Remote Monitoring and Controlling of a Material Science Experiment

Wattanapong KURDTHONGMEE

School of Engineering and Resources,
Walailak University, Thasala, Nakorn Si Thammarat 80160, Thailand.

ABSTRACT

The computer industry's remarkable ability to integrate more transistors into a small area of silicon is increasing the intelligence of our devices and simultaneously decreasing their cost and power consumption. In addition, the proliferation of wired and wireless networking spurred by the development of the world-wide web and demands for mobile access are enabling low-cost connectivity among computing devices. It is now possible to connect every computing device into a true world-wide web that connects the physical world of sensors and actuators to the virtual world of our information utilities and services. This paper examines an application of an integration of the intelligent chip with the network connectivity into a material science experiment designed to study the sorption of woods. The intelligence and network connectivity infrastructures of the system eliminate laborious tasks previously required during experiment control and data collection processes.

Key words: Sorption of woods – Network - Ready microcontrollers

INTRODUCTION

Many supporting technologies are currently available for constructing a network infrastructure that pervades everyday life on a massive scale (1). Furthermore, over the last 10 years, many of our work practices have been applied to the electronic media. Even consumer products such as ovens, toasters, and dishwashers are now automated through embedded computation. In fact, 98% of the computing devices sold today are embedded in products whose use does not require the user to be aware of their presence. A new revolution is about to happen, one in which we will gain immense additional value by connecting all these computational components together. This opportunity presents us with some important challenges for building useful services, designing more robust and easily manageable systems, and guaranteeing user privacy and security.

The internet has been the most vital factor in the enhancement of this opportunity. Historically, the implementation of TCP/IP protocol in the mid-1980s required the computational resources typically found on a desktop or work station machine. The embeddable microprocessors were considerably less powerful, and the ability to support full protocol stacks at a reasonable performance level was beyond the capability of these under-powered computational engines. Internet connectivity was thus limited and costly. Fifteen years later, a microcontroller that cost just a few dollars, in combination with one megabyte of memory, was just as capable as a

desktop computer had been in 1985. Such a device can support a compact embedded operating system, interface to a 10 Mbps network, and run a TCP/IP stack and a web-server interfacing with the now standardized ubiquitous HTTP protocol. In this paper we use the term "*internet-ready microcontroller*" to refer to this type of device.

What benefit do we derive from embedding a web-server into an appliance? The basic functionality of the web enables client programs or browsers to fetch web pages and display them on a browser window. Hyperlinks within a file can provide further reference to other files, either close to or remote from that site. More importantly, in the case of an appliance, a link may also provide a reference to a Common Gateway Interface (CGI) script. This script executes and returns HTML to the browser. Because HTML can be generated dynamically by the script, it may incorporate real-time data that are being derived from sensors (2,1). Thus any appliance connected in this way can be monitored through a CGI script and the results presented to a user in a convenient graphical form. Similar mechanisms can also be used to directly control an appliance from a remote browser.

In this paper, we examine and discuss a new application of the internet-ready microcontroller. Section 2 of the paper contains a description of a material science experiment which incorporated the use of the microcontroller. Details of the system, both hardware and software components, are then described in Section 3. The conduct of the experiment, experimental results, and discussion are presented in Section 4.

MATERIALS AND METHODS

An Experiment to Measure Sorption of Woods

Wood and wood-based materials are among the most important building materials in Thailand. Wood is an organic material with a very complex composition. Due to its complex composition and its many-sided usage there still are many unanswered questions concerning wood in use in buildings. In buildings, wood is subjected to shrinkage, swelling, mould growth, and rot if it is exposed to unfavorable environmental conditions. These phenomenon are all related to moisture content and moisture conditions in a building structure. Wood and wooden materials in buildings are continuously exposed to smaller or larger changes in climate throughout the night and day and throughout the year. There is still a need for improving and developing the basic knowledge about moisture in wood subjected to changing climatic condition. The aim of our experiment is to investigate the moisture sorption of wood when it is exposed to moisture in a variety of controlled conditions. Due to the limitation on the length of this paper, we describe in detail only one type of experiment in which both temperature and humidity are constant.

In the experiment (3), a specimen of wood is directly exposed to a source of moisture by putting it into water or salt solution for a constant period of time and at constant room temperature and humidity. The volume of absorbed water is then measured by weighing the specimen immediately after it has been removed from the solution. In order to get the correct result, this experimental procedure needs to be carried out continuously several times over a period of time lasting from 1 day to 1 week or until the saturation point is reached. The experimental result is normally presented in the form of graph showing the relationship between the weight of water or solution within the specimen under test and the time taken.

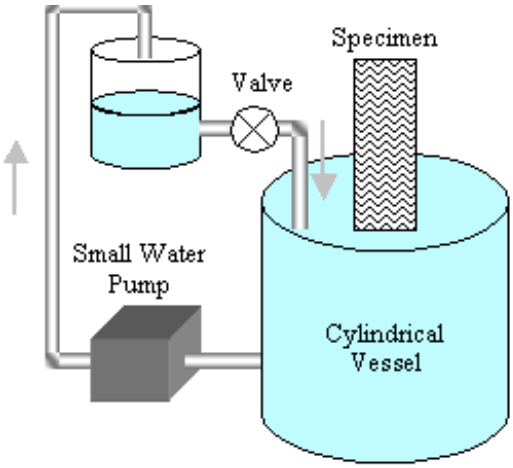

Figure 1. Sketch of the sorption apparatus

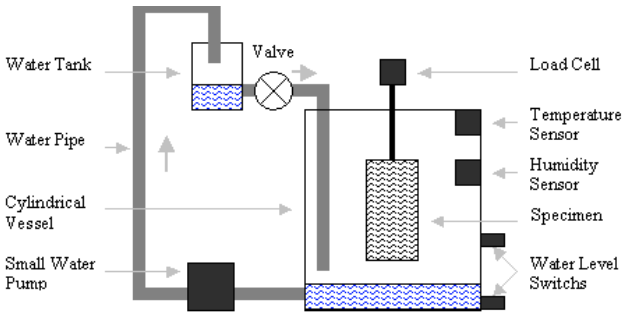

Figure 2. Cross section of the sorption apparatus

The apparatus, depicted in **Figure 1**, consists of a cylindrical vessel containing water or salt solution which is placed within a container with controlled temperature and humidity. No mechanical part is used to vertically move the specimen out of the moisture source. Instead, we employ a less complex system consisting of a small water pump and a water tank to empty the water within the vessel before the specimen is weighed. The water is stored in the water tank temporarily during the measuring process and is allowed to flow back into the vessel afterward through the control valve. The water level within the vessel is controlled by water level switches (see **Figure 2** for more detail). By using this apparatus, it is possible and easy to fix a load cell to the specimen.

System Architectural Details

It is arguable that only a simple and inexpensive stand-alone embedded

system is sufficient for automatic performance of the experimental procedures and control of the experiment, if the buffer of the embedded system is always available and the experimental data are uploaded before the buffer is full. However, if these conditions are not met some invaluable experimental result may be lost. The limitation of the stand-alone embedded system can be fully overcome by the utilization of the internet-ready microcontroller if the working environment is network-ready as in almost all university-level research laboratories.

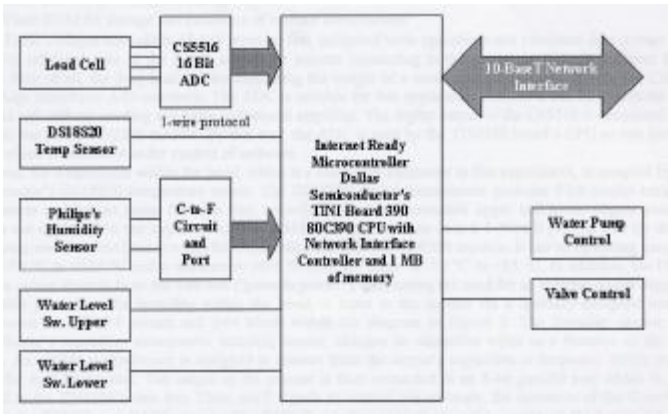

Figure 3. The hardware architecture of the Internet-Ready Microcontroller-based System for studying the sorption

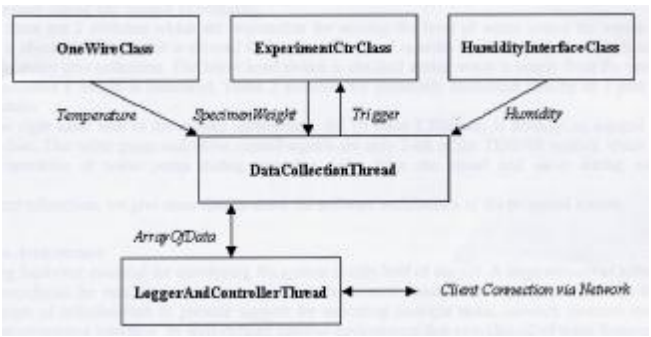

Figure 4. The Software Architecture of the Internet-Ready Microcontroller-based System for studying the sorption of woods

Hardware Architecture

The hardware architecture of the internet-ready microcontroller-based system for studying the sorption of woods is depicted in **Figure 3.** The heart of the system is a Dallas Semiconductor's TINI390 module (model DS-TINI-1). The module is a compact (31.8 mm x 102.9 mm) 72-pin SIMM board. It is an Ethernet-ready hardware

implementation and supports all of the following features:

- Hosts the TINI runtime environment in a validated hardware design
- 10 Base-T Ethernet for networking
- Extensive I/O capabilities of the DS80C390 microcontroller exposed through Java™ APIs
- Dual serial ports
- Dual 1-Wire® net interfaces
- Dual CAN (Controller Area Network) controllers
- 2-wire synchronous serial bus
- General-purpose digital I/O
- Easy system expansion using parallel bus interface
- Real-time clock for time stamping
- Flash ROM for storage and execution of runtime environment
- 512K/1 Mbyte nonvolatile SRAM provides fast, unlimited write operations and persistent data storage

On the left-hand side of the figure are all the sensors connecting to the processor through different kinds of interface. First, the 5-kg load cell for measuring the weight of a wood specimen is connected to the CS5516, a 16-bit bridge transducer A/D converter. The ADC is suitable for this application because it can be interfaced directly to the load cell without needing any extra operational amplifier. The digital output of the CS5516 is connected directly to the data bus of the TINI390 module. By this means, the ADC is seen by the TINI390 board's CPU as one location of memory which is selectable under the control of software.

Second, the temperature within the container, which is a controlled parameter in this experiment, is sampled by Dallas Semiconductor's DS18S20 temperature sensor. The DS18S20 digital thermometer provides 9-bit celcius temperature measurements and has an alarm function with nonvolatile user-programmable upper and lower trigger points (this feature is not exploited in our application). The DS18S20 communicates over a 1-Wire® (4) bus that by definition requires only one data line (and ground) for communication with the TINI390 module. It has an operating temperature range of -55 °C to +125 °C and is accurate to ±0.5 °C over the range of -10 °C to +85 °C. In addition, the DS18S20 can derive power directly from the data line ("parasite power"), eliminating the need for an external power supply.

Another parameter, the humidity within the container, is input to the system via a specially-designed interfacing circuit shown as a C-to-F circuit and port block within the diagram in **Figure 3.** The humidity sensor, Philips Semiconductor's capacitive atmospheric humidity sensor, changes its capacitive value as a function of the relative humidity. An astable multivibrator is designed to convert the sensor's capacitive value into frequency which is used as an input for an 8-bit counter. The output of the counter is then connected to an 8-bit parallel port which is, in turn, interfaced to the TINI390's data bus. There are 2 signals from the operation of the C-to-F circuit to control via software: the RESET and GATE signals. The RESET signal is used to clear the counter at the beginning of each sample of humidity. The GATE signal of constant activation time is used to enable the counter to count its input clock from the oscillator during any sample of humidity.

Finally, there are 2 switches which are responsible for sensing the level of

water within the vessel. The upper level switch is checked when the water is allowed to flow into the vessel in order to ensure that the water is always at the same level for every data collection. The lower level switch is checked when the water flows out from the vessel just before the specimen's weight is measured. These 2 switches are physically connected directly to 2 port pins of the TINI390 module.

From the right hand side of the system architecture, the 10 Base-T Ethernet is actually an integral part of the TINI390 module. The water pump and valve control signals are only 2-bit of the TINI390 module which are used to control the operation of water pump for emptying water from the vessel and of the valve during water filling respectively.

Software Architecture

Providing hardware essential for developing the system is only half of the task. A large amount of software is also required to coordinate the execution of every task in order to correctly control the experiment. Fortunately, the TINI390 has many layers of infrastructure to provide support for executing multiple tasks, network protocol stacks, and an application programming interface. Its well-defined runtime environment that provides all of these features allows us, as a system developer, to focus primarily on the details of the application.

The software that comprises TINI390 's runtime environment can be divided into two categories: native code executed directly by the microcontroller and an API interpreted as bytecodes by the Java™ Virtual Machine. Application code is written in Java and utilizes the API to exploit the capabilities of the native runtime and the underlying hardware resources.

With regard to the software architecture of the proposed system which is depicted in **Figure 4** the heart of the system is *LoggerAndContollerThread*-class. This class is the server class which spends eternity in the run-method responsible for processing network connections. It starts by creating a *ServerSocket*-object to listen for inbound connections from remote clients. The inbound connection is mostly in the form of a request for the experimental data to be transferred to the remote client.

The *LoggerAndContollerThread*-class, once it has been released, creates another thread, a *DataCollectionThread* inherited from Java's *Thread*-class, which is responsible for periodic sampling of the experimental data. The thread calls *OneWireClass* in order to obtain a current temperature. It also activates *ExperimentCtrlClass* to start emptying the water within the vessel, measuring the specimen's weight and re-filling the vessel. The weight of the specimen is returned to the caller class. **Figure 5** shows the statechart representing the operation of the *ExperimentCtrlClass*.

On the statechart in **Figure 5** the operation of the *ExperimentCtrlClass* starts at the "Water Pump" state which means that the water pump is on in order to empty the vessel while "LowerLevel" signal, from the water level switch lower, is false. After the "LowerLevel" signal is true, the state is changed to "Load Cell" state which continues until the EndOfConversion signal is false signalling the end of data conversion time of the ADC. The next and final state of the class is "Valve" state. In this state, the class keeps on checking "HigherLevel" signal which comes from the water level switch upper.

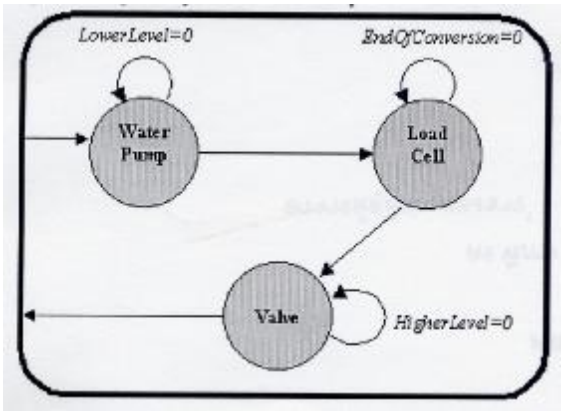

Figure 5. Statechart of the *ExperimentCtrlClass*-class

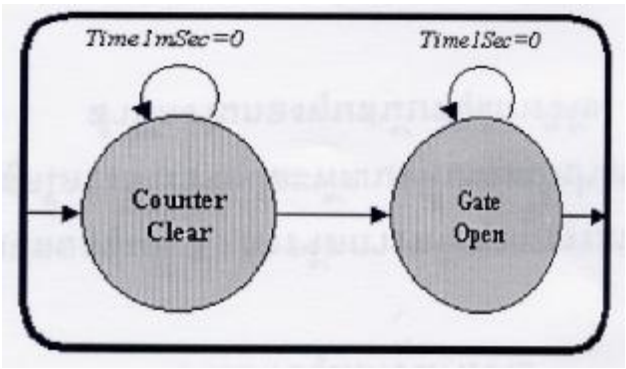

Figure 6. Statechart of the *HumidityInterfaceClass*-class

The final class which is called by the *DataCollectionThread* is a *HumidityInterfaceClass*-class. The main responsibility of this class is to control the process of converting from frequency to the corresponding humidity. The statechart of this class is shown in **Figure 6.** Notice that the entry state of the class is the "Counter Clear" state which has the duty to clear up the counter. In this state, the clear pulse of 1 mSec duration is sent to the counter. The next state is the "Gate Open" state which is responsible for generating a 1 Sec gate pulse. While the gate pulse is active, the counter counts the entire incoming clock generated by the C-to-F circuit. The count value, immediately after the gate pulse is inactive, is the frequency which can be translated into the current humidity value by means of a conversion table.

Apart from the previously described classes, we also develop a *Communication*-class which is responsible for automatically transferring the data file to the available FTP-servers when the system's buffer is almost full. This is to ensure that all data are

collected properly and stored appropriately. Furthermore, if the operator wants to see the experimental data real-time, the *Communication*-class supports a CGI script invocation. The real-time experimental data can be shown directly on the web browser.

IMPLEMENTATION, PRELIMINARY EXPERIMENTAL RESULTS, and DISCUSSION

Following the proposed architectural details described in the previous section, we implemented the proof-of-concept system by using the TINI390 module with 1 MB of memory on board. The softwares for every task of the system were developed in Java (5) with some TINI390 specific APIs, i.e. APIs for interfacing with the DS18S20 temperature sensor over the 1-wire protocol. During software development and testing, the standard Java compiler was utilized. The compiled source code, a Java-class, was then converted into TINI390's specific byte code format by using a freely available Java-tool named TINIConvertor. The output was then sent out to the TINI390 board, connected to the network, via FTP. To control the operation of the TINI390 over the network during software development phase, the Telnet was used. It was evident that the applications we employed to develop the system were available free of charge. This limited the cost of system development to what had to be paid for the system's hardware only.

However, the weakness of using the TINI390 module was the difficulty of verifying whether the written software was correct or not without executing it directly on a real target. This means that the system developer must possess a fairly high-level of skill in order to develop the system successfully within a short time.

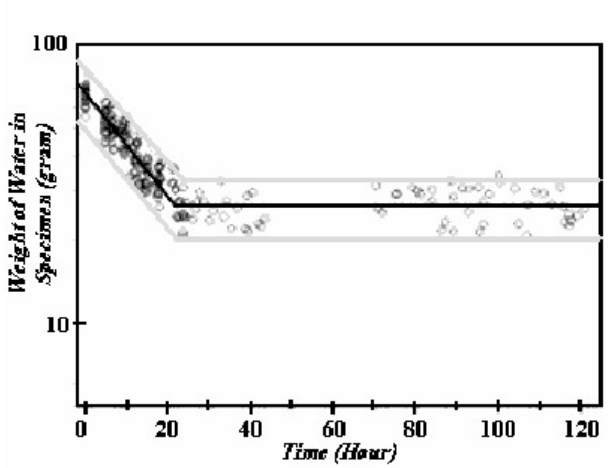

Figure 7. The sample experimental results from the proposed system

After testing the operation of the proposed system by allowing it to collect experimental data for a short period and analyze them, we concluded that the experimental data indicated that the system performed fairly well. The experimental data were comparable to those previously collected solely by a human operator. The sample experimental results are shown in **Figure 7.**

REFERENCES

1) Want R Harrison BL Fishkin KP Gujar A. Bridging Physical and Vitual Worlds with Electronic Tags, *ACM SIGGHI'99 Pittsburgh*, May 1999.
2) Lynch JP Law KH Kiremidjian AS Kenny TW Carryer E Partridge A. The design of a wireless sensing unit for structural health monitoring. Proceedings of the 3rd Int. Workshop on Structural Health Monitoring, Stanford, CA, USA, Sep 12-14, 2001.
3) Chomcham A Skaar C. Dynamic sorption and hygroexpansion of wood wafers exposed to sinusoidal varying humidity. *Wood Sci Tech 1983; 17: 259-77.*
4) Maxim Semiconductor Inc, *DS18S20 Data Sheet*, Electronic Document Available from www.maxim-ic.com.
5) Sun Microsystems Inc, *Java API Reference Document*, Electronic Document Available from java.sun.com.

บทคัดย่อ

วัฒนพงศ์ เกิดทองมี[1]

การติดตามตัวแปรและควบคุมการทำงานระยะไกลในการทดลองทางวัสดุศาสตร์

ด้วยความสามารถในการบรรจุทรานซิสเตอร์จำนวนมากมายลงในชิ้นส่วนวงจรรวมขนาดเล็กได้ส่งผลให้อุปกรณ์อิเลคทรอนิคส์และคอมพิวเตอร์มีความเฉลียวฉลาดเพิ่มขึ้น สวนทางกับราคาและความต้องการกำลังไฟฟ้า นอกจากนั้น ด้วยการเข้ามามีบทบาทของระบบเครือข่ายคอมพิวเตอร์ทั้งในรูปแบบที่มีสายและไร้สาย อันเป็นผลสืบเนื่องจากการเติบโตของการสื่อสารข้อมูลผ่านทางอินเตอร์เน็ตและเครือข่ายอื่น ๆ ยังทำให้เกิดอุปกรณ์อิเลคทรอนิคส์ที่พร้อมเชื่อมโยงกับเครือข่ายคอมพิวเตอร์ที่มีราคาถูกและถูกบรรจุไว้ภายในชิพวงจรรวมเพียงตัวเดียว โดยรวมอาจกล่าวได้ว่า เราสามารถเชื่อมโยงอุปกรณ์อิเลคทรอนิคส์ในแทบทุกรูปแบบเข้ากับเครือข่ายคอมพิวเตอร์ได้ ซึ่งอุปกรณ์อิเลคทรอนิคส์นี้ยังครอบคลุมถึงตัวตรวจจับและประมวลผลสัญญาณ บทความนี้ผู้วิจัยได้นำเสนอการประยุกต์ใช้งานอีกรูปแบบของอุปกรณ์เชื่อมโยงกับระบบเครือข่ายคอมพิวเตอร์ โดยการนำเอาอุปกรณ์ดังกล่าวมาใช้ในการควบคุมและเก็บบันทึกผลการทดลองในทางวัสดุศาสตร์ “การทดลองเพื่อการศึกษาความสามารถในการดูดซึมความชื้นของไม้” ผ่านทางเครือข่ายคอมพิวเตอร์ ผลที่ได้จากการประยุกต์ใช้งานคือ การลดภาระในการเก็บรวบรวมข้อมูลในระหว่างการทดลองที่ใช้เวลายาวนาน และประกอบด้วยขั้นตอนการทดลองที่จำเป็นต้องทำซ้ำหลายครั้งในหลายเงื่อนไขของตัวแปร

[1] สำนักวิชาวิศวกรรมศาสตร์และทรัพยากร อำเภอท่าศาลา จังหวัดนครศรีธรรมราช 80160