Wireless Sensor Data Acquisition Design for Engineering Applications

Adam Lewis, A. Abdelgawad, and **N.-F. Tzeng**
Center for Advanced Computer Studies
University of Louisiana at Lafayette, Lafayette, LA 70504, U.S.A.
{awlewis, ama1916, tzeng}@cacs.louisiana.edu

Abstract

Wireless sensor data acquisition (WSDA) is preferred over its wired counterpart due to its easier deployment and simpler maintenance. These characteristics make WSDA attractive to a wide array of engineering applications. This article deals with a framework for WSDA comprising both legacy sensors and modern sensors commonly found in the oil and gas sector. The framework includes three components: a wireless receiving and transmission (ReT) module, a signal conditioning and digitizing (CoD) module, and a management and control (MaC) module. One ReT module implementation based on TinyOS and Crossbow MICA2 motes has been completed with a goal to integrate different legacy sensors usually existing in a given oilfield, via the aid of proper CoD module implementations. This framework design and its module implementations serve as the basis for fast WSDA prototyping that is key to the oil and gas sector. The introduced WSDA framework is readily applicable as well to other engineering applications where legacy and modern sensors co-exist.

1 Introduction

Engineering applications often call for acquiring data gauged by various sensors placed in the field or attached to the objects of interest. Energy exploration and production (E&P) is among such applications relying heavily on remote monitoring of production sites equipped with sensors. Many types of legacy sensors have been deployed by the E&P industry, ranging from flow meters, pressure sensors, and temperature sensors to acoustic sensors for sand detection. Those legacy sensors and measurement instruments for production site monitoring are connected together typically using analog 4-20 mA wires linked by a programmable logic controller (PLC). Recently, oilfields have gradually adopted the "fieldbus" technology where the PLC is replaced with a multipoint mechanism similar to local-area switch networking.

Meanwhile, recent advances in wireless and microelectronic technologies enable the fabrication of wireless sensor nodes, which incorporate sensor signal digitization and transmission logics. These modern sensor nodes have seen their growing deployment in the oilfields for gathering additional data or for replacing legacy sensors used therein. It becomes critical to design and develop a framework for fast prototyping wireless sensor data acquisition (WSDA) composed of both legacy and modern sensors. Such a WSDA framework aims to integrate all types of production sensors into one wireless sensor network for efficient data acquisition and remote management. It encompasses seamlessly both sensors with analog control & signaling logics and devices with fieldbus-based digital interfaces [1].

Our proposed WSDA framework includes three components: a wireless receiving and transmission (ReT) module, a signal conditioning and digitizing (CoD) module, and a management and control (MaC) module. While different implementations of the ReT module are possible, this article discusses in detail an implementation based on TinyOS [2] and Crossbow MICA2 motes [3]. Other implementation options for the ReT module will be discussed in Sec. 2 as well. A CoD module has been implemented to work with two legacy sensors for the E&P application: a flow meter and a differential pressure transmitter (DPT). These devices produce different types of analog signals: the flow meter communicates with other devices using DC voltages while a DPT generates DC current outputs proportional to the measured differential pressure. In fact, DC voltages and DC currents represent all possible types of analog outputs of legacy sensors. The MaC module serves to manage and control, in a unified manner, the sensor network formed (by constituent, diverse sensors) and the sensor data acquisition rates. It should be noted that this proposed WSDA framework is readily applicable to various engineering applications where legacy and modern sensors co-exist.

2 WSDA Framework

Our proposed WSDA (wireless sensor data acquisition) framework is depicted in Figure 1, where the ReT module and the CoD module can be implemented in many forms. The CoD module permits (1) various legacy sensors to provide their data readings in a unified way to the ReT module for transmission, and also (2) received data/control signals to be converted back to their respective analog signal forms and ranges (in voltages or DC currents) native to those legacy sensors involved. The ReT module deals with receiving and transmitting values in a given resolution (say, 12 bits). Upon reception, it interfaces with the CoD module, which converts the received values before forwarding to

This work was supported in part by the U.S. Department of Energy under Award Number DE-FG02-04ER46136 and by the Louisiana Board of Regents under Contract Number DOE/LEQSF(2004-07)-ULL.

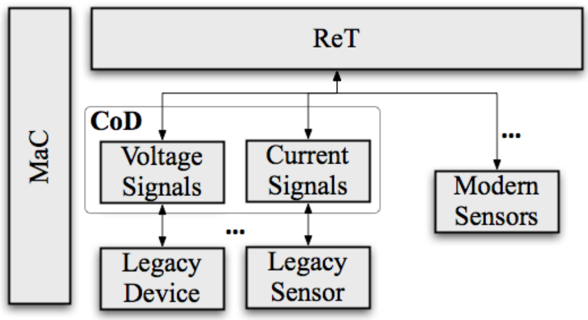

Fig. 1. WSDA framework.

corresponding legacy sensors. In addition, it conditions and digitizes the analogous sensor readings to the given resolution for transmission.

As it is responsible for wireless signal reception and transmission, the ReT module may be implemented in various fashions. For example, one may build it using a Crossbow's mote which is governed by the XMesh communication protocol realized in TinyOS [2]. Alternatively, one may choose to implement it by means of a different RF gear set following Zigbee (a standard application protocol). Another option is to employ a pair of SAW-based RF transmitter and receivers with an integrated encoder and decoder logics, suitable for remote control/command, security, and automation (without incorporating any communication protocol).

The CoD module provides a linkage between legacy sensors and the radio module defined by ReT. It includes multiple components: a hardware circuit that conditions and normalizes analogous signals from legacy sensors, a logic unit to convert conditioned signals to digital forms (of a given resolution), and software which processes digitized values into ones ready for transmission by ReT. Same components are also charged to deal with received values, involving process them back to proper forms before converting them into proper analog signals for use by corresponding legacy sensors.

As management and control on sensor data are basic to WSDA, our developed framework includes a MaC module which facilitates sensor data readings to be

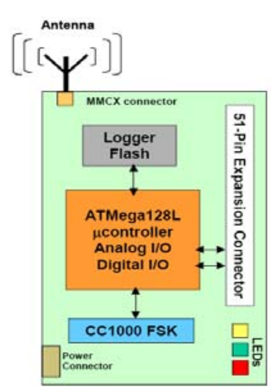

Fig. 2. Block diagram of the MICA2 mote.

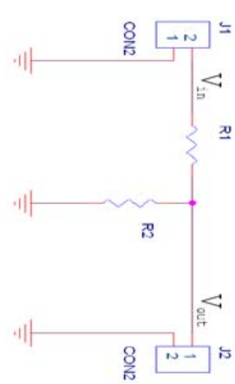

Fig. 3. Voltage dividing circuit.

logged and displayed in a uniform fashion, no matter whether they are from legacy sensors or modern ones. The MaC module also enables quick additions and modifications to data collection types and amounts when involved sensors and devices change. It comprises multiple units implemented in the form of portlets as detailed later in Sec. 7.

Key to the WSDA framework is its well-defined module functionalities and interface specification between CoD and ReT. CoD does not need to be aware of the details of the radio transmission but need only know how to pass (or receive) the data to (or from) ReT on transmission (or reception). Similarly, ReT knows nothing about the details of the analog to digital conversion process, the native analog signals (in voltages or DC currents), or the signal values ranges. It only needs to know that a CoD has registered with it and is to provide the data in a normalized format (following the given resolution).

The WSDA framework allows a level of interchangeability between components. The hardware circuits used to condition and normalize a 4-20 mA signal can be reused with different ReTs that use entirely different communication protocols. Thus, CoDs that operate with ReTs that use the default TinyOS protocols can be used with other ReTs that implement standard Zigbee application protocols.

While certain modern sensors incorporate the functionality of CoD, this framework permits to integrate data acquisition over those modern sensors together with their legacy counterparts through ReT, with acquired data managed and controlled by MaC. As can be seen in Fig. 1, the proposed framework is suitable for any engineering application where various types of sensors co-exist, since its ReT interfaces seamlessly with legacy sensors via CoD and with modern sensors directly.

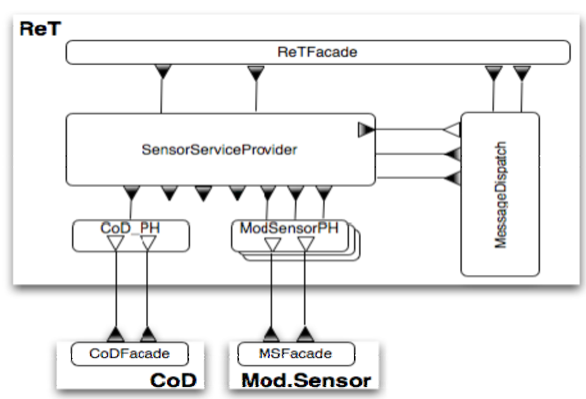

Fig. 4. ReT component design.

3 Software Architecture

Software reusability has long been sought for improved productivity, shortened time-to-market, and lowered development and testing costs. It can be enhanced through a set of proven solutions for recurrent problems, known as *design patterns*. In general, software design patterns do not specify implementation details, making it possible to accommodate various scenarios via reusing their high-level abstractions [19].

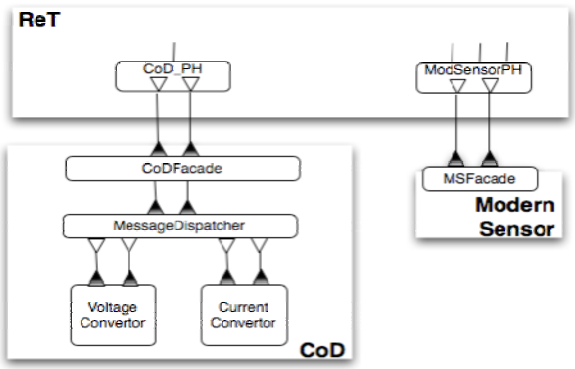

Fig. 5. CoD component design.

To obtain sound reusability, ReT is implemented to consist of three software design patterns: the Façade, ServiceInstance, and Dispatcher patterns [17, 18], as illustrated in Fig. 4. The Façade design pattern is used when one needs a single, unified interface to a collection of sub-services. In our case, the entry point to ReT for other applications is the ReTFacade application interface. Applications use this interface to post requests for launching data acquisition from the devices, initiating data transmission/reception, and registering new devices. The ReTFacade component passes these messages either to the SensorServiceProvider for administrative requests or to the MessageDispatcher for all other messages. The SensorServiceProvider is responsible for keeping track of each instance of the CoD or ModernSensor acquiring data. The MessageDispatcher is aware of each sensor connected to the system via a registration interface in the dispatcher, which is called when SensorServiceProvider instantiates a new sensor instance. Messages are then routed directly to the sensor by the MessageDispatcher component. ReT is aware only of a Proxy object for each sensor. This instance of the Proxy pattern [18] handles the interface between ReT and the devices. A single proxy will be place for CoD while multiple instances may exist for each ModernSensor to which the ReT communicates.

CoD may collect data from more than one instance of either a Voltage Signals device or Current Signals component, as depicted in Fig. 5. Here, a variation of the ServiceInstance pattern [17] is developed, with CoDFacade also serving as the ServiceProvider component of a multi-to-multi resource multiplexer/ demultiplexer. The ServiceProvider component realizes the service type of corresponding sensors, and it varies with the type of sensors.

4 Implementation of ReT Module

As a proof-of-concept attempt, one implementation of the ReT module has been accomplished by using Crossbow Technologies MICA2 motes and the TinyOS operating system [2]. Small and energy-efficient, TinyOS was developed by UC Berkeley as an open-source code in support of large-scale, self-configuring sensor networks. It offers a rich set of provisions for integrating diverse peripherals.

As the hardware component of our ReT implementation, MICA2 motes employ the Chipcon CC1000 FSK modulated radio and come in three models, according to their RF frequency bands: the MPR400 (915 MHz), MPR410 (433 MHz), and MPR420 (315 MHz). All models utilize an Atmega128L micro-controller and a frequency-tunable radio with an extended range, as depicted in Fig. 2. A variety of modern sensors and data acquisition boards can be connected to a MICA2 mote.

Communication is two-way between the ReT and connected CoD. The ReT provides an interface that can be used to pass commands to the ReT and any connected sensors. For example, radio transmission power can be adjusted by sending a command to the ReT.

The interface between the ReT and other components in the framework reduces the effort required to link with existing industrial control systems. The application programmer need only implement the ReT to CoD interface in order to link a new device into the ReT's network. In a similar fashion, it is possible to replace an existing ReT with a ReT that communicates to an entirely different network.

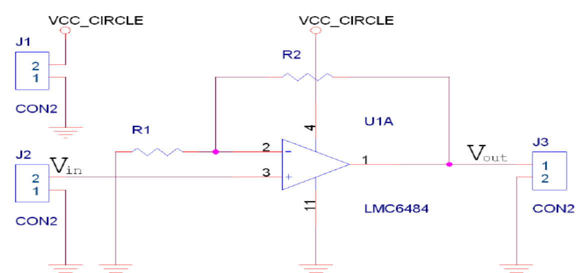

Fig. 6. Voltage amplification diagram.

5 Implementation of CoD Module

As legacy sensors and measurement devices usually produce analog outputs in the form of voltages or 4–20 mA DC signals, our CoD module includes circuits for output conditioning and amplification to a specified voltage range before A/D conversion is applied [12]. Two sets of circuit design have been devised and implemented: one for voltage conditioning and amplification and another for DC signal conditioning and conversion to voltage, as follows.

Voltage Outputs

An amplifier circuit is designed to amplify the flow rate signal from 1V to 2.5V before digitized. As illustrated in Fig. 3, an LMC6484 CMOS quad rail-to-rail input and output op-amp is used in the amplifier circuit, providing a common-mode range that extends to both supply rails. This rail-to-rail performance combined with excellent accuracy (due to a high common-mode rejection ratio) makes it unique among rail-to-rail input amplifiers. It is ideal for systems, such as data acquisition, that require a large input signal range. Maximum dynamic signal range is assured in low voltage and single supply systems by the LMC6484's rail-to-rail output swing, which is guaranteed for loads down to 600Ω. This guaranteed low load characteristic and its low power dissipation make LMC6484 especially well-suited for battery-operated systems. The circuit demonstrated in Fig. 3 was among application examples given in the LMC6484 op-amplifier data sheet [9].

Given the gain of the circuit is calculated by the next equation, the values of Resistors R_1 and R_2 equal to 10 KΩ and 15 KΩ bring V_{in} of 1V to V_{out} of 2.5V, as required:

$$\frac{V_{out}}{V_{in}} = 1 + \frac{R_2}{R_1} \quad (1)$$

It should be note that this designed circuit is <u>universal</u> for conditioning and amplifying voltage outputs of any legacy sensor, no matter what its output voltage value might be. As the A/D converter takes 2.5V as its input [12], a legacy sensor's voltage output can always be rectified to the proper range before conversion through choosing appropriate R_1 and R_2.

<u>Voltage Inputs</u>

For a legacy sensor with voltage inputs to receive control information or data wirelessly, CoD employs a voltage dividing circuit shown Fig. 6, with R1 and R2 governed by the equation below:

$$V_{out} = V_{in} \frac{R2}{R1 + R2}$$

so that the received information and data, after being converted to an analogous form, can be conditioned to match the voltage range suitable for the sensor input.

<u>Current Outputs</u>

The 4–20 *mA* current loop has been a popular sensor output form for industrial and process sectors alike. Its popularity comes from its ease of use, its performance, and its simple wiring (requiring just the two wires). Both the supply voltage and the measuring current are over the same two wires. This current loop makes cable break detection simple: if the current drops to zero, a cable break happens. Additionally, the current signal is immune to any stray electrical interference (unlike a voltage signal), deemed particularly important to sensors applied to harsh environments. Those advantages make the current loop output common to many legacy sensors.

A current-to-voltage converter translates current signals to proportional voltage outputs. It includes an operational amplifier for simple linear signal processing and a resistor for dissipating current, as depicted in Fig. 7. The resistance between the operational amplifier's input and output determines the voltage range for specific current signals. In current-to-voltage converters that handle a range of currents, design consideration accounts for the DC offset caused by both the input device and the operational amplifier. The output voltage of the circuit is calculated by the equation of $V_{out} = R_1 \times I_{in}$, giving rise to V_{out} in the range of 0.4V to 2V, for R_1 = 100 Ω when the current loop of 4–20 mA is applied. To minimize the bias current, R_2 is chosen to equal R_1. This circuit yields a desirable voltage range for A/D conversion, according to the current loop reading produced by a legacy sensor. The circuit depicted in Fig. 6 was an application example listed in the LM741 op-amplifier data sheet [10]. Since the 4–20 mA current loop is the popular output form of sensors deployed in many engineering applications, this devised CoD circuit is generally suitable for those applications.

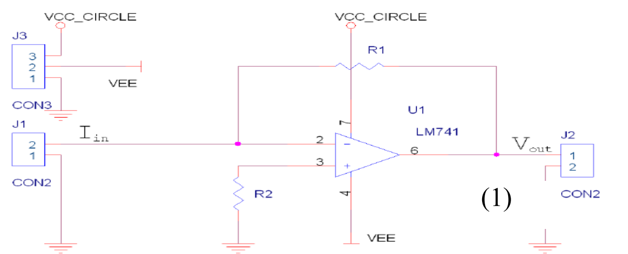

Fig. 7. Current-to-voltage converting circuit.

<u>Current Inputs</u>

For a legacy sensor with current loops as its inputs, a dual circuit to Fig. 7 is needed for converting voltages, which are obtained from digitalized data or control information received wirelessly, to 4–20 mA DC currents. As shown in Fig. 8, a voltage-to-current converting circuit can be included in the CoD module. The circuit was an application example given in the AM422 op-amplifier data sheet [11]. The AM422 is a low cost monolithic voltage–to–current converter specially designed for analog signal transmission. In order to get an output current (I_{out}) range of 4–20 mA with the input DC voltage (V_{in}) ranging from 0.4V to 2.0V, the circuit discrete components of Fig. 5are specified as follows: R0 = 25 Ω, R3 = R4 = 33 KΩ, RSET = 2.64 KΩ, R5 = 40 Ω, R1/R2 = 2.25, RL = 0 – 500Ω, C1 = 2.2μF.

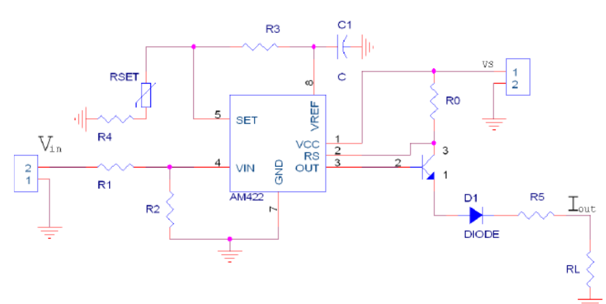

Fig. 8. Voltage-to-current converting circuit.

6 Case Studies

6.1 Flow Meter

One of the basic quantities measured in petroleum production and testing is the volumetric flow though pipelines. A common device used to measure the volumetric flow rate determines its reading by utilizing the linear relationship between the rotational frequencies of a turbine and the linear flow rate over a specified range of flow values. The NuFlo Measurement System's Model MC-II Flow Analyzer, as illustrated in Fig. 9, makes use of the relationship to produce flow readings [5]. It is one of our target devices.

The flow meter generates a pulse output signal of 1V at a rate that can be controlled by flow meter's totalizer firmware. In other words, one can configure the device to scale the volume increment at the amount from 1.001 units to 100.00 units, with one increment to emit one pulse. Our devised circuit shown in Fig. 3 serves to condition and amplify the outputs of this flow meter.

6.2 Differential Pressure Transmitter

The accurate measurement of differential pressure (DP) is required at many points in the oilfield. In general, DP is defined as a measurement of fluid force subtracted from a higher measurement of fluid force (in terms of pounds per square inch).

The Yokogawa Electric Corporation Model EJA110A differential pressure transmitter (DPT, as demonstrated in Fig. 10) has been adopted by many E&P facilities [6]. The core of this DPT is a silicon resonant sensor, which is very reliable even under extremely harsh conditions often found in the oilfield. The output of EJA110A is a 4 – 20 mA DC signal corresponding to the measured differential pressure (conveyed as a standard 4–20 mA current loop). Our circuit (see Fig. 5) converts DPT outputs into DC voltages ranging from 0.4V to 2V, ready for digitization.

Fig. 9. MC-II flow meter.

Fig. 10. EJA100A DPT.

7 Management and Control

Data values collected from sensors and measurement devices are transmitted over a WSDA system normally to one (or multiple) designated sink node(s). To facilitate management and control on data acquisition, a generic management portal is developed as part of our WSDA framework, referred to as the management and control (MaC) module.

The MaC module enables quick additions and modifications to data collection types and amounts when involved sensors and devices change. It interfaces with an application server, called XSERVE (which is responsible for collected data logging into an archival database), to permit data displays and processing. MaC consists of three units: (1) the Well Data Sensor portlet, (2) the Charts portlet, and (3) the Sensor Network Configuration portlet that together realize required functions to manage WSDA.

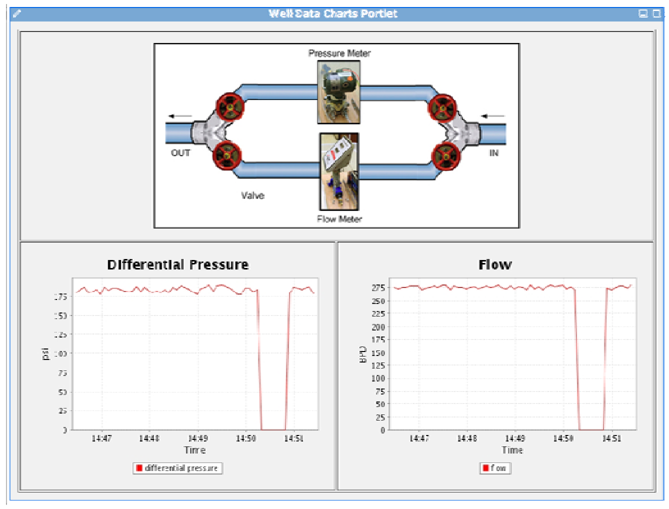

Fig. 11. Xsensor portlet.

The three portlets are constructed using Gridsphere (which is an open-source Java-based web portal [7]) and its accompanying GridPortlets. Gridsphere eases the development and deployment of portlet web applications for efficient administration. It provides tools needed for fast development and prototyping of the MaC portlets.

As illustrated in Fig. 11, the Well Sensor Data portlet displays the current raw data collected from the flow meter, DPT, and other modern sensors (e.g., Crossbow's temperature, light, and acceleration sensors). The data values are presented in a uniform, human-readable table form, no matter whether they are from legacy sensors or modern ones.

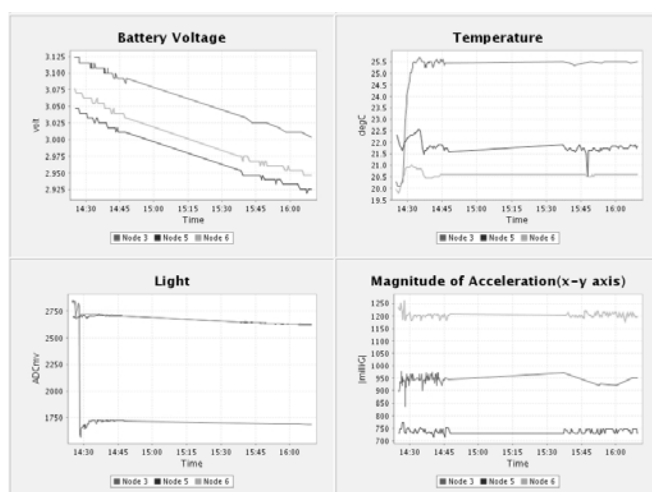

Fig. 12. Charts portlet.

The Charts portlet exhibits charts of a variety of collected data from the database server, as demonstrated in Fig. 12, where data shown were collected from two different locations. The portlet has the ability to display the data under different filters: time, range exceptions, and sensor node ID, where the latter two are yet to be completed.

The Sensor Network Configuration portlet enables one to modify the communication parameters governing connections to XSERVE and the database server. For example, the user can select the server IP address, TCP port used to connect to the database, and the TCP port used to connect to the sensors. It can also be used to specify the appearance of other portlets; e.g., the number of raw data packets shown in the Xsensor porlet.

8 Related Work

A software pattern is a description or template for solving a problem that can be used in many different situations. The concept of a pattern was adopted from the realm of architecture and was popularized in computer science by [18]. The concept of design patterns was extended into the embedded space earlier [20].

The design of our WSDA framework is inspired by the hardware abstraction architecture proposed earlier [13] and implemented in TinyOS 2.0 [14]. Our framework follows the same pattern of separating concerns among a hardware presentation layer, adaptation layer, and interface layer, with clearly defined interfaces between the layers.

The interoperability of components in our framework happens due to the flexibility of software units and generalized hardware circuits involved in interfaces. The interfaces have been designed using a workflow model employing various techniques described in [15, 16].

9 Conclusion

A framework has been introduced for wireless sensor data acquisition (WSDA), capable for handling legacy sensors and modern sensors uniformly. The framework consists of three components to enable data acquisition of all types of sensors: a ReT (wireless receiving and transmission) module, a CoD (conditioning and digitizing) module, and a MaC (management and control) module. CoD involves conditioning and amplifying circuits to deal with DC voltage outputs and current loops produced by all possible legacy sensors, before they can be digitized for wireless transmission. Two dual circuits are employed to deal with digitized data received, namely, converting the data back to their corresponding voltage ranges or current readings of the legacy sensors. This framework design and its module implementations serve as the basis for fast WSDA prototyping, readily applicable to various engineering applications employing both legacy and modern sensors.

10 References

[1] Fieldbus Foundation, "Foundation: System Engineering Guidelines." URL – http://www.fieldbus.org.

[2] P. Levis *et al.*, "TinyOS: An operating system for wireless sensor networks," *Ambient Intelligence*, 2004.

[3] Crossbow Technology Inc., "MICA2 Datasheet." URL – http://www.xbow.com.

[4] W. Cai, X. Jin. Y. Zhang, K. Chen, and J. Tang, "Research on Reliability Model of Large-Scale Wireless Sensor Networks," *Proc. of Int'l Conf. on Wireless Communications, Networking and Mobile Computing*, pp. 1 – 4, Sept. 2006.

[5] NuFlo Measurement Systems. URL – http://www.c-a-m.com/content/products.

[6] Yokogawa Electronic Corp., "EJA110A Differential Pressure Transmitter." URL – http://www.yokogawa.com/fld/PRESSURE/EJA/fld-eja110a-01en.htm.

[7] GridSphere. URL – http://www.gridsphere.org.

[8] W. Ahmad and M. Khan, "A Digital Technique for Measurement of Differential Pressure and Flow," *IEEE Transactions on Industrial Electronics and Control Instrumentation*, vol. IECI-25, no. 1, pp. 26–29, Feb. 1978.

[9] National Semiconductor Corp., "LMC6484 CMOS Quad Rail-to-Rail Input and Output Operational Amplifier," Jan. 1996. URL – http://www.datasheetcatalog.com/.

[10] National Semiconductor Corp., "LM741 Operational Amplifier," Aug. 2000. URL – http://www.national.com/.

[11] Analog Microelectronics, "Vlotage/Current Converter AM422," Apr. 1999. URL – http://www.analogmicro.de/.

[12] J. Hauer, P. Levis, V. Handziski, and D. Gay, "TinyOS Extension Proposal 101: Analog-to-Digital Convertors (ADCs)." URL – http://www.tinyos.net/tinyos-2.x/doc/html/tep101.html.

[13] V. Handziski, J. Polastre, J.-H. Hauer, C. Sharp, A. Wolisz, and D. Culler, "Flexible Hardware Abstraction for Wireless Sensor Networks," *Proceedings of the Second European Workshop on Wireless Sensor Networks*, pp. 145-157, Feb. 2005.

[14] P. Levis *et al.*, "T2: A Second Generation OS for Embedded Sensor Networks," *Technical Report TKN-05-007*, Telecommunication Networks Group, Technische Universitat Berline, Nov. 2005.

[15] L. Warner, A. Junior, F. Polpeta, and A. Frohlick, "Operating System Support for Handling Heterogeneity in Wireless Sensor Networks," *Proc. of 10th IEEE Conference on Emerging Technologies and Factory Automation*, Sept. 2005.

[16] L. Warner, A. Junior, A. de Oliverira, and A. Frohlich, "Operating System Support for Data Acquisition in Sensor Networks," *Proc. of the 11th IEEE Conference on Emerging Technologies and Factory Automation*, pp. 582-585, Sept. 2006.

[17] D, Gay, P. Levis, and David Culler, "Software Design Patterns for TinyOS," *ACM Transactions on Embedded Computing Systems*, vol. 6, no. 4, Sept. 2007.

[18] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995.

[19] F. Rincon *et al*., "Model Reuse through Hardware Design Patterns," *Proc. of IEEE Design, Automation, and Test in Europe Conference and Exhibition* (*DATE '05*), pp. 324-329, Mar. 2005.

[20] B. P. Douglass, *Real-Time Design Patterns: Robust Scalable Architecture for Real-Time Systems.* Addison-Wesley, 2002.