Smart Sensor Platform for Industrial Monitoring and Control

Harish Ramamurthy, B. S. Prabhu and Rajit Gadh
Wireless Internet for the Mobile Enterprise Consortium
University of California, Los Angeles
Los Angeles, California, USA.

Asad M. Madni
BEI Technologies, Inc.
Sylmar, California, USA

***Abstract*— a wireless smart sensor platform (based on patent pending technologies "A Generic Wireless Transducer Interface" [1] and "Application of generic reconfigurable wireless interface for industrial automation scenarios" [2]) targeted for instrumentation and predictive maintenance systems is presented. The generic smart sensor platform, with 'plug-and-play' capability, supports hardware interface, payload and communications needs of multiple inertial and position sensors and actuators, using a RF link (Wi-Fi, Bluetooth, or RFID) for communications, in a point-to-point topology. The design also provides means to update operating and monitoring parameters as well as sensor/RF link specific firmware modules 'over-the-air'. Sample implementations for industrial applications and system performance are discussed.**

I. INTRODUCTION

Intelligent wireless sensor-based controls [2] have drawn industry attention on account of reduced costs, better power management, ease in maintenance, and effortless deployment in remote and hard-to-reach areas. They have been successfully deployed in many industrial applications such as maintenance, monitoring, control, security, etc [3]. In this research, the focus is to address the issues faced by instrumentation systems and predictive maintenance industrial applications and to design a solution to cater to the issues faced by these applications.

Instrumentation systems are open/closed loop control systems like motor control. They are formed using sensors and actuators and the objective is to control certain parameters, or state of the system. All the system elements are always in communication with each other, typically, requiring real-time performance. They also require in-built fault-tolerance for communication/node failure – to return to a safe-state in a deterministic amount of time.

Predictive-maintenance involves tracking physical state of equipment or machine, and to take action, if an acceptable or allowed state(s) is violated. Predictive-maintenance applications are not active all the time in order to conserve energy. The sensors are either periodic or event-based; they wake up, check status and go back to sleep. In case of any violation, they raise an alarm or record the digression. They are very useful in keeping machine down-times low and help locate the problem before the machine breaks down.

Both these systems employ different types of sensors (e.g., position, accelerometers, gyros, etc.) and actuators (e.g., motors) often deployed within the same network, having different capabilities, interfaces, and supporting different protocols for data and communications. Formation of systems from such diverse distributed sensor elements entails versatile control modules, which understand different sensor protocols and utilize them. In addition, the operational challenges are exacerbated when different RF links have to be used to satisfy the requirements of bandwidth, payload, delay, jitter, range, noise immunity and others (including cost) for communication.

The proposed Smart Sensor Platform is an attempt to develop a generic platform with 'plug-and-play' capability to support hardware interface, payload and communications needs of multiple inertial and position sensors, and actuators/motors used in instrumentation systems and predictive maintenance applications. Communication is carried out using a RF link (Wi-Fi, Bluetooth, Mote or RFID), in a point-to-point topology. The design also provides means to update operating, monitoring parameters and thresholds as well as sensor and RF link specific firmware modules 'over-the-air'. It is composed of two main components – a sensor-wireless hardware interface and system integration framework, which facilitates the defining of interaction between sensors/actuators based on process needs. The intelligence necessary to process the sensor signals, monitor the functions against defined operational templates, and enable swapping of sensor and RF link resides on the microcontroller of the hardware interface. A variety of industrial motion sensors like gyro, Inertial Measurement Unit (IMU), linear position, absolute and incremental encoders and actuators like motor, have been interfaced and successfully tested with the platform.

The organization of this paper is as follows. Section 2 covers related work on sensor networks, and specific initiatives for industrial automation. Section 3 describes the intelligent wireless sensor architecture. Implementation details, snapshots, simulation and experimental results of the current implementation are presented in Sections 4 and 5. Finally Section 6 reports the conclusions on the research.

II. RELATED WORK

The field of automation has continuously evolved-starting from early days of register level programming for data acquisition and point-to-point wired links for communication, to the current virtual instruments and Ethernet, a wired communication paradigm for networking industrial systems. Developmental efforts in this area can be broadly classified as:

A. Industrial Initiatives

Includes the design of industrial open protocols for wired communication also known as field buses like CAN, DeviceNet and ControlNet; proprietary system formation tools - Virtual Instruments from National Instruments, Factory solutions from ABB, etc. Further development involved open data exchange or messaging framework, for e.g. OPC foundation which is trying to establish a standard data exchange standard so that interoperability among products (hardware and software) from different manufacturers is achieved [4, 5]. Strong potential for wireless is envisaged in enterprise-wide asset monitoring and maintenance on an open protocol for communication like ZigBee. Industrial initiatives have focused on the system formation issues, but have been unable to exploit the advantages of wireless technology.

B. Academic Initiatives

Though wireless sensor networks have become mature [6, 7, 8, and 10], the focus has been on environment monitoring, military and homeland security applications. Though viability studies have been conducted for using wireless communication in these industrial applications, not much impetus has been given over full system deployment. Application-specific wireless implementations have been proposed but a generic system building approach has not been investigated.

Deployment of wireless infrastructure in industries will occur incrementally and interoperability (between different systems) and extendibility (different application needs) will form the requirements of prospective solutions. The smart sensor platform research initiative is an attempt to develop such an end-to-end solution with support for incremental deployment, extendibility and scalability.

III. Design

The motive of the smart sensor project is to create 1) a general purpose hardware interface for diverse sensors and actuators, which can be customized for any application through over-the-air firmware downloads and 2) create a data processing infrastructure at the backend to implement applications. The proposed solution consists of a network of sensors, and actuators communicating with the central control unit using standard RF-links. The basic scenario is shown in Figure 1. The sensors are directly connected to the central control unit (workstation here) through a RF link, which can be Bluetooth or WiFi.

Each sensor or actuator is equipped with a reconfigurable generic wireless interface or smart sensor interface. The interface extracts data from the sensors and commands the actuator and provides a data communication interface to the central control unit. A sensor/actuator coupled with smart sensor interface is termed as a smart sensor node.

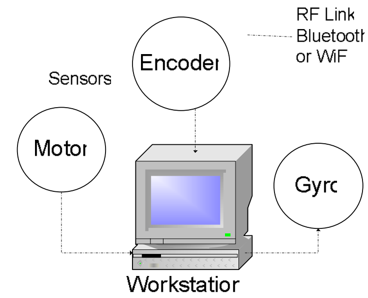


Figure 1. General Application Scenario

A. Smart Sensor Node: Hardware Design

The sensors/actuators found in industrial applications can be classified by analog, digital, serial (or combination of these) signals used for data communication. The smart sensor interface interprets sensors/actuators' signals, and converts it into digital data/commands. For this 14-bit 200ksps ADC, 8 channel 10-bit 9.6ksps ADC, DAC, 16 GPIO, and USARTs are used. The hardware design is shown in Figure 2.

B. Smart Sensor Node: Software Design

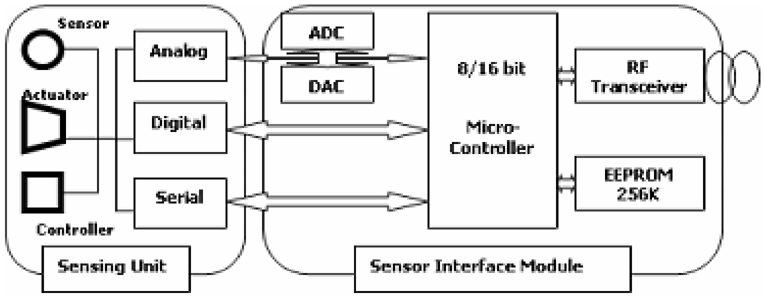
The digital data extracted by the hardware interface has to be bound by a context and processed to convert it into useful information. This intelligence is provided by the software that resides on the smart sensor interface.

The software design of the smart sensor interface is shown in Figure 2. The software module stack on the smart sensor interface consists of three layers. The bottom layer is the device driver which directly interfaces with the hardware interface and extracts digital data. The device manager interfaces with the device drivers and exposes a multiple-data channel interface to the firmware layer. In the software framework, each sensor/actuator is composed of a combination of digital, analog or serial channels. Establishment of context to the extracted channel data is done at the firmware layer. The firmware layer "composes" the sensor by combining data from multiple data channels. It also implements the application specific functionalities like real-time performance, data communication protocol with central control unit, smart sensor node management, etc.

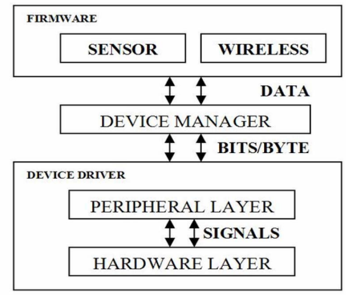
This separation of data acquisition tasks across three layers in the smart sensor interfaces helps support functionalities like over-the-air update of parameters, plug-n-play of sensors, multiple sensor support, multiple wireless technology support, universal data interface etc.

C. Application Integration Software

The application integration software resides on the central control unit and handles application-specific customization of the smart sensor nodes. Based on the Java



Hardware Design



Software Design

Figure 2. Smart Sensor Node Design

Beans framework, the software enables formation of systems from discrete smart sensor nodes. Specific description for real-time and predictive maintenance applications is provided in the implementations section.

IV. IMPLEMENTATION

A. Real-Time Control

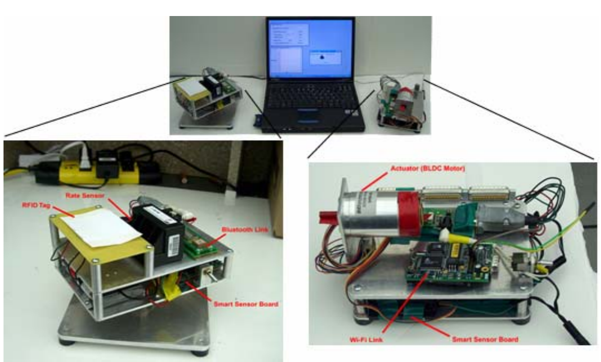
The objective of this implementation is to demonstrate the non-deterministic real-time performance of the smart sensor node. Deterministic real-time performance cannot be achieved with the smart sensor node as wireless communication is used, which is prone to errors. In order to achieve near real-time performance the smart sensor node tracks the traffic of wireless channels and uses a simple TCP-like congestion control scheme to regulate the traffic. Once the node senses congestion, high traffic, or connection loss, it brings the node into a "safe-state". The node then simply waits for the central control unit to reconnect or signal degradation to abate.

The system built for demonstration was a proportional gyro-motor-encoder system (Figure 3). In this proportional gyro-motor-encoder system, each sensor/actuator pair is connected to a smart sensor interface and uses Bluetooth to communicate with the central control unit. The gyro senses the angular tilt and communicates it to the central control unit, which in turn sends appropriate command to the motor. Further, the encoder attached to the motor tracks the position of the motor. In this application the safe state of the system is to bring the motor to a halt.

B. Predictive Maintenance:

A typical factory environment is considered, where the health of the machinery/equipment is regularly monitored and any digressions/violations from the tolerable behavior during operation are recorded. The recorded information of a machine typically consists of information like threshold violations, time of the event, extent of the event, etc. The status of machines is typically checked by a qualified machinist who inspects the machine when the main power has been switched off. Any proposed solution should thus operate passively and data should be stored locally.

In the current implementation, smart sensor nodes, equipped with sensors to monitor the status of a machine, store the health information in a RFID tag. RFID tag is used as a plain wireless non-line-of-sight data storage [9]. In this mode, the maintenance personnel can retrieve the required health information by querying the tag even when the central computer has been switched off.



Linear Sensor with RFID tag (in white color)

User Interface on a PDA

Figure 3. Implementation Snapshots

To demonstrate we consider as an example, an application scenario where every threshold violation of the linear sensor has to be recorded with timestamp. The threshold parameters are set through the application software module during deployment. We use the ISO15693 (13.56 MHz) tags for storing data. These tags have memory ranging from 256 bytes to 2KB. A handheld reader connected to a PDA is used to read tag data. On the PDA, the records are presented in a tabular format. Snapshots of the current implementation are shown in Figure 3.

V. EXPERIMENTAL RESULTS

Experiments were conducted to study the relevant performance metrics such as link delays, bandwidth with varying distance, traffic and packet bursts. For lack of space we only provide the delay experiments here. For detailed description of experiments please refer [11].

Round trip delay is an important characteristic of a control system. It places an upper limit on the responsiveness of a system. Delay performance of different wireless technologies was tested and is presented below (refer Fig 4).

To summarize, the effects of the following parameters on delay of Bluetooth are (refer Figure 4):

- Distance: With increasing distance, the delays become larger and jittery.
- Traffic: No considerable effect
- Packet Bursts: Mild effect with performance degrading with more packets per burst

For Wi-Fi

- Distance: Performance degrades with distance, delay increases and becomes jittery.
- Traffic: Performance worsens with increasing traffic. The effect is more pronounced at larger distances.
- Packet Bursts: As time to access channel is constant, bigger payloads experience less per-byte delays

Thus, Bluetooth seems to fit better in industrial application scenarios where limited bursts of data need to be delivered in real-time in a noisy environment. Wi-Fi seems to fit better in scenarios where huge amount of data need to be transmitted in a less noisy environment.

VI. CONCLUSIONS

A wireless smart sensor platform targeted for instrumentation systems and predictive maintenance was presented. Sample implementations for instrumentation systems and predictive maintenance applications were discussed and presented. Tests were carried out to determine system performance and were presented. The experimental results show that a sustained near-real-time system can be set up with the smart sensor nodes.

REFERENCES

[1] H Ramamurthy, XY Su, BS Prabhu and R Gadh, "A generic wireless transducer interface", US Patent Disclosure, UCLA Case No. 2005-245-1 (Pending).

[2] H Ramamurthy, B. S. Prabhu and R Gadh, "Application of generic reconfigurbale wireless interface for industrial automation scenarios", US Patent Disclosure, UCLA Case No. 2006-002-1 (Pending).

[3] U.S DOE, "Industrial Wireless Technology for the 21st Century" – Report, Technology Foresight, Winter 2004/ TF-2004-1.

[4] Ryan Wynn, "Plug-and-play sensors", Machine Design, 75(10), May 2003, pp. S10.

[5] *OPC HDA Specifications*, Version 1.20.1.00 edition, 2003, Available for download at: www.opcfoundation.org.

[6] IEEE Standard for a Smart Transducer Interface for Sensors andActuators- IEEE 1451.3 edition, 2004.

[7] J. M. Hellerstein, W. Hong, S. Madden, and K. Stanek, "Beyond Average: Towards Sophisticated Sensing with Queries," in *International Workshop on Information Processing in Sensor Networks*, March 2003.

[8] Y. Yao and J. E. Gehrke, "Query Processing in Sensor Networks," in First Biennial Conference on Innovative Data Systems Research (CIDR), January 2003.

[9] B. S. Prabhu, Xiaoyong Su, Harish Ramamurthy, Chi-Cheng Chu, and Rajit Gadh, "WinRFID – A Middleware for the enablement of Radio Frequency Identification (RFID) based Applications", Invited chapter in Mobile, Wireless and Sensor Networks: Technology, Applications and Future Directions, Rajeev Shorey, Chan Mun Choon, Ooi Wei Tsang, A. Ananda (eds.), John Wiley, December 2005.

[10] W. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," *IEEE Transactions on Wireless Communications*, vol. 1, pp. 660–670, October 2002.

[11] Harish Ramamurthy, B. S. Prabhu, Rajit Gadh, "Experimental studies on Bluetooth and Wi-Fi radio links for real-time control", Tech Report, WINMEC-REWINS-006, Dec 2004.

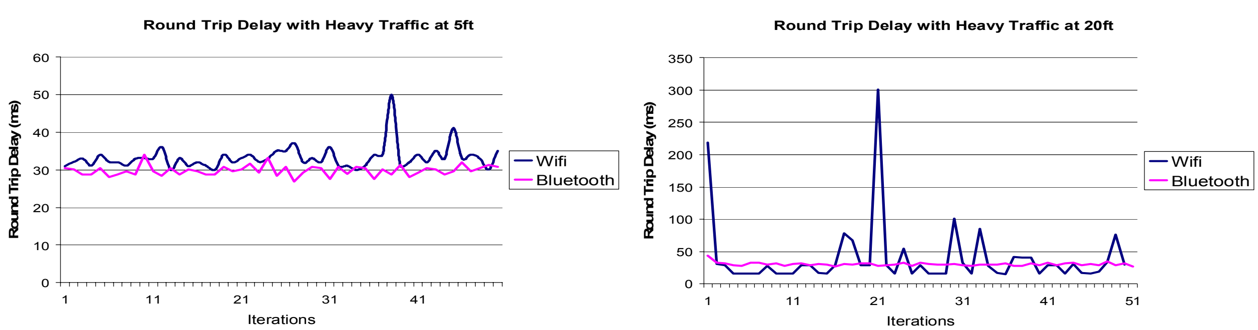


Figure 4. Experimental Results: Round-Trip Delay