ESL-TR-01/12-02

INTERNET DATALOGGING AND DISPLAY

LoanSTAR Deliverable Report

Final Report

James Sweeney, Jr.
Charles Culp, Ph.D., P.E.
Jeff S. Haberl, Ph.D., P.E.

December 2001

PREFACE

This report is one of a series of reports that documents the development of the LoanSTAR and Technical Assistance deliverables. The developments are broken down into two divisions, Task C and Task D. The following two tables itemize the deliverables for each Task. The information included in this report represents the LoanSTAR and Technical Assistance deliverable Task C-Number Four.

Task C Deliverables	**Description**
1.Access and Display LoanSTAR Data via the Internet for Interested Agencies	Energy Systems Lab's web-based interface to the LoanSTAR Energy Databases, referred to as the Energy and Environmental Data System (EEDS).
2.Report Print/ Viewing Options (i.e., PDF format)	Effort to move reporting to Adobe Inc.'s Portable Document Format.
3.Online Feedback Ability (automated email to ESL)	Online feedback form built into the ESL's EEDS.
4.Internet Data Logging and Display a.Develop Internet-Based Real-time Display Software b.Buy/Develop and Test a Date Logger with Integrated TCP/IP Connectivity c.Buy/ Develop and Test Software to Poll a TCP/IP Logger	Research and development of internet based power measurement and energy data logging techniques.

Task D Deliverables	**Description**
1.Automated Email Polled Data Problem Alerts. (ESL internal)	The automatic notification via email of the sites that failed polling for a given polling job.
2.Automate IPNs Review Process to an 'Exception Only' Basis (ESL internal)	Streamlining the IPN review process by improving problem tracking and reporting.
3.Convert Ipns plots to PDF format (ESL internal)	Effort to move IPN reporting to Adobe Inc.'s Portable Document Format.
4.Automated the Data Analysis/Quality Verification Process to 'Exception Only' (ESL internal)	A series of checks that monitor raw files loaded into the database for low level data quality analysis.
5.Move IPNs, MECRs, AECRs to an Internet Base	Effort to combine reporting in Adobe Inc.'s Portable Document Format with a secure Web-based interface.

EXECUTIVE SUMMARY

The information included in this report represents the LoanSTAR and Technical Assistance deliverable Task C-Number Four, "Internet Data Logging and Display". The papers included in this report represent the work of several individuals at Texas A&M University's Energy Systems Lab and Department of Electrical Engineering. "An Internet Based Power Measurement Technique" was published in The Institute of Electrical and Electronic Engineers Instrumentation and Measurement Technology Conference in Budapest, Hungary, May 2001. It represents the combined research efforts of James Sweeney and Dr. Charles Culp of the Energy Systems Lab and Dr. Mark Yeary, Benjamin Swan and Larry Archer of the Electrical Engineering Department. "Technology Enablers for Next-generation Economic Building Monitoring Systems" was presented at the International Conference for Enhanced Building Operations in Austin, Texas, May 2001. It represents the work of James Sweeney and Dr. Charles Culp of the Energy Systems Lab.

ACKNOWLEDGEMENTS

This project would not have been possible without the support that was provided by the following persons and/or the agencies or companies for which they work: Ms. Theresa Sifuentes and Mr. Dub Taylor at the Texas State Energy Conservation Office,

This project also would not have been possible without the support that was provided by the following persons that work at the Texas A&M University's Energy Systems Lab: Ms. Kim Carlson and Mr. Kelly Milligan.

Finally, this project would not have been possible without the support that was provided by the following persons that work at the Texas A&M University's Department of Electrical Engineering: Dr. Mark Yeary and the Summer 2000, Electrical Engineering 485 InterDAQ Research Team.

IEEE Instrumentation and Measurement
Technology Conference
Budapest, Hungary, May 21-23, 2001

An Internet Based Power Measurement Technique

M. Yeary[1], B. Swan[1], J. Sweeney[2], C. Culp[2], L. Archer[1]

[1]Dept. of Electrical Engineering, Texas A&M University,
College Station, TX 77843-3128, USA
tel: +1 979-862-1640 fax: +1 979-845-6259
e-mail: mbyeary@ee.tamu.edu, imer@tamu.edu, larcher@tamu.edu

[2]Energy Systems Laboratory, Dept. of Mechanical Engineering
Texas A&M University, College Station, TX 77843-3581, USA
tel: +1 979-847-8781, email: {jsweeney,cculp}@esl.tamu.edu

Abstract *– The current energy savings technology relies on conventional data logging systems, in which two major barriers exist. Formost is the fact that retrieving the energy data is not convenient, and the cost of the data logging equipment is high. The solution presented here is to accomplish these goals is to include a miniature web server in a remote-logging module, which we designed as part of our device. Thus allowing data to be accessed more frequently, via the Internet. As it currently stands, the state of California in the United States is experiencing power grid problems as residential and industrial energy demands increase. If an energy savings program is to be implemented, then an energy monitoring strategy must also devised. Our Internet appliance provides a solution, and this paper summarizes our implementation details and provides a computer screen-capture of the data being posted onto the Internet.*

Keywords *– Internet appliance, data acquisition, Java, energy measurement*

I. INTRODUCTION

The measurement of a building's power consumption is important to increase the efficiency of its energy consumption and to aid in the future design of ecologically conscientious buildings. Monitoring power consumption is currently limited to larger buildings because of the high cost of associated equipment. Monitoring a building along with regulating its energy consumption, also known as continuous commissioning, can reduce a buildings energy consumption by 10% to 40%. The Energy Systems Laboratory at Texas A&M University has applied this technique to over 290 buildings and obtained an average savings of 22% [1].

Since applying the abovementioned energy savings technology relies on conventional data logging systems, two major barriers exist. Formost is the fact that retrieving the energy data is not convenient, and the cost of the data logging equipment is high.

Our device, called the InterDAQ (i.e., Internet Data AcQuisition), combats both of these problems and is a complete redesign of the data collection system. Data collection in remote buildings will be less-expensive, faster, and more versatile. The cost reduction will be reduced from $2,000 to $3,000 for existing devices to $400 to $500 for InterDAQ. Since dial-up, with its associated cost, is no longer required, data will be accessed more frequently, via the Internet. The proposed solution to accomplish these goals is to include a miniature web server in a remote-logging module, which we designed as part of InterDAQ. Dallas Semiconductor has such a miniature web-server called a Tiny Internet Interface, which is a Java server (i.e., embedded system designed to run Java applications and interface between external hardware and a network). This device will be referred to as the Internet interface unit throughout this paper. InterDAQ used the Internet interface unit to control a measuring station and then uploaded the data to a web server in two formats for easy accessibility.

The measuring module contains a voltage transformer, a current transducer, a Watt-hour transducer and a temperature sensor. In addition it contains a set of one amplifier and two rectifying circuits in order to read AC voltages and currents. This is connected to the 1-Wire compatible T8AD analog-to-digital (A/D) converter. Since each of these measuring modules has an A/D converter with a unique 64 bit identification code, Java programs run on the Internet interface unit are able to poll specific measuring modules storing the data. The Internet interface unit also manages the taking of voltage and amperage measurements, calculating power (V*I), and writing this information along with a timestamp into text format. The Internet interface unit can be accessed through the Internet via any browser or file transfer protocol. An overall block diagram is seen in Figure 1 below.

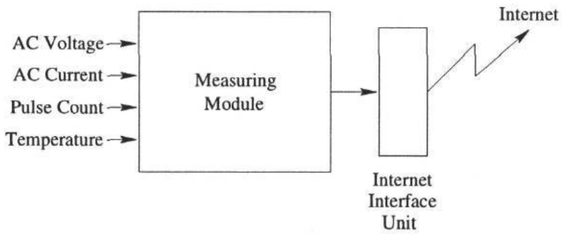

Fig. 1. Overall Block Diagram

II. STATE-OF-THE-ART

Remotely located data loggers have been used extensively for collecting measurement data [2][3][4][5][6]. Convenient retrieval of the data from the data logging device has always presented a problem [7][8]. One solution, for example, takes advantage of the recent advances in EEPROMs. The technique electronically stores logger data in the RAM (random access memory) of an EEPROM, which may then download the data to a personal computer [9]. Data storage at the remote site with access via dial-up lines adds constraints of cost and not being able to view data in real-time, phone line unreliability, and noise problems.

In the past, there has been an interest in logging the temperature information of a building in order to optimize the heating and cooling of the building [2][10]. Temperature measurements along with energy consumption measurements have been used to successfully reduce energy consumption in buildings [1]. The InterDAQ device described in this paper provides a solution to the above mentioned challenges since data will be automatically formatted into an HTML format that facilitates very convenient access from the Internet. Moreover, since our device uses a Java program for control, the data from several different sensors may be captured and calculations may be made before data is accessed by a central server on the Internet.

As the Internet gains in prominence, the use of Internet appliances to gather data has also increased. The collection of information from remote data gathering stations, by a central controlling unit allows both more convenient retrieval and more accurate data. The latter is due to the fact that the measurements will be entirely related to factors at the collection site [11].

III. RESULTS

The results section of this paper is organized as follows. Subsection A provides the design specifications of InterDAQ, the internet appliance. Subsection B discusses the design implementation. Subsection C presents the laboratory measurements.

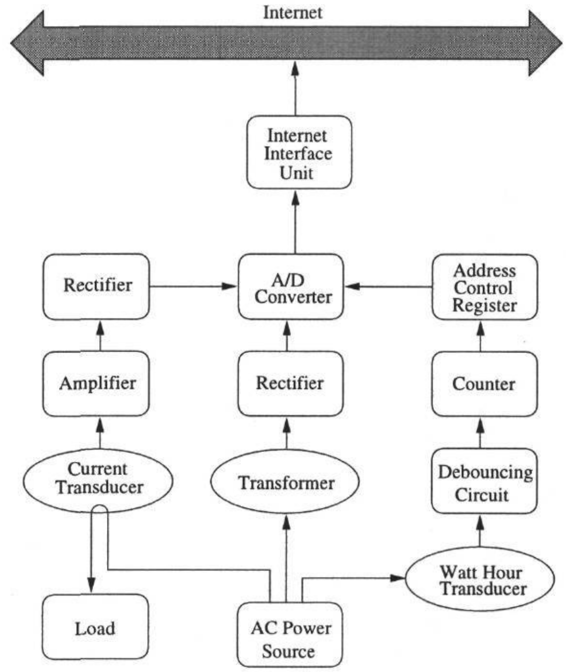

Fig. 2. System Block Diagram

A. Design Specifications

The main objectives of this project were as follows: to take voltage and current measurements and convert all this analog data into digital data, make temperature measurements, make Watt-hour transducer measurements and send this information to the Internet interface unit. Figure 2 details the main functions provided. InterDAQ performs the necessary tabulations to the acquired data, writes it into an ASCII file format, and then makes the data available on the Internet via a web browser (or through FTP). Below is a more detailed explanation of each task that the system will need to perform:

- Poll the measuring module and retrieve any available data.
- Take readings from both the current transducer and voltage transformer and convert analog data into digital data.
- Acquire digital data from a Watt-hour transducer.
- Store all digital data in RAM memory.
- Perform the necessary calculations and log into the Internet interface unit's FTP directory using the Java programs.
- Upload data as requested by the central server.

B. Implementation Details

The scope of the InterDAQ project ranged from raw measurements of voltage and current measurements up to Java programming and networking. In this section of the paper, we will step through each segment of the InterDAQ design, which is broken into five main areas: the current sensing circuit, the voltage sensing circuit, the precision rectifying circuit, the digital interface circuit, and the Java polling program. Power is defined as the vector dot product of voltage and current. For the experiment given in the next section of this paper, we employed a resistive load. Thus the InterDAQ computes the power drawn as the product of an RMS current measurement and an RMS voltage measurement.

Although any current transducer could be used, the testing was done with a 0-50 Amp current transducer (CT). The CT provides the starting point for the load's current measurement and converts an alternating current into an alternating voltage. The induced voltage had a root mean square (rms) value between 0 and 0.33 V. The 0-50 Amp CT is linear throughout its range, thus 0 A, 25 A, and 50 A currents produced 0 VRMS, 0.166 VRMS, and 0.333 VRMS outputs respectively. These voltages were passed through a gain 15 amplifier to increase the dynamic range (0 to 5 V) before A/D conversion.

To obtain a DC voltage proportional to the load's AC voltage, a 166J3 Hammond transformer was used in conjunction with a precision rectifier [12] and a filter capacitor. The transfer function is shown in Figure 3. The second order nature of the function is expected. The second order polynomial fit is remarkably close to the data points and shows a correlation very close to perfect (R=0.9998). This correlation compares very favorably with the correlation values found in [3].

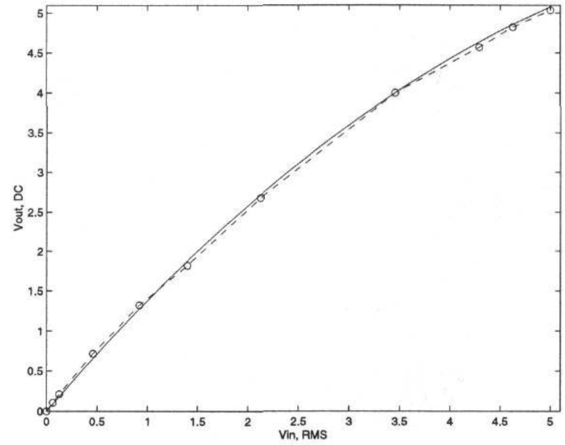

Fig. 3. Rectifier Transfer Function. The circles indicate the lab data and are connected by a dashed line, while the solid line indicates a second order polynomial fit to this data.

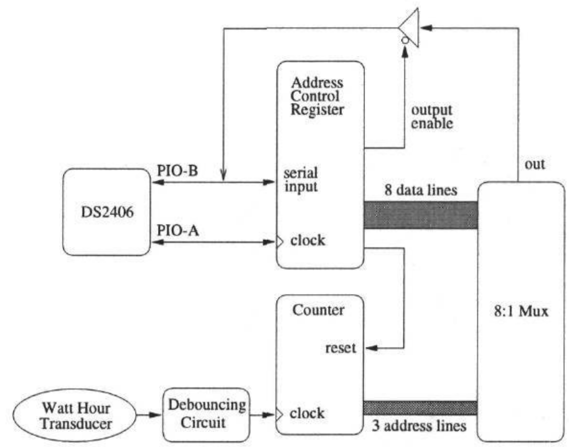

Fig. 4. Digital Circuit for Processing Watt-Hours

Figure 4 depicts the circuit used to process the Watt-hour measurement. The Watt-hour transducer produced one dry contact relay closure for every 100 Watt-hours of energy usage measured. This relay closure (200 milliseconds in duration) is debounced by a monostable flip-flop. This debounced pulse is used to clock an 8-bit counter. The value in the counter is the energy (in 100's of Watt-hours) used since the last time the counter was read. The rest of the circuit deals with reading this counter value.

The DS2406 is a Dallas Semiconductor dual-addressable switch. It has two channels, PIO-A and PIO-B. Each of these channels can sense the logic level of the line connected to it, as well as send a logic signal by pulling the line low. In the digital board, Channel A is used as a clock signal and Channel B is used to send data to the Address/Control Register and receive data from the multiplexer. Since channel A is a clock, the signals sent to it through the 1-Wire bus will always be alternating 0,1,0,1. The actual data to be written by channel B will remain constant for 2 clock cycles (4 bits). So a signal will be sent in two clock cycles (four 1's for a logical high and four 0's for a logical low). Each bit must be sent in 4's and will be received in pairs by the Address/Control Register. In normal operation (one full reading of the 8-bit counter), the Address/Control Register receives 16 bits of data from the DS2406 switch.

The first six bits contain the 3-bit address (A2A1A0) of the bit to be read. These are fed to the multiplexer to select one of the 8 bits from the counter. Bit 7, CR, is used as a counter reset signal. The Output Enable (OE) bits are used to enable or disable the tri-state buffer connected to the output. As shown on the schematic, the output of the multiplexer passes through the tri-state and then back into the I/O channels B on the 2406. The output enable signal keeps the buffer in High Impedance mode until the ACR is full. This prevents the output signal from interfering with the incoming data on channel B. When the ACR is full, the OE signal enables the buffer and the signal can pass through to channel B, where it is read by the 2406. The last bit, Shift Register Clear (SR CLR) is used to clear the ACR. At the beginning of the next cycle, the first bit of the control code is shifted in, forcing bit 14 into the last position. A high signal in this position clears the two shift registers and prepares them for the next incoming control code.

The Java program that runs on InterDAQ contains 5 classes. These are: Sensor, Control, LogFile, Temperature, and Convert. The flow chart for the basic Java polling program is shown below in Figure 5.

The Sensor class defines an object that can store all the relevant information about a sensor connected to the logger. This information includes: (1) raw sensor value, (2) units for the reading, (3) conversion factor for the reading, (4) time and date of the sample, (5) 64-bit 1-Wire ID of the sensor or device used to communicate with the sensor (this is stored as an array of 8 bytes, as well as the string representation of the ID), (6) description of the sensor, and (7) channel number (if the device has selectable channels, like an A/D converter). The class also contains accessor and mutator Java tasks for each of the fields listed above. In addition, there are two Java tasks that return a formatted string that can be used as a log file entry. This string can be formatted as plain text (outTXT) or with HTML tags (outHTML).

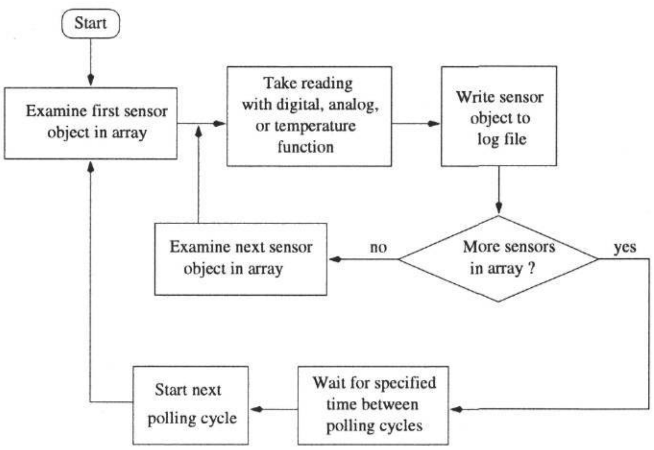

Fig. 5. Flow of Java Code

The Control class contains the main Java task and the central functionality of the program. First, it parses the command-line options passed to the program and directs the flow of the program according to these (which config file to use, one sample mode or normal continuous sampling, etc). It will then sample (either in a loop, or one time only) and cycle through an array of Sensor objects. For each object, it determines the type of sensor and calls the appropriate sampling Java task. The sample value is stored in the Sensor object that is then written to the log file. This process is done for each sensor in the list (which could be read from a configuration file). The LogFile class advances the file pointer to the end of the file before writing, so extra log entries are appended to the previous file (if one exists). This task also handles external access to the file. The Temperature class provides the method (A *method* is defined as a subroutine, whereas a *task* is defined as a function that passes both input and output variables in the Java programming language [13].) necessary to acquire the temperature data. The DS1820 1-Wire temperature probe connected to the logger.

The Convert class provides the method for retrieving analog readings from any of the 8 A/D converter channels on the T8AD. It also contains the read digital task for reading a digital value from any of the four DS2406 addressable switches on the board. This task requires that our special control and counter circuit be connected to both channels of a DS2406.

C. Laboratory Measurements

Figure 6 depicts the measurement information as seen on the Internet, while the InterDAQ is in operation. There were four main test plans for this project, which ensures that the current transducer, voltage transformer, Watt-hour transducer, and temperature sensor perform their specific tasks effectively.

In order to perform tests for the current transducer, we attached a resistive load (power factor = 1.0) to an 120V AC power source. This created a situation in which current was drawn at different amperage levels, namely: 0.833A, 1.66A, 2.5A, 3.33A, 4.166A and 5A. This enabled us to acquire the current readings and connect the output of the current transducer to our electronic circuit. The voltage transformer was connected directly to the AC power source in order to determine the voltage as it was applied to the load. The acquired voltage was then stepped down in order to allow measurements of the voltage level without overloading our measurement circuitry. The Watt-hour transducer was connected to the power supply and measurements of the power consumption are set at 100 KW/hr intervals. Each time the load consumed 100 Watt-hours of energy, it activates a pulse that InterDAQ captures and records.

In terms of sampling times, with the current Java code for data acquisition, one complete sample cycle includes: one temperature reading using a DS1820 1-Wire temperature probe; two A/D converter readings (voltage transformer and current transducer) using the T8AD board; and one

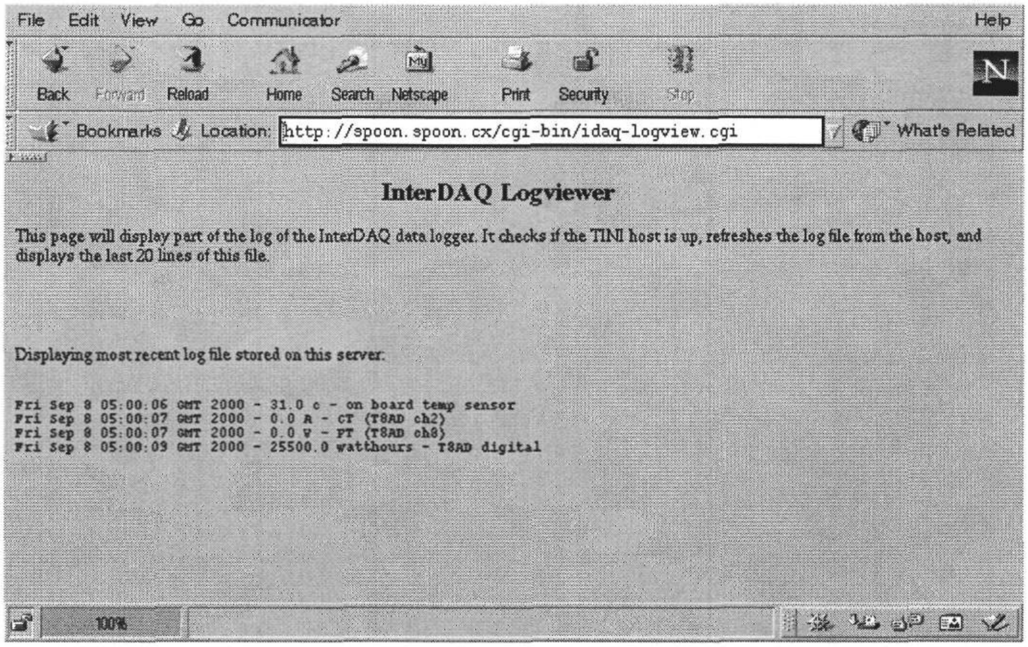

Fig. 6. Webpage Depicting one Cycle of Measurement Data

digital reading using the T8AD and extra circuit used to store counts from the Watt-hour transducer. The full cycle, including sampling and writing to a file on the Internet interface unit, takes an average of about 3 seconds. PolyConvert is a Java task that utilizes the Newton-Raphson algorithm for finding the roots to a polynomial, which was used to allow a second order polynomial fit to the measured data. The average durations for the different program functions are as follows: A/DC read 150 ms, Digital read 700 ms, Temperature 100 ms, File write 300 ms, PolyConvert 60 - 150 ms.

IV. CONCLUSION

We have described a new data-logging internet appliance, called InterDAQ, which provides a unique solution to the power management and logging problem in the age of Infomatics. The main contribution of this paper is that it discusses the construction of a device to accomplish the task of gathering data and making it available over the Internet. Specifically, our Internet appliance will allow for the capture of voltage, current, temperature and time stamp information, which is then formatted by our device with HTML tags to allow for convenient viewing on a website.

References

[1] J. Haberl, S. Thamilseran, T. Reddy, D. Claridge, D. O'Neal, W. Turner, "Baseline calculations for measurement and verification of energy and demand savings in a revolving loan program in Texas" *ASHRAE Transactions*, vol 104 ,Pt. 2, 1998.

[2] D. Claridge, J. Haberl, R. Sparks, R. Lopez, K. Kissok, "Monitored commercial buildings energy data: reporting the results" *ASHRAE Transactions: Symposia*, pp. 881-889, 1992.

[3] S. Li, X. Zhou, K. Morris, "Measurement of snow and sea ice surface temperature and emissivity in the Ross Sea," *Proc. of the International Geoscience and Remote Sensing Symposium*, vol 2, pp. 1034-1036, 1999.

[4] A. Bousbaine, "Development of a digital data acquisition system for localised power loss measurement in electrical machines," *Proc. of the Ninth International Conference on Electrical Machines and Drives*, no. 468, pp. 194-198, 1999.

[5] D. Douglass, "Real-time monitoring and dynamic thermal rating of power transmission circuits," *IEEE Transactions on Power Delivery*, vol. 11, no. 3, pp. 1407-1418, 1996.

[6] A. Belardinelii, G. Palagi, R. Bedini, A. Ripoli, V. Macellari, D. Franchi, "Advanced technology for personal biomedical signal logging and monitoring," *Proc. of the IEEE International Conference on Engineering in Medicine and Biology Society*, vol. 3, pp. 1295-1298, 1998.

[7] B. Way, D. Hosom, J. Ware, R. Trask, G. Allsup, "Vector Measuring Current Meter (VMCM) upgrade," *Proc. of MTS/IEEE*, vol. 1, pp. 120-124, 1996.

[8] D. Tseles, D. Piromalis, J. Melidis, "RISC technology microcontroller based smart measurement and control device," *Proc. of MELECON*, vol. 2, pp. 1172-1175, 1996.

[9] D. Bechrakis and P. Sparis, "A flexible data logging device for wind potential measurements and statistical magnitudes," *Proc. of the 6th IEEE International Conference on Electronics, Circuits and Systems* , vol 3, pp. 1483-1486, 1999.

[10] Z. Swider, L. Trybus, M. Sniezek, "Two-channel temperature controller for buildings," *Proc. of IEEE Conference on Control Applications*, vol. 1, pp. 21-26, 1994.

[11] M. Bertocco, F. Ferraris, C. Offelli, M. Parvis,, "A client-server architecture for distributed measurement systems," *IEEE Transactions on Instrumentation and Measurement*, vol. 47, no. 5, pp. 1143-1148, 1998.

[12] A. Sedra and K. Smith, *Microelectronic Circuits.* 4th Edition. New York: Oxford University Press, 1998.

[13] D. Comer, *Internetworking with TCP/IP - Principles, Protocols, and Architectures.* 4th Edition. Upper Saddle River, New Jersey: Prentice-Hall, 2000.

TECHNOLOGY ENABLERS FOR NEXT-GENERATION ECONOMIC BUILDING MONITORING SYSTEMS

JAMES SWEENEY JR.
SENIOR RESEARCH ASSOCIATE

CHARLES H. CULP, P.E., PH.D.
ASSOCIATE DIRECTOR

ENERGY SYSTEMS LAB
TEXAS A&M UNIVERSITY
COLLEGE STATION, TEXAS

ABSTRACT

The measurement of a building's electrical and thermal consumption provides the necessary data used to increase energy efficiency. Measurements typically range from hourly to monthly. Monthly data can be used to determine if savings are being maintained. Hourly data provides added detail for diagnostics. Currently, the cost of a complete monitoring system deters use in buildings under 30,000 to 50,000 square feet. Buildings can be optimized using techniques like Continuous Commissioning[SM] (CC[SM]) and experience a reduction in consumption ranging from 10% to 40% [1]. Using hourly data has proven to be very effective in maintaining the initial level of savings over an extended period of time [1]. The Energy Systems Laboratory at Texas A&M University (TAMU) has applied CC[SM] to over 100 buildings and obtained an average savings of 22 percent. Currently, whole-building and sub-metering relies on conventional dial-up based data logging systems. Development of a next-generation data acquisition system is essential to achieve a lower cost for building energy monitoring and analysis. The next-generation system discussed in this paper is a complete redesign. It will be Internet-enabled and secure; take advantage of current advances in smarter sensors, use embedded micro-controllers and mixed signal processors and use Java and XML.

INTRODUCTION

On-going metering and monitoring is an essential part of the CC[SM] process [1]. Monitoring a building's energy and environmental conditions generally requires the measurement of the building's energy usage and demand, thermal consumption, lighting, temperature, relative humidity and sensitive gases such as carbon dioxide. Hourly intervals are recommended for monitoring, at a minimum.

The material and installation costs of current implementations may run from $5,000 to $20,000 for an installed system. Electronic interfacing and configuration of these systems is also problematic and error prone [2]. Current systems utilize dialup technology, often yielding weekly data with associated long distance charges.

This article discusses building monitoring components and protocols and technologies that will increase the measurement system's integration, control and maintenance.

SYSTEM COMPONENTS

The building monitoring system components may be divided into four subsystems: 1) sensor/transducer, 2) data collection and control, 3) local building server, and 4) master building server. In practice these subsystems are quite distinct, distributed and from multiple vendors. Table 1 lists the subsystems, their functions, their components, inputs and outputs.

Sensors:

Physical variables like electrical current, electrical power, and temperature change the output signal of sensors. Sensor performance must be specified. Generally, items like signal output, environmental ranges of operation, accuracy, linearity and repeatability must be carefully specified. Sensor outputs can be voltage or current signals, which are typically associated with "dumb" or analog sensors. "Smart" sensors usually output digital values in which the sensor electronics may perform calibrations and scaling functions internally. Other sensors like totalizers, can output pulses, which need to be counted and totalized. This next generation system should be designed to accept a range of sensors.

Data Collection and Control

This component is usually called the "logger". The logger controls collection of sensor data and communications. Next generation devices will allow increased data

Subsystem	*Function(s)*	*Components*	*Input*	*Ouputs*
Sensor/Transducer	Convert a physical variable to an electrical signal and possibly a digital value	Sensors of various types and "intelligence"	Monitored physical parameter	Analog and digital signals
Data Collection and Control	Collect data and instrument status, performs local configuration control and communicates externally	Data loggers	Sensor signal(s)	Data communications and control signals
Local Building Server	Aggregate and analyze local building data, upload data to master building server, pass or generate control information to the logger	Small to mid-computer system with a relational database system (RDBMS) and application specific software	Data streams from data collection devices Configuration data from other servers	Database upload to master building system Control data to the logger
Master Building Server	Aggregate building data, manage measurement devices (loggers), analyze data for quality, savings, and unknowns Control network of local servers and loggers	Large computer system with RDBMS and online analytical processing (OLAP) systems	Data streams from data collection devices and/or database uploads	Control signals and reporting

Table 1. Building Monitoring System Components.

checking, validation and analysis to be done at the point of collection.

Local Building Server

The local building server aggregates and analyzes data streams from the data collection and control subsystem for direct action or pre-processing for the master building server. This server also generates or passes control data to the local logger. Local building servers may be installed to handle certain sections of a large building or integrate the whole building. In some cases the local building server may be built into the logger.

Master Building Server

This system primarily handles data from local building servers. Local building servers may be aggregated in geographic or application specific context for analysis. Primary duties include aggregating building data, managing measurement devices (loggers), and analyzing data for quality, diagnostics, and savings. Modern EMCS may encompass all of the subsystems. Simple monitoring systems may only include sensors, and data collection and control subsystems.

In order for these subsystems to interoperate, they must be networked with standard communication protocols. The Internet provides an open communication protocol, Transmission Control Protocol/Internet Protocol (TCP/IP).

COMMUNICATIONS AND INTERFACES

The Internet, supported by TCP/IP, has become the standard for computer-to-computer communications. The Open Systems Interconnect Reference Model (OSI) describes TCP/IP. OSI contains seven layers that define the functions of data communications protocols [3]. They are the physical, data link, network, transport, session, presentation and application layers.

TCP/IP contains four layers, three of which are from the OSI model, plus an additional layer. OSI application, transport, and network layers describe TCP/IP. An additional layer, the Internet layer, is also defined (Table 2). This layer is above the network layer and is the heart off IP [3]. IP is responsible for packet delivery. TCP resides in the transport layer and is responsible for reliable data delivery service with end-to-end error detection and correction [3]. Secure networking of data and control is a principal challenge in distributed building monitoring. TCP/IP has emerged as the standard for Internet enabled data loggers and transducers. TCP/IP is a more reliable and faster communication vehicle compared to traditional dialup communications.

Stack Layer	*Description*
4. Application Layer	Application layer and processes use the transport layer protocols to deliver data
3. Transport Layer	Provides data delivery in the form of the transmission control protocol and the user datagram protocol
2. Internet Layer	Handles routing and transmission of data to the transport layer
1. Network Layer	Routines for accessing physical networks

Table 2. TCP/IP Stack.

TCP/IP and Security

TCP/IP protocol is a packet-switched, digital communications technology designed to reliably transport data files among various computers around the world. Building monitoring and control data is generally security sensitive and the Internet is insecure. Establishing a Virtual Private Network (VPN) for a building monitoring system is a means to secure the information. A VPN provides a secure point-to-point tunnel through the public Internet. These tunnels are created and destroyed on demand. The "tunneling" of data is the repackaging of data from one network to another [4].

The repackaging of data occurs at OSI layers two through five. Link-level (data-link and network) encryption prevents network traffic analysis and attacks. Protocol-level (network and transport) encryption encrypts only data, facilitating traffic analysis. IPSecure (IPSec) is a secure network protocol that acts at the network, protecting and authenticating IP packets between participating IPSec devices, such as routers [5].

The data packets may also be encrypted with sophisticated algorithms in the application layer (seven). Secure Sockets Layer (SSL) protocol is a means to provide privacy and reliability between two communicating applications. SSL operates in the application layer of the OSI model. This protocol enables a server and a client to authenticate each other and to negotiate an encryption algorithm and cryptographic keys before the application protocol transmits or receives its first byte of data. [6]. This allows another application to sit on top of SSL. Communications between data collection subsystems and building servers may be secured this way. VPNs also include bandwidth management. Packets may be tagged with priority and time-sensitivity information, allowing traffic to be routed based on its delivery priority [4]. Priority and time-sensitivity packet information is crucial for the building monitoring system to expeditiously transmit and receive data through the building's local network. Standard and open protocols like TCP/IP, IPSec and SSL provide base communication technologies to build the monitoring system on.

Smart Sensors Networking

A "smart" sensor is a sensor with some processing of the physical value sensed and usually has the processed value digitized. A "smart" sensor interface standard has recently become a reality with the emergence of the Institute of Electronics and Electrical Engineers (IEEE) 1451.

IEEE 1451 defines an interface for the connection of sensors and transducers to microprocessors, control and field networks, and data acquisition and instrumentation systems that are network independent [7]. Although the building engineer may not be interested in knowing the details of the protocol standards, it is important to realize that sensors complying with this standard are interchangeable yielding simpler installations and maintenance.

IEEE 1451 Details

The standard is divided into four main parts:

1) transducer to microprocessor communication interface (IEEE 1451.2),
2) networked smart transducer model (IEEE 1451.1),
3) multi-drop distributed system for interfacing smart transducers (IEEE 1451.3 -proposed), and
4) mixed-mode transducer interface (IEEE 1451.4 -proposed).

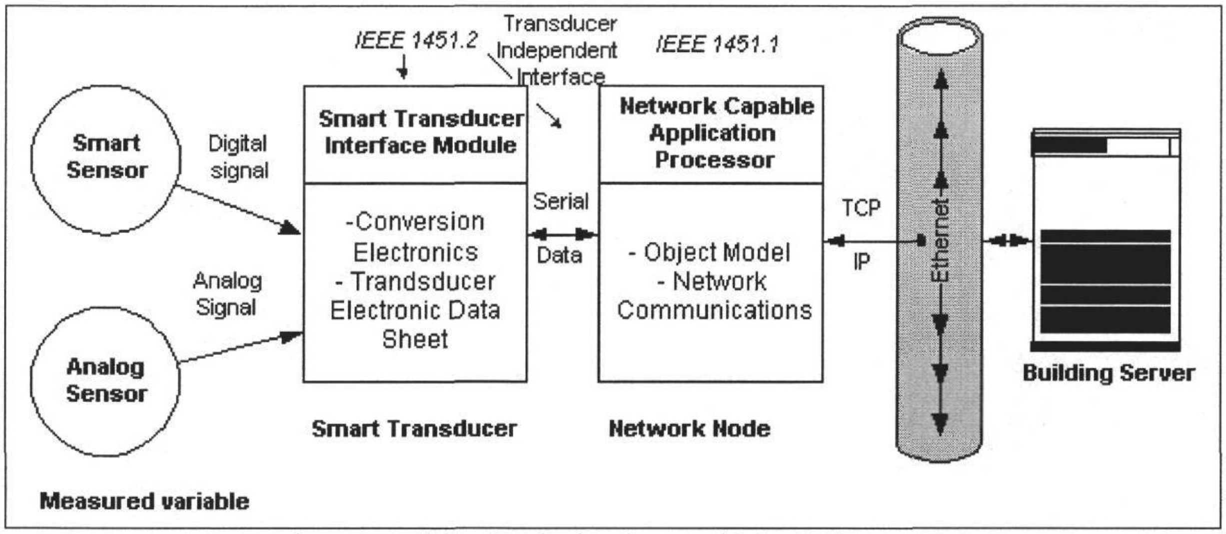

Figure 1. Building Monitoring System with IEEE 1451.

IEEE 1451.1 defines the Transducer Electronic Datasheet (TEDS) and the Smart Transducer Interface Module (STIM). Every transducer that has a TEDS in non-volatile memory contains the type, attributes, operation and calibration of the transducer. This table is stored in the STIM. The mandatory data is 178 bytes [7].

Only two of the eight TEDS fields are required and must remain with the STIM for its lifetime. The remaining six fields are optional [8]. This capability will allow built in diagnostics, which will reduce install costs and on-going maintenance.

The STIM is a networked and intelligent transducer node that supports up to 255 sensors or actuators of various signal mixes. The STIM is connected to a network node called Network Capable Application Processor (NCAP), shown in Figure 2. The STIM transparently communicates with the network via the Transducer Independent Interface (TII), which links the STIM to the NCAP [8].

<u>IEEE 1451 Relevance</u>

The network smart transducer model defines a common object model for encapsulating the transducers interoperability at the application layer. IEEE 1451 provides methods that support network-neutral communication with both publisher-subscribe and client-server mechanisms [9]. This facilitates building monitoring data streams to the NCAP where the data may be stored, pre-processed, and/or pushed-pulled, depending on the application requirements (Figure 1).

Open standards in networking from the Internet to the sensor network interface are enabling secure, distributed building monitoring. The data collection and control subsystem may contain a NCAP, enabling distributed monitoring and control of IEEE 1451.2 enabled sensors and actuators (Figure 1). Secure data transmission from the data collection and control subsystem is provided by IPSec and/or SSL over the building local Ethernet. VPNs may facilitate secure data transmission from the data collection and control subsystem to the master building server.

EMBEDDED SYSTEMS & INTERNET APPLIANCES

Low cost embedded designs, systems on a chip (SoC), and Internet appliances (IA) are paving the way to low-cost hardware implementations that includes the logger and local server functionality. Embedded devices are found in consumer electronics, computer peripherals, and control systems in automobiles, aircraft and other industrial applications. Their operating systems and application programs are combined on the same device. Eight, sixteen and even thirty-two bit microcomputers are found in embedded applications. High performance microprocessors have become today's embedded controllers [10].

Hardware is divided into subsystems that are built around the micro-controller. The integration of volatile and non-volatile memories into microprocessors has permitted local storage of measured parameters without significantly increasing cost [11].

The integration of microprocessors, communication subsystems and volatile memories is widespread. New classes of embedded devices are evolving. SoCs are emerging as effective application designs, which are more complex than the traditional application specific integrated circuits of the past. Analog Devices' ADuC812 data converter is one such device. This device is a general-purpose data acquisition system on a single chip. It contains an eight-channel analog to digital and eight-channel digital to analog subsystem, serial communication facilities and an 8051 compatible micro-controller with volatile and non-volatile memories. This device has been demonstrated in the IEEE 1451 context [8]. In this application, the ADuC812 was implemented as a STIM. If the ADuC812 is embedded with a network adapter and stack, the STIM and NCAP could feasibly be combined into one system. This will make feasible an open, network-accessible data collection system.

SoC and high power microcomputers are also enabling the development of low cost single board computers (SBC). These systems are complete X86 compatible machines that have graphic, serial, parallel, and network subsystems.

SoC may reduce overall system cost because fewer components are necessary to get the same functionality. Benefits of a single chip system are not only power, weight and size but the price may be lower in high volume [10]. By combining the SBC with small footprint embedded operating systems (Linux, WinCE, BeIA), micro-relational databases and a Java Virtual Machine, complex applications may be run at the embedded level. SoCs are at the core of a new class of Internet enabled devices called Internet Appliances (IA). Commercial integrators are offering SoCs targeted at the IA market combining industry standard microprocessors, embedded programmable logic and memory [4].

SYSTEM PROGRAMMING

Building applications must have a strong framework that facilitates reusable components, operating system independence, and network operation. Sun Microsystems's programming paradigm, Java, accomplishes these. Applications must be developed fast to increase time to market. Sophisticated rapid application development tools such as an integrated development environment are available today for Java.

JAVA and Object-Oriented Programming

The Java programming language is an object-oriented that may be interpreted or compiled. Quite often programs written in Java are converted into byte codes, which are the portable machine language of the Java Virtual Machine (JVM) and interpreted on demand.

Portability.

Java was designed to be network oriented. In buildings, monitoring networks are composed of several systems with a variety of CPU and operating system architectures. A Java application can execute anywhere on the network. The Java compiler generates an architecture-independent object file format called byte code that can run on many processors. The machine specifics are handled by the architecture specific JVM. Java has been ported to various embedded systems from such low level devices as Dallas Semiconductor's Tiny Internet Interface to X86-based IA design such as National Semiconductor's Geode.

Distributed and Secure.

Java was built from the ground up with security in mind. Any code may be run with restricted permissions that prevent it from doing harm on the host system [12]. Defense occurs in access control of the memory space. For restricted programs, the JVM does not allow direct access to memory of the supporting system. The JVM also goes through a process known as bytecode verification whenever it loads an untrusted class [12]. This ensures that the JVM is protected from crashes and attacks. Java provides classes and interfaces for authentication and cryptography. These pieces of security allow Java to verify that any data is authentic and not modified in transit.

Java provides dynamic networking capability over multiple OS platforms. Java's design is powerful because it facilitates the clean definition of interfaces and makes it possible to provide reusable software, as Java objects are portable.

Java is a distributed application-programming paradigm that facilitates network oriented application communications across the building monitoring system. Data exchanged between these subsystems must be standardized and open.

<u>XML</u>

The World Wide Web Consortium's eXtensible Markup Language is changing the landscape of inter-application and systems data interchange by offering a framework of "self-describing" data. The self-describing nature of XML provides a common data format for data context.

XML is a text-based data description language that facilitates interoperability using an open standard. This markup language for data simplifies communications between applications and platforms. As HTML has enabled the display of information on the web, XML enables data interchange between systems. The structure and specifics of data are not lost in XML encoded data as with application specific and proprietary formats.

XML may be used at all levels of communication across the building monitoring system. Low-level communication from the data collection and control subsystems may encode data streams in XML to simplify upstream processing.

Figure 2 is an example of a Site Energy Report in a simple XML document. XML documents are built with various types of markup. Elements are the most common form of markup [13]. Elements are delimited by angle brackets pairs (<>, </>). In this example, the *</site-energy-report>* element contains many sub-elements. All elements within the *</site-energy-report>* support the same syntax whereby the data in between the angle brackets pairs is the content. The site-energy-measurement element contains attributes that are assigned specific values. The "entry" sub-element of the site-energy-measurement element has attributes: timestamp, position, description, units and value. Each of these attributes is assigned a numeric or character value.

This text file may be parsed and processed by another application. An example process would be to parse the site-energy-report and upload the data to a database or report generating system.

```
ENERGY XML EXAMPLE:
<site-energy-report>
<date>2001-04-10</date>
<time>08:00</time>
<site-information>
        <name-1>Evan's Library</name-1>
        <name-2>Texas A University</name-2>
        <site-number>492</site-number>
        <location>
           <address></address>
           <city>College Station</city>
           <state>Texas</state>
           <country>USA</country>
        </location>
        <use>Library facility</use>
        <area>square footage of the
        building</area>
        <description>This building was built in
        X. It has point of interest A,
        etc</description>
</site-information>
<site-energy-measurement>
        <entry timestamp="2001-04-02 00:00"
        position="1" description="Whole
        Building Electric" units="kWh"
        value="1200.01" />
        <entry timestamp="2001-04-02 00:00"
        position="2" description="Hot Water
        Flow Meter" units="Gal/h"
        value="2546.21" />
</site-energy-measurement>
</site-energy-report>
```

Figure 2. Energy XML Example.

SUMMARY

Next-generation networked monitoring systems will revolutionize future building monitoring systems in terms of better integration, control, and maintenance. Data collection and control subsystems may be networked through the Internet, enabling an open and powerful distributed monitoring system. Building monitoring systems developers may use the IEEE 1451 protocol with secure Internet communications to build an open, object-based system.

The technology discussed herein represents methods to improve monitoring in buildings. Today, sequential data files are obtained using dialup technology. Secure Internet technology allows flexible secured access to data without dialup costs and other limitations. Java and XML software technology provides a platform independent rapid and extensible development environment for building monitoring systems. XML technology facilitates extensible data interchange and usability by employing data descriptions. These descriptions provide context for multiple applications to share common data across the Internet.

REFERNCES

1. Kats, G.H., et al, *Energy Efficiency as a Commodity: The Emergence of an Efficiency Secondary Market for Savings in Commercial Buildings*, U.S. Department of Energy Protocol, April 1996.

2. Orihara, A., et al, An Autonomous "Decentralized System Platform under Multi-vendor Environments in Building Automation", *IEEE Proceedings*. ISADS. pp. 409-415, 1997.

3. Hunt, C., *TCP/IP Network Administration*, 2^{rd}. Ed., O'Reilly and Associates, Inc. California, 1998.

4. Becker, S., "Unlocking the gate to mobile Web Access", *Portable Design*, pp.33– 48, April 2001.

5. Internet Engineering Task Force, March 2001, *IEFT IPSec Homepage,* Available: http://www.ietf.org/html.charters/ipsec-charter.html, (Read 04-15-2001).

6. Freier, A.O., March 1996. *The SSL Protocol Version 3.0*, Available: http://home.netscape.com/eng/ssl3/ssl-toc.html. (Read 4/10/2001).

7. Lee, K., "IEEE 1451: A Standard in Support of Smart Transducer Networking. IEEE Instrument and Measurement Technology Conference", *2000 IMTC IEEE Proceedings*, Vol. 2. pp. 525-528.

8. O'Mara,B., Conway,P., "Designing a IEEE 1451.2–Compliant Transducer", *Senors*. Vol 17 No. 8, August 2000.

9. Manders, E. J., Barford. L.A., "Diagnosis of a Continuous Dynamic System from Distributed Measurements", *2000 IEEE Proceedings*.

10. Karakehayov, Z.,Christensen, K.S., Winther, O., "*Embedded Systems Design with 8051 Microcontrollers – Hardware and Software*", Marcel Dekker, Inc. USA, 1999.

11. Nagaraju,M., Kumar,T.P. "*Networked Electronic Emergy Meters with Power Quality Analysis*", Signion Systems Ltd., Hyderabad, India, pp 45-57

12. Flanagan, D., *Java in a Nutshell*, 3^{rd}. Ed., O'Reilly and Associates, Inc. California, 1999.

13. Walsh, N., October 1998, *What is XML? at Xml.com Homepage*, Available: http://www.xml.com/pub/a/98/10/guide2.html , (Read 4/16/2001).