A Modular Design for Wireless Structural Health Monitoring Applications

Fabio ANGELETTI, Mario PAOLI, Ugo Maria COLESANTI, and Andrea VITALETTI
Department of Computer, Control and Management Engineering
University of Rome La Sapienza
E-mail: angeletti@dis.uniroma1.it, paoli@dis.uniroma1.it, colesanti@dis.uniroma1.it, vitaletti@dis.uniroma1.it

Received: 31 August 2015 /Accepted: 15 October 2015 /Published: 30 November 2015

Abstract: This article presents the research and development activity on the Modular Monitoring System (MMS), a low-power wireless modular architecture for Structural Health Monitoring (SHM) introduced in [1]. The MMS is a device capable to easily interface the vast majority of sensors employed in SHM thanks to a novel modular architecture. In addition, the low-power design allows the MMS to support long-lasting monitoring activities and to support the deployment of modern Wireless Sensor Network (WSN) techniques.

Keywords: Wireless sensor networks, Structural health monitoring, Modular architecture, Wireless, Low-power.

1. Introduction

Structural Health Monitoring (SHM) deals with the detection of damages to which civil and industrial structures, such as roads, bridges, canals, buildings and aero-space vehicles are subjected to. It can prevent collapses and breaks, avoiding permanent damages to structures, thus simplifying and improving the effectiveness of their maintenance. Depending on application scenarios, SHM requires many different types of sensors, including pressure sensors, vibrating-wire strain gauges, inclinometers, crackmeters, etc. Nowadays, most SHM systems in the market are wired. However, the deployment of a wired system in a wide area or in a harsh environment, can pose both economical and practical limitations. For this reasons, WSNs have been proposed as an ad-hoc network infrastructure to support SHM, avoiding prohibitive costs of wired systems and easing the on-field deployment. Nowadays, SHM supported by WSN is an active and well-established research field and some wireless SHM systems are now entering the market. The work presented in this paper introduces the MMS, a wireless, low-power, scalable hardware architecture dedicated to SHM. A key feature of the MMS is its high modularity that allows easy customization of the platform depending on the number and the type of sensors required by the specific application scenario. In addition, MMS fully supports multi-hop wireless communication paradigm and mesh networks. The remainder of this paper is organized as follows: Section 2 presents the state-of-the-art on both wired and wireless SHM systems. In Section 3 we explain the motivations supporting the design choices made during the development of the MMS, while in Section 4 we introduce the features characterizing our system. In Section 5, we present the hardware
2. State of the Art

Nowadays, most SHM systems available on the market, such as the Geomonitor by Solexperts [2], are wired. However, deploying those systems can be cumbersome: besides the installation costs, a detailed deployment plan is required to face evolving needs of different construction phases. Moreover, cables are subjected to accidental cuts and damages and, in some scenarios, their installation is infeasible or inappropriate (e.g., historical buildings). Along with wired systems, some standalone data loggers dedicated to SHM are commercially available, among the others: Solexperts SDL [2], Geokon 8002-16-1 [3] and Keller GSM-2 [4]. These systems are simpler to install, but they do not allow real-time remote monitoring and require frequent in-situ access by qualified personnel to collect data.

The introduction of wireless communications in SHM gives immediate advantages in terms of easier deployments and reduced maintenance and personnel costs. However, when monitoring devices are battery-powered, the use of wireless communications is among the most energy demanding feature that can significantly limit the devices lifetime. Prominent examples of wireless monitoring systems are: National Instruments Wireless Data Acquisition (WiFi-DAQ) [5] and MicroStrain's wireless sensor network [6]. While the former supports IEEE 802.11 standard, the latter is compliant with IEEE 802.15.4. Both products adopt a conservative approach by limiting wireless communication to 1-hop.

Supporting multi-hop wireless communications was investigated in several research papers in the last decade [7-9]. Multi-hop networking provides a number of advantages: scalability (larger areas can be covered), reliability (failure and multiple routing paths without single point of failure) and ease of deployments (the presence of multi-hop relay nodes allows to bypass obstacles like walls and metal structures). Recently, National Instruments presented the NI WSN [10]: a multi-hop battery-powered WSN supporting up to 36 nodes configured in a mesh network and up to 3 years node lifetime.

3. Motivations

As seen in the previous section, WSNs are slowly entering the market of SHM applications. An attempt to develop a robust solution and to test it in realistic application scenarios was made in the GENESI Project [11]. The main goal of that project was to design and implement a “novel generation of green wireless sensor networks which can be embedded in buildings and infrastructures at the time of construction and be able to provide a monitoring and control intelligence over the whole structure lifetime”. The project involved the monitoring of a bridge construction site in Fribourg [12] and the construction of a tunnel for the B1 underground line in Rome [13], by means of WSNs. The outcomes of these experimental activities, highlighted the advantages of WSN technology compared to old monitoring techniques. According to the application requirements provided by the SHM experts, GENESI nodes can support a number of heterogeneous sensors. However, a GENESI node can manage only a single sensor per type, while there are some applications in which multiple instances of the same sensor are needed. As an example, in a 3-axis deformation analysis, a single wireless node may need to interface with three instances of a vibrating-wire strain gauge while to monitor a concrete junction the node may need to interface a current-loop inclinometer sensor and a resistive displacement sensor. Other commercial solutions, such as the NI WSN described in Section 2, can handle multiple instances of the same sensor but can not support different sensor families at the same time. In general, the development of new ad-hoc devices addressing the specific requirements of an SHM deployment is impractical, while the adoption of commercial solutions in such contexts is not optimal in terms of flexibility, size and costs. This brought us to propose a novel low-power slotted modular system made by a set of modules connected through an internal communication bus. This solution features one wireless master module managing a group of extension modules, each one designed to interface a specific sensor set. The flexibility of the proposed architecture, named MMS, allows us to support a vast number of SHM applications by simply plugging into each node the required extension modules. By changing the master module, it is also possible to easily modify the wireless technology, thus effectively adapting to the heterogeneous needs of indoor and outdoor communication requirements.

4. System Architecture

4.1. Preliminary Considerations

The original MMS architecture presented in [1] was based on a master/slave communication abstraction where “a master module, provided with radio capability and responsible for most of computational tasks, communicates through a low-power shared bus with a maximum of four extension modules (slaves)”. The new architecture embraces the same principle but requires the design of a new low-power shared bus.

The low-power shared bus in [1] was based on the Serial Peripheral Interface (SPI) Protocol with dedicated Chip Select (CS) and interrupt lines for each module. The rationale for this choice was to
minimize the power consumption of the whole platform by allowing the master module to selectively activate each extension module acting on the corresponding chip select line, thus, without affecting the power consumption of the other installed on the platform. Despite the effectiveness of such solution from the power consumption perspective, the elevated number of dedicated lines (2 shared lines for the power supply, 3 shared lines for the SPI and 2 dedicated lines, CS and interrupt, for each extension module) forced us to develop a backplane board with a pre-defined number of slots at design time. In the original release, shown in Fig. 1, we chose to support one master module and 4 extension modules, as a good trade-off between size and modularity.

This choice quickly revealed its limits: the additional backplane board increased the overall size of the platform and its costs for the electronics, and it also required extra costs for the ad-hoc housing. To overcome these limits we took hints from existing modular housing solutions available on the market and we updated our design as described below. We found out that most of the considered modular products are based on O type DIN rail standard [14], a widely used solution for mounting circuit breakers and industrial control equipment inside racks. To ease the communication between modules, such solutions commonly provide a 5 lines shared bus. Hence, we implemented our system in a modular DIN housing and we redesigned the low-power shared bus to be compliant with the reduced number of lines offered by DIN bus. This led us to the new MMS architecture described in the following sections.

4.2. Low-Power Shared Bus

The availability of only 5 shared lines in the bus drastically reduces the implementation options for a master/multi-slave communication system. Excluding power and ground lines, the low-power shared bus has to be implemented on the remaining 3 shared lines. To overcome this constraint, we used the Intergrated Circuit protocol (I²C) which is a multi-master/multi-slave serial communication protocol based on two open-drain lines bus [15]. The communication protocol, which is master initiated, is based on the transmission of a 7 bit address, i.e., the slave address to which the master wants to communicate, followed by a read/write bit and a set of protocol dependent commands. The third open-drain shared line has been used by the extension modules to trigger the master to start a communication. The master module polls to slaves to identify which extension module triggered the interrupt on the third line. Fig. 2 shows the new architecture with a detailed view of the new low-power shared bus implementation.

This new solution easily scales with the number of extension modules (the standard implementation of the I²C bus is designed to support up to 127 devices), thus increasing the flexibility and optimizing the size of the MMS. However, from the power consumption perspective, we need to pay attention to several drawbacks:

- **Wake-up.** When the master module issues an I²C start command, all the extension modules wake up to perform the address matching, even if the communication is addressed to one module only.
- **Polling.** When an extension module starts a communication, it notifies the master using the interrupt line. Subsequently the master polls all the devices to find the initiator.
- **Latency.** The I²C protocol needs more data to be sent over the bus with respect to the SPI one, this increases the latency and the power consumption of the MMS.

In Section 6.3 we analyzed each of these aspects measuring the power consumption overhead related to this new implementation with respect to the original low-power shared bus. The results show that the overall consumption of the MMS is marginally affected by this choice.

4.3. Master Module

As in the first version of the MMS, the master module is responsible for managing the wireless connectivity within the WSN and manages the extension modules. When switched on, the master performs a discovery routine for dynamically assign the I²C addresses to each extension modules and to
retrieve the configuration information from them. Then, the master computes the sensing schedule and switches to the operational mode. Below a typical master-initiated interaction with the extension module is described:

1. The master issues an I²C start command over the bus and sends the address of the extension module followed by a write bit and a data request command.
2. The extension module wakes-up and starts the conversion while the master enters a low-power state, waiting for an interrupt.
3. The master wakes up when an extension module pulls down the interrupt line.
4. The master polls all the modules searching for the initiator.
5. When found, the master sends a Data Read command to retrieve the new data.
6. When the read finishes the extension module puts the interrupt line in high impedance and enters sleep state again.
7. The master module checks the interrupt line level for other extension module that are willing to communicate. If the line is still low, it jumps to step 4 otherwise it goes back to sleep.

4.4. Extension Module

The extension modules provide the hardware interface to external sensors. Each module communicates over the low-power shared bus with the master by means of an I²C-capable microcontroller. When powered-on, the extension modules participate to the I²C address assignment managed by the master module. Once an extension module gets an address, it switches to a sleep state and wakes up only upon the detection of an I²C start command on the bus. The extension module can initiate a communication by pulling down the shared interrupt line of the low-power shared bus, which activates the polling procedure by the master.

Each extension module provides a register configuration area that enables the interaction with the master module. The register area is divided in 5 subsections as follows:

- The **Information area [read-only]** stores information such as device type and revision. This area allows the master to identify the extension module, e.g., sensors supported, channels available, etc.
- The **Command area [write-only]** is used by the master to trigger commands to the extension module such as a read command.
- The **Settings area [read/write]** stores settings for each sensor channel such as the sensor type connected, the periodic sampling value, etc.
- The **Channel flags area [read-only]** is used by the extension module to notify that a new data is available.
- The **Channels data area [read-only]** is used by the extension module to store the last sensor value read.

An example of extension module register implemented on our test board is shown in Table 1.

The hardware interface provided by each module depends on the sensors it supports: it can be fully digital, e.g., to interface RS-232 or RS-485 peripherals, or analog, to connect sensors such as current-loop, vibrating wire strain gauges, resistive etc. Each extension module can support different kinds of sensors or several sensors of the same type.

**Table 1. Extension module register area example.**

<table>
<thead>
<tr>
<th>Address</th>
<th>Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>Device type</td>
</tr>
<tr>
<td>0x02</td>
<td>Device description</td>
</tr>
<tr>
<td>0x03</td>
<td>Cmd channel 0</td>
</tr>
<tr>
<td>0x04</td>
<td>Cmd channel 1</td>
</tr>
<tr>
<td>0x05</td>
<td>Cmd channel 2</td>
</tr>
<tr>
<td>0x06</td>
<td>Cmd channel 3</td>
</tr>
<tr>
<td>0x07</td>
<td>Configuration channel 0</td>
</tr>
<tr>
<td>0x0e</td>
<td>Configuration channel 1</td>
</tr>
<tr>
<td>0x17</td>
<td>Configuration channel 2</td>
</tr>
<tr>
<td>0x1f</td>
<td>Configuration channel 3</td>
</tr>
<tr>
<td>0x27</td>
<td>Data flag channel 0</td>
</tr>
<tr>
<td>0x28</td>
<td>Data flag channel 1</td>
</tr>
<tr>
<td>0x29</td>
<td>Data flag channel 2</td>
</tr>
<tr>
<td>0x2a</td>
<td>Data flag channel 3</td>
</tr>
<tr>
<td>0x2b</td>
<td>Data channel 0</td>
</tr>
<tr>
<td>0x3b</td>
<td>Data channel 1</td>
</tr>
<tr>
<td>0x4b</td>
<td>Data channel 2</td>
</tr>
<tr>
<td>0x5b</td>
<td>Data channel 3</td>
</tr>
</tbody>
</table>

4.5. Power Module

The modular design of the new MMS allows us to support a number of interchangeable power modules which provide the power supply to the whole MMS through the power line on the bus. The modules support any kind of battery type providing a voltage between 3 and 5.5 V as well as the 240 V AC and energy harvesting solutions. Notice that this flexibility was not possible in the previous release of the MMS, since the power unit had to be integrated into the backplane at design time.

5. Hardware and Firmware Development

We developed and assembled the new version of the MMS at the Department of Computer, Control and Management Engineering of the University of Rome “La Sapienza”. The current version includes one master module, two demo extension modules and two power modules. As opposed to the first prototype, this version fits in a complete modular housing solution based on pluggable slots and snap-in connections for external sensors (Fig. 3). The whole system was housed in a steel IP66 enclosure with a mounted DIN rail, an external antenna and IP66 cable glands.
5.1. Master Module

The master module is responsible for radio communication and for managing the extension modules. It is based on the MagoNode OEM [16], a wireless hardware platform specifically designed for WSN applications. The MagoNode is based on the Atmel Atmega128RFA1 (RFA1) System-On-Chip microcontroller (MCU) equipped with 128 kB of ROM (Read Only Memory), 16 kB of RAM (Random Access Memory) and an embedded 2.4 GHz radio transceiver fully compliant with the 802.15.4 standard. The radio range is extended through a power efficient RF (Radio Frequency) front-end which enhances radio performance while keeping the power consumption low. The main figures in terms of power consumption are: radio transmission 27.7 mA @+10 dBm, radio reception 14.5 mA and <2 µA in sleep. The master module is equipped with a NOR flash for persistent data storage, three status LEDs, a mini-usb plug and an RP-SMA (Reverse Polarity SubMiniature version A) antenna connector. An optional wireless HART [17] communication module can be installed based on application requirements.

The firmware is written using TinyOS [18], an event-driven, open-source operating system (OS) dedicated to Wireless Sensor Networks. TinyOS is designed to cope with typical constraints imposed by Wireless Sensor Networks: low computational capabilities, limited memory and scarce energy resources. TinyOS also provides a set of libraries implementing low-power wireless protocols for medium access control and routing. The implementation of the MMS firmware on TinyOS allows an easy and effective integration of our system in large-scale WSNs deployments.

5.2. Demo Extension Module

We developed a general-purpose extension module for demo applications which provides 4 24-bit analog channels, 2 excitation current outputs and 2 external voltage references. The device logic is handled by an efficient ARM Cortex M0+ MCU. The MCU provides the I²C hardware interface and the interrupt channel required by the low-power shared bus communication, an SPI interface for the 24-bit ADC and a set of General Purpose Input/Output (GPIO) pins for driving the 3 status leds of the module. As opposed to the first MMS prototype which could switch off the power of the extension modules, in this case the low-power shared bus has an always-on power line. This forces all the extension modules to remain (by default) in a sleep mode when inactive: they are enabled only when an address match event on the I²C peripheral occurs. The demo extension module supports a sleep mode with a current consumption of 1.5 µA1. As the master module, the firmware was implemented using the TinyOS operating system which provides the primitives for I²C and SPI communication and for handling the sleep mode of the device.

5.3. Power Module

We currently support two basic power modules: an AA and D type 3.6 V thionyl chloride battery. A module to power the MMS via the 240 V AC is in an advanced development stage and we are also working on the integration of an energy harvesting module supporting solar cells.

6. Experiments

In our experiments of the new MMS design, we first measured the power consumption during a local data acquisition (i.e., with wireless radio disabled), then we evaluated the system in a Wireless Sensor Network testbed. The former measurement validates the effectiveness of the low-power design in our architecture, while the latter demonstrates how the MMS features energy consumption levels suited for WSN applications. Finally, we performed a set of measurements aimed to demonstrate how the energy consumption overhead introduced by the new design of the MMS (see Section 4.2) is negligible if compared to the overall system consumption.

6.1. Local Data Measurements

All the measurements were done powering the MMS system with a Rigol DP1308A programmable DC power supply, providing 3.0 V. The MMS was

---

1 An errata specific to the MCU revision used in the extension module avoided to reach this value which refers to the newest chip revisions.
connected in series to a Rigol DM3068 digital multimeter that sampled the current consumption at 10 kHz. The measurements were taken from an MMS made of a master module and a single demo extension module connected to a displacement sensor, i.e., a potentiometer with 4 kΩ series resistance. We performed a single master-initiated sampling request which consists in the following steps:

1. The master module sends the address of the extension module and sets a bit in the command area register to trigger one sample from a single channel on the selected extension module.

2. The extension module starts the conversion while the master goes back to sleep waiting for an interrupt.

3. The extension module generates an interrupt to the master notifying that it has a new data sample available.

4. The master polls all the modules, reading the data flag area. Whenever a flag notifies new data available, the master retrieves the new data.

In a second experiment, we repeated the same procedure using an additional demo extension module connected to an identical transducer where the master performs a sample request from each module. The current consumption of both tests are shown in Fig. 4. The data conversion time, which is equal to 50ms, is determined by the sampling rate of the ADC which was configured to 20 samples per second. As expected, the current consumption during data conversion is doubled in the two extension modules configuration since each module performs the conversion at almost the same time. In addition, it is clearly visible how the sample request and data retrieval phases last twice in the two extension modules configuration with respect to the single one.

Table 2. Local measurement - consumption details.

<table>
<thead>
<tr>
<th>State</th>
<th>Single ext. module</th>
<th>Double ext. module</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Avg. current (mA)</td>
<td>Energy (µJ)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Avg. current (mA)</td>
</tr>
<tr>
<td>Sleep</td>
<td>0.0035</td>
<td>N/A</td>
</tr>
<tr>
<td>Sample request</td>
<td>3.65</td>
<td>28</td>
</tr>
<tr>
<td>Data conversion</td>
<td>1.69</td>
<td>281</td>
</tr>
<tr>
<td>Data retrieval</td>
<td>2.79</td>
<td>104</td>
</tr>
<tr>
<td>Total</td>
<td>1.95</td>
<td>413</td>
</tr>
</tbody>
</table>

6.2. WSN Testbed

We tested the two extension module configuration, described in the previous section, in an indoor wireless sensor network deployment located at the basement of our department. The testbed setup (Fig. 5), consisted in 10 IRIS wireless devices [19] building a multi-hop wireless sensor network together with the MMS. Each IRIS mote was configured to generate one packet of 88 bytes every minute and to transmit it over the network.

![Graph showing current consumption](image)

Fig. 4. MMS current consumption during local data acquisition.

Table 2 summarizes the average current and energy consumption of each phase measured on both configurations. Note that for the two extension modules configuration there are overlapping effects between phases: when the master module sends the start conversion command to the second extension module, the first module is already converting. Similarly, during the data retrieval phase, when the master module starts to download the new data from the first module, the second one is still converting. This explains the differences in the average current consumption during the sample request and data retrieval phases of the two modules configuration with respect to the single one.

![Testbed map](image)

Fig. 5. Testbed map.

The MMS was set to locally sample one channel from each extension module generating 2 sensor
readings every minute in order to transmit the sampled data over the wireless medium at the same rate as the IRIS motes. All data packets are routed through the multi-hop network toward a gateway, which is responsible for data aggregation and storing. We used the Collection Tree Protocol [20] as routing algorithm and the BoX-MAC medium access control protocol [21], both provided by the TinyOS operating system. To save energy, the MAC protocol was set with a radio duty cycle of 2% which represents a common value when long lasting wireless sensor networks are deployed. The testbed ran for 5 hours and collected statistical information embedded in the packets transmitted every minute by each node.

Table 3 reports detailed power consumption statistics, averaged over all the IRIS motes and compared with the one of the MMS. We observe that the power consumption of the MMS is similar to the one of the IRIS motes. Despite these measurements strongly depend on the position of the MMS in the network topology, this similarity shows that our solution is comparable in terms of energy requirements to a common WSN platform. Furthermore, the energy consumption of the sampling activity is less than 1% of the overall consumption that is largely dominated by the radio activity. We are aware that different transducers (e.g., the current-loop family) can consume more energy during conversion. This would significantly impact the fraction of energy consumed by the sampling activity. However, this consumption is exclusively related to the adopted transducer and does not depend on our architecture.

### 6.3. Design Overhead

As discussed in Section 4, we adopted a new design for the low-power shared bus, increasing the flexibility and reducing the cost of the MMS. However, these advantages come at the expense of a higher management complexity of the system that caused an increase of the power consumption. Obviously, the higher is the number of extension modules, the higher is the complexity, thus the power consumption overhead increases accordingly. In this section, we evaluate the overhead of the new MMS design in terms of power consumption when compared to the original design.

Wake-up, polling and latency are the three main factors that cause the consumption overhead. We consider a configuration made of a master module and 4 extension modules performing a local measurement cycle as the one described in Section 6.1 to measure the contribution of each of those factors to the overall power consumption.

Wake-up: When the master module starts an I²C communication, it writes on the bus the address of the module it wants to communicate with. This activity wakes up all the extension modules that start an address matching procedure. Thanks to the new architecture of the M0+ MCU of the extension modules, the address matching procedure is carried out only employing the I²C peripheral integrated in the chip, without the need to awake the MCU (this mode is called sleepwalking [22]). This feature wakes up the MCU only upon an address match event, reducing the overall power consumption of the address matching procedure. At the standard I²C 100 kbit/s datarate, the address transmission requires 90 µs. Assuming 3.0 V and 50 µA current consumption of the sleepwalking peripheral, the energy required by an extension module to perform an address matching procedure on the low-power shared bus is 14 nJ. Considering that for each request there are 3 extension modules that do not match the address, the overall energy wasted for each request is 42 nJ.

Polling: The master module polls each extension module by looking at the data flag area which notifies whenever a new data is available. On the low-power shared bus this operation is performed as in Fig. 6: the master first send a write request followed by the register address of the data flag area and successively a read request.

![Fig. 6. Polling I²C communication detail.](image)

The extension module replies with the content of the data flag area and, if no data is available, the communication ends. The whole procedure, which is 8 bytes long, is performed in 840 µs on a standard 100 kbit/s datarate I²C channel. The measured current consumption of the MMS during a polling procedure is 4 mA that, assuming 3.0 V supply, brings to an energy of 10.1 µJ per extension module. In the
configuration under test, there are 3 polled extension modules over 4 that do not provide new samples, hence, the wasted energy is 30.3 μJ.

Latency: We observed that the new design of the low-power shared bus requires, on average, 6 additional control bits each 10 data bits transmitted with the old implementation. This constraint increases latency in communications and, in turn, power consumption. The reason why more control bits are needed relies on the more complex I2C bus management compared to the SPI protocol used by the old design. Assuming a local data measurement procedure as the one described in Section 6.1, the protocol transmits 6 additional control bytes. Considering 4 mA current consumption of I2C communication, 3.0 V supply and a standard 100 kbit/s I2C datarate, the additional energy for each extension module measurement introduced by latency is 5.8 μJ. Note that we are not considering the latency introduced by the different supported datarates of each protocol since it is limited by noise immunity requirements.

Based on these measurements, the overall energy overhead of the low power shared bus in a local measurement procedure is 36.14 μJ which represents a consumption overhead of 9% when compared to the overall energy consumption measured in a single extension module configuration of Table 2. Recalling that the sampling activity accounts for less than 1% of the overall energy consumption of the MMS, we can conclude that the energy overhead of the new MMS design is totally negligible.

7. Conclusion and Future Work

In this work, we presented further developments of the Modular Monitoring System, a novel low-power wireless modular architecture designed for SHM applications. Our platform, the MMS, takes advantage of a low-power shared bus connecting slotted extension modules that interact with a master in a master/slave communication abstraction. The extension modules, which can be combined as needed, allows the MMS to face the continuously evolving needs of most SHM scenarios. Thanks to its peculiar characteristics, the MMS overcomes commercial state-of-the-art WSN solutions for SHM, like the NI WSN, which do not offer enough flexibility to fulfill requirements of many application contexts in a both cost-effective and efficient way.

The new implementation overcomes the size and costs issues of the first prototype presented in [1] offering an improved architecture based on DIN standard modular housing. This solution also offers an increased flexibility by supporting interchangeable power sources. We validated the effectiveness of the system low-power design, performing energy measurements during data acquisition from actual transducers. In addition, we tested the MMS within a real WSN deployment, to demonstrate the compliance of the system in such applications.

In the near future, we plan the develop several additional extension modules able to support most of the sensors used in SHM, such as strain wire gauges, crack meters, inclinometers, displacement sensors, etc. and additional master modules to support different wireless frequencies (e.g., 433 MHz, 868 MHz, 915 MHz) and certified industrial wireless protocols like Wireless Hart.

References


[21]. D. Moss, P. Levis, BoX-MACs: Exploiting physical and link layer boundaries in low-power networking, Computer Systems Laboratory Stanford University, 2008, pp. 116-119.


2015 Copyright ©, International Frequency Sensor Association (IFSA) Publishing, S. L. All rights reserved. (http://www.sensorsportal.com)