | 150.6 |
| 434 MHz RI | 1.3 | 1.3 |
| 868 MHz RI | 1.2 | 1.2 |
| 2.4 GHz RI | 1.2 | 1.2 |
| Total | 137.8 | 154.3 |

Table 4: Time costs on the radio interfaces control tasks.

| Operation | 434 MHz RI | 868 MHz RI | 2.4 GHz RI |
|---|---|---|---|
| Transmission channel switch | 30.45 ms | 30.4 ms | 0.151 ms |
| Transmission power change | 55.8 $\mu$s | 55.8 $\mu$s | 70.1 $\mu$s |
| Sleep and wake up | 15.4 ms | 15.4 ms | 0.364 ms |
| Spectrum energy scan | 194.48 ms | 908 ms | 410.13 ms |

Table 5: Effective rate at the application level for different TX modes.

| TX mode | 434/868 MHz RI (38.4 kbps) | 434/868 MHz RI (115.2 kbps) | 2.4 GHz RI (250 kbps) |
|---|---|---|---|
| Broadcast | 28.94 kbps | 31.25 kbps | 28.8 kbps |
| Unicast | 20.67 kbps | 24.51 kbps | 28.47 kbps |

Effective rate provided at the application level was measured for broadcast and unicast TX modes. 434 and 868 MHz RIs have been tested for two different bit-rate configurations. P2P protocol is employed and a 90 Bytes packet payload. Table 5 gives the results. It can be observed how the theoretical RI maximum bit-rates shrink at the application level to around 30 Kbps. This decay occurs because of the protocol overhead, encryption, and a data processing bottleneck placed at the reception.

## 7. Conclusions

CWSNs appear as a new concept that proposes collaborative and learning models focused on improving spectrum efficiency and communications performance. Because of the novel state of this research field, there are no many specific devices to keep investigating this promising concept.

The presented work supposes a versatile platform to work over CWSNs development and deployment. Combining hardware and software modules, it offers a flexible modular design widely adaptable over the range of WAHSNs applications. The sort of device shown here features some novel properties with respect to other current CWSN devices, such as its capability to communicate over three different ISM RF bands.

The hardware fits the conditions and requirements of WAHSN environments. Low power consumption, size, and cost limitations are taken into account in order to achieve real test-benching purposes, application development, or even possible complete implementations.

To keep the downtrend on complexity, consumption, size, and cost, an expansion system for the platform was implemented, allowing developers to create ad hoc attachable functionalities or improvements for the nodes.

cNGD's firmware was developed with a focus on abstracting the developer and the application from the cognitive model and the direct hardware management. Furthermore, a single MiWi stack was adapted to deal with the three RF transceivers instead of using three different stacks. The average program memory saved with this firmware is of 31%. The firmware also exhibits scalability and modularity, being adjustable to diverse applications. To carry out the cognitive algorithms, the software architecture implements the Connectivity Brokerage scheme, including some extra modules to achieve a greater flexibility and reduce complexity of the cognitive model structure.

Tests carried out over the platform prove some of its valuable capabilities and features. Current consumption is under 1 mA during sleep modes. Protocol stack and RIs management tasks have been measured. Longest time for a full stack management task is under 155 $\mu$s and a spectrum energy scan all over the three frequencies takes place in less than 1.5 s. Effective rate tests reveal similar performance amongst RIs and maximum communication bit-rates provided at the application are around 30 kbps.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgment

This work was partially funded by the Spanish Ministry of Economy and Competitiveness, under INNPACTO program (Reference Grant IPT-2011-1197-370000, PRECOIL; IPT-2011-1241-390000, LINEO; IPT-2011-0782-900000, S2S).

## References

[1] CISCO System Inc., "Cisco visual networking index: global mobile data traffic forecast update, 2010–2015," Tech. Rep., 2011.

[2] I. Howitt and J. A. Gutierrez, "IEEE 802. 15. 4 low rate—wireless personal area network coexistence issues," *Wireless Communications and Networking*, vol. 3, pp. 16–20, 2003.

[3] J. Mitola III, *Cognitive radio: an integrated agent architecture for software defined radio [Ph.D. thesis]*, Royal Institute of Technology (KTH), Stockholm, Sweden, May 2000.

[4] O. B. Akan, O. B. Karli, and O. Ergul, "Cognitive radio sensor networks," *IEEE Network*, vol. 23, no. 4, pp. 34–40, 2009.

[5] D. Cavalcanti, S. Das, J. Wang, and K. Challapali, "Cognitive radio based wireless sensor networks," in *Proceedings of the 17th International Conference on Computer Communications and Networks (ICCCN '08)*, pp. 491–496, August 2008.

[6] S. Mustafa, A. Parsa, S. Pollin, I. Hsiang Wang, and P. Malagon, "Connectivity brokerage—enabling seamless cooperation in wireless networks," Tech. Rep., Berkeley Wireless Research Center, 2010.

[7] H. Harada, "Software defined radio prototype toward cognitive radio communication systems," in *Proceedings of the 1st IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN '05)*, pp. 539–547, November 2005.

[8] S. M. Mishra, D. Cabric, C. Chang et al., "A real time Cognitive Radio testbed for physical and link layer experiments," in *Proceedings of the 1st IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN '05)*, vol. 1, pp. 562–567, Baltimore, Md, USA, November 2005.

[9] G. J. Minden, J. B. Evans, L. Searl et al., "KUAR: a flexible software-defined radio development platform," in *Proceedings of the 2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, pp. 428–439, April 2007.

[10] OpenAirInteface, http://www.openairinterface.org/.

[11] G. Berardinelli, P. Zetterberg, O. Tonelli, A. F. Cattoni, T. B. Sorensen, and P. Mogensen, "An SDR architecture for OFDM transmission over USRP2 boards," in *Proceedings of the 54 Asilomar Conference on Signals, Systems and Computers (ASILOMAR '11)*, pp. 965–969, November 2011.

[12] TelosB sensor node, http://www.xbow.com/Products/productdetails.aspx?sid=252.

[13] ANT platform, http://www.thisisant.com/.

[14] Iris sensor node, http://www.xbow.com/Products/productdetails.aspx?sid=264.

[15] WaspMote, http://www.dinesgroup.org/projects/images/pdf_files/iris_datasheet.pdf.

[16] F. López, E. Romero, J. Blesa, D. Villanueva, and A. Araujo, "Cognitive wireless sensor network device for AAL scenarios," in *Ambient Assisted Living*, vol. 6693 of *Lecture Notes in Computer Science*, pp. 116–121, 2011.

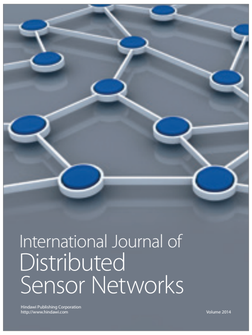
The Scientific World Journal

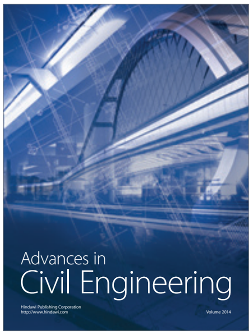
International Journal of
Rotating Machinery

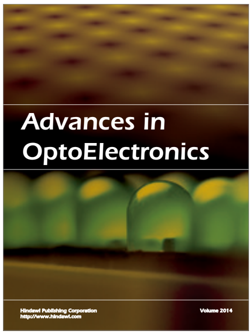
Journal of
Engineering

Journal of
Sensors

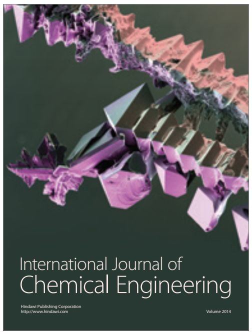
Advances in
Mechanical Engineering


International Journal of
Distributed Sensor Networks


Advances in
Civil Engineering

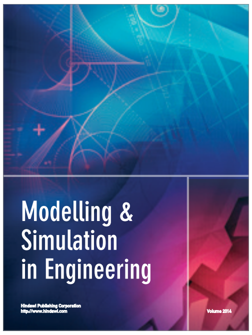

Advances in
OptoElectronics

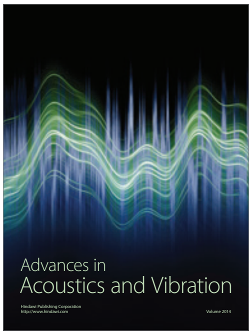
Journal of
Robotics


International Journal of
Chemical Engineering

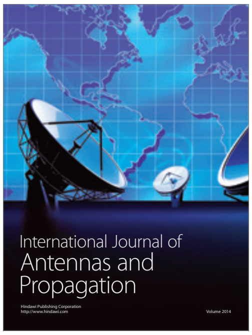
VLSI Design

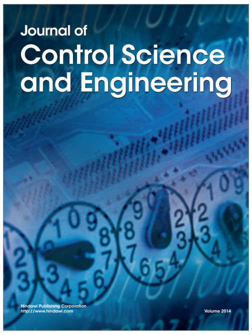
International Journal of
Navigation and Observation

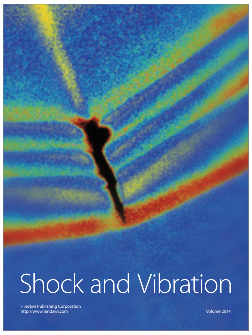

Modelling &
Simulation
in Engineering


Advances in
Acoustics and Vibration

Active and Passive
Electronic Components


International Journal of
Antennas and Propagation


Journal of
Control Science
and Engineering


Shock and Vibration

Journal of
Electrical and Computer
Engineering