DATA-LOGGING USING COMMERCIAL OFF-THE-SHELF COMPONENTS

LOUBSER RC

Sugar Milling Research Institute, c/o University of KwaZulu-Natal, Durban, 4041, South Africa
rloubser@smri.org

Abstract

When considering data-logging for experiments, times have changed since the days when an analogue to digital card was chosen and an electronic expert was given the task of connecting a transducer to the computer. The Sugar Milling Research Institute has a pilot pan for research into various areas of sugar factory process, such as ideal molasses exhaustion. The data logging system originally installed on the pan used a system based on the Philips I^2C communication protocol. This system was implemented relying on internal timing in the computer. With the rapid increase in computing speeds it is no longer possible to support the legacy hardware.

It was thus decided that the obsolete hardware should be replaced with commercial off-the-shelf components. Several communication architectures were evaluated to decide on which one to use. The conclusion was that a universal standard which embraces modern technologies has yet to emerge. After considering several of the architectures, Modbus over Ethernet was chosen since support was available from many sources.

This paper considers the ways in which systems can be developed, and highlights how devices can be added or replaced and integrated into the system, often requiring only small changes in configuration. The system can be independent of supplier and eliminate the need for specialist electronic development.

Keywords: data-logging, laboratory equipment, Modbus, Ethernet, automation

Introduction

In the early years of data-logging using computers, the trend was to use analogue to digital cards that were plugged in to the communication bus of the computers. These devices needed specialist electronic knowledge to design the associated signal conditioning printed circuit boards. This approach was soon supplemented with external data acquisition devices which used standard personal computer interfaces such as the printer (LPT) or serial (COM) ports and also newly developed communication systems such as the Hewlett-Packard Interface Bus (HPIB) also known as the general purpose interface bus (GPIB) described by Davis (2007).

More recently, the Universal Serial Bus (USB), described in Anon (2008c), has been applied in laboratory data acquisition devices. Some problems have been experienced with electrical noise in an industrial environment where USB is used.

When the instrumentation for the laboratory pan at the Sugar Milling Research Institute (SMRI) was developed, the Philips Inter Integrated Circuit (I^2C) communication protocol described in Anon (2008d) and Wolf (2008) was chosen. The protocol was reliable and could be implemented using the LPT port on a micro-computer. The downfall of the system was that it used the timing characteristics of the computers available at the time. With the rapid increase in computer speed, the system could not be maintained using the same technology. Although the software could have been rewritten to embrace the newer computer hardware, some of the critical electronic components are no longer available. An alternative technology using readily available and simple to use components had to be found.

All the technologies mentioned rely on a dedicated personal computer to log the data. This was considered undesirable in a process laboratory where potential water leaks and condensation pose an electrical safety hazard. A system that could be controlled remotely using the existing local area network (LAN) infrastructure was considered the most desirable approach, provided it could be implemented as described by Seitz and Samuelian (2008).

Existing system

Pilot pan

The pan consists of a cylinder with a central impeller for forced circulation. Heating is achieved using an external element wrapped around the base of the pan. An externally applied vacuum is used to lower the boiling point.

Parameters that are measured include the temperature of the massecuite, and the vacuum from which boiling point elevation is calculated. Two electrodes are provided to measure conductivity as an indicator of concentration. Digital scales are used for measuring the masses of syrup and condensate.

Computer infrastructure

In a recent upgrade of the computer system, the majority of the ageing desktop computers were replaced with a central computer or server where the programs run, and ‘thin clients’, which are terminals through which the user accesses the central computer. The network was wired using unshielded twisted pair (UTP) and wireless access. The network protocol was transmission control protocol/internet protocol (TCP/IP) over Ethernet.

Where a data-logging system could be implemented using an application running on the terminal server and communicating using the Ethernet, it would be possible to eliminate the need for a dedicated data-logging computer. The data acquisition system could then be mounted in a water-resistant box without the need for user access.

Elements of new system

Analogue

The existing transducers installed in the pan were to be used for taking the measurements. The parameters and data acquisition details are shown in Table 1.

Any signal conditioning had to be accommodated within the analogue to digital (A/D) control box.

Table 1. Parameters measured.

Parameter	Transducer
Temperature	Resistance Temperature Detector (RTD PT100)
Vacuum	Differential diaphragm type transducer with 0-10V output
Motor current	0-10V output from controller
Torque	Strain gauge
Mass	Digital scale – RS232 interface
Power	Output to thyristor drive circuit 0-10V

Analogue to digital converter and signal conditioning
To minimise the requirement for specialist design and manufacture, all the circuitry used for conditioning and A/D conversion had to be commercially available and preferably conform to a standard. Although not essential, a DIN (Deutsches Institut für Normung) rail arrangement was preferred. This would allow the easy replacement of modules with equivalent modules should a failure occur. Of particular importance was that the technology used should not be unique to a specific supplier. The communication between the A/D and the acquisition computer was either Ethernet or wireless LAN. This communication used TCP/IP.

Data acquisition and computing
The number of inputs from the instrumentation is small. The pan is a research tool; consequently, flexibility in the operation procedure is essential. The task of the software was purely to log and display the output from a limited number of transducers and possibly set the power of a heating element. To avoid introduction of a dedicated computer, the existing remote desktop server was chosen to perform the function of data acquisition computer.

Technology selection

Analogue to digital conversion
There are many protocols for networking devices. These include Controller Area Network (CAN) bus (Schofield, 2006), Fieldbus (Anon, 2007a; Anon, 2007b) and its derivatives and DeviceNet (Allen-Bradley, 2008), to name a few. These systems connect between a control module and the transducers themselves. The controller module has to convert between the wiring scheme associated with the system and Ethernet.

A search of the offerings of local suppliers showed that the Fieldbus-derived device networks were the best supported. The choice was then what variant to use. There were two competitors, namely Profibus (or Process Fieldbus) (Anon, 2008b) and Modbus (published by Modicon) (Anon 2008a).

Profibus is a mature standard designed for robust industrial control. It is an ideal platform for implementation of a Supervisory Control And Data Acquisition (SCADA) system. With its level of sophistication, it goes far beyond the needs for a simple data acquisition system based on about five channels.

Modbus is well supported in the open source community. It is a serial communication protocol which uses RS485 connections (amongst others) to communicate to the node devices. Although one of the criticisms of Modbus is the inability of the system to report errors effectively, it has the advantage of easy implementation, as commented in Anon (2007c), with many user discussion groups. Drivers for Modbus communication over Ethernet are readily available. Modbus can easily be integrated into a SCADA network.

Communication protocols
The protocol selection is primarily determined by the choice of communication bus, in the case of this project Modbus. In other words, Modbus over TCP was used for moving measured values from the transducer to the data acquisition computer.

Driver software
SCADA: Half the function of a SCADA system is to acquire data and present it. As with all software, there are commercial products and open-source developments.

For the laboratory pan in a relatively small system, the expense associated with a commercial package could not be justified. This left the option of finding an open-source SCADA version. After studying the various offerings, it was concluded that there was no package that contained drivers for all the systems available. Consequently, it would be necessary to add drivers for specific devices.

After weighing the learning requirement against the advantages offered by the various packages, it was decided to generate a small application for the purpose, using a familiar computer code development environment.

OPC: Object Linking and Embedding (OLE) was developed to allow one computer program to access data from another. Data can be transferred by linking between the two applications or the data from the application, the OLE server, can be embedded in the other, the OLE client. OLE for Process Control (OPC) is a further development which allows data to be transferred to and from a process transducer. To achieve this seamless transfer of data between the transducer and computer, strict rules, or open standards, have been put in place by the OPC Foundation (Anon, 2008e) to ensure compatibility of the software components. Each type of device in the system would have a specially developed OPC server through which it communicates. The data-logging program would have a component which would act as an OPC client through which the data could be transferred.

This approach once again assumes that an OPC server exists for the communication protocol that was chosen.

Bus-specific drivers: The use of Modbus over TCP had already been identified as the most promising candidate for the communication protocol. Delphi was chosen as the software development platform to exploit existing programming skills. A brief search for software components for driving Modbus over TCP using Delphi showed that these were readily available for purchase. This served to underline the conclusion that Modbus has an extensive support community.

Program description

Main form

The main form is shown in Figure 1. Five functions that the program needed to perform were identified:

- Configure transducers and data acquisition equipment
- Acquire data
- Display data
- Store data
- Set control signals.

Each of the analogue quantities had to be set up separately and stored. A button was introduced to invoke the configuration forms. This was placed on a separate form because a fair level of expertise is required to select the correct addresses and ranges.

When configuration is complete, data acquisition and control can start. As this is the function that will be used most often, the few controls required were incorporated in the main form for easy user access.

Provision was made to display the data in the form of a mimic panel, tabulated data and a graph. Each of these could be activated with a button.

Features for saving and printing the data were also included.

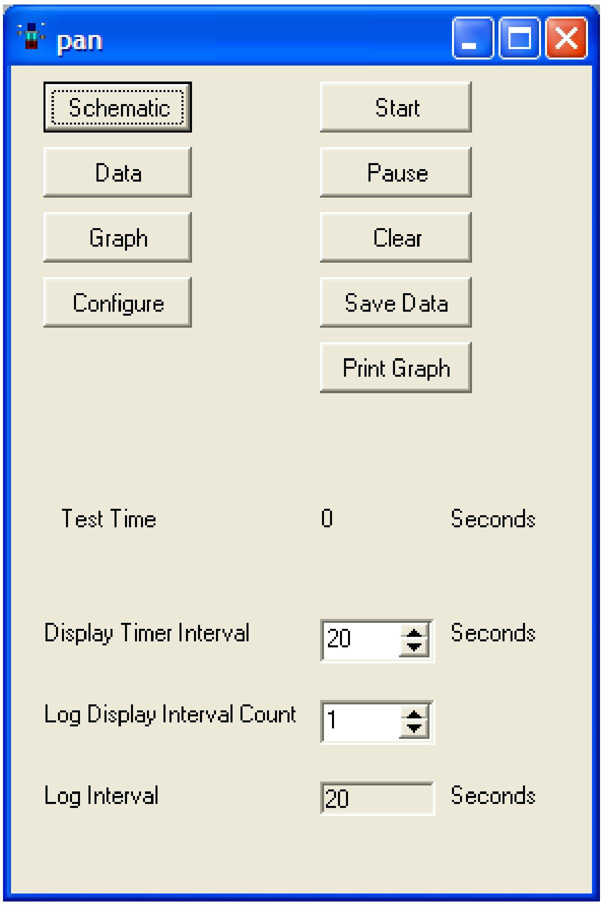

Figure 1. Program main form.

Configuration

The configuration step requires detailed knowledge of the instrumentation, network and interface. Each TCP to serial Modbus converter requires that the network administrator allocate an IP address to be used by the module. Thereafter, the Modbus address is set on each interface using switches on the module. The range, offset and gain need to be set to convert the transducer output into engineering units. A typical configuration for a channel is shown in Figure 2.

Data presentation

The data could be presented in three ways. A mimic, shown in Figure 3, showed the output from the various transducers. In addition to this a table was constructed with the data. This could be exported for analysing the process later if necessary. A graph was provided for easy identification of trends, so that appropriate changes could be made in the control levels. Analogue and digital outputs could be changed through relevant fields on the mimic panel.

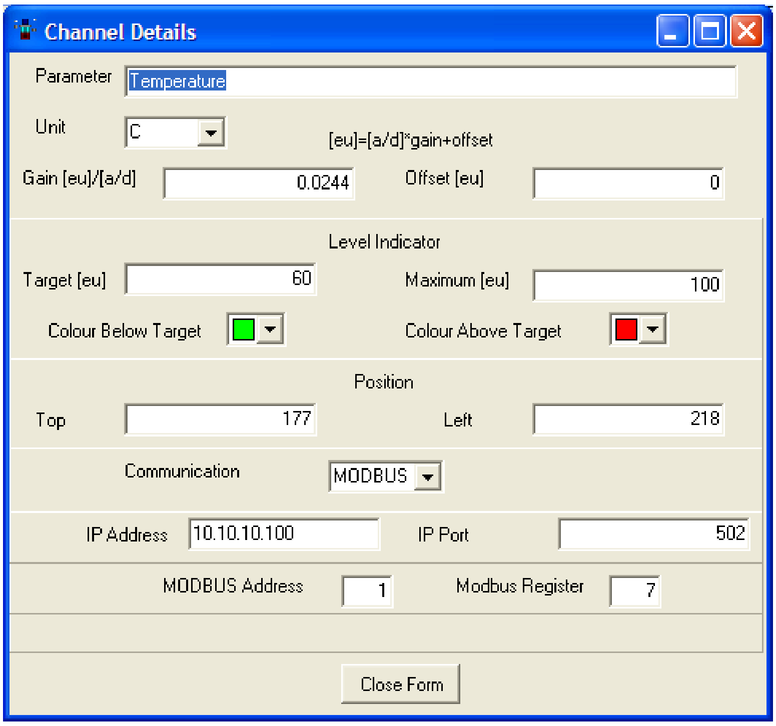

Figure 2. Configuration window.

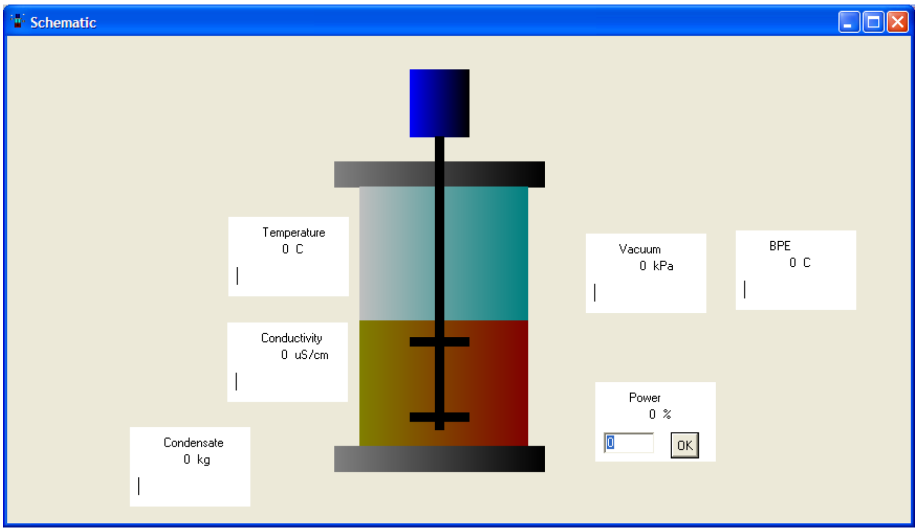

Figure 3. Mimic panel.

Conclusion

A small laboratory pan control system was constructed for use at the SMRI. The constraints imposed on the project were that the system needed to be integrated into the existing computer network without a dedicated personal computer to act as controller. The system had to be compatible with the activities in the process laboratory and not pose additional safety hazards.

An approach of converting transducer output to a digital signal close to the measurement point eliminated the need for large cable harnesses. This helped to reduce the time required for repairing cable faults and facilitated easy installation.

The option of using a SCADA system was evaluated. This would have been ideal for a larger system. In the case of the laboratory pan it was decided to purchase a Modbus driver and integrate the system using a familiar high level programming language.

The use of digital electronic modules has eliminated the complication of larger numbers of cables which can develop hard-to-find faults. A dedicated computer with the associated maintenance and space issues was also eliminated.

Acknowledgements

The author would like to thank Mike Gooch for his assistance in choosing the components, assembling the system and preparing this paper.

REFERENCES

Allen-Bradley (2008). Networks and communications DeviceNet, http://www.ab.com/networks/devicenet.html, Rockwell Automation Inc. [accessed 10 January 2008].

Anon. (2007a). Fieldbus Foundation, http://www.fieldbus.org [accessed 10 January 2008].

Anon. (2007b). Fieldbus, http://en.wikipedia.org/wiki/fieldbus [updated 30 November 2007, accessed 4 January 2008].

Anon. (2007c). Talk:Modbus, http://en.wikipedia.org/wiki/Talk:Modbus [updated 19 April 2007, accessed 4 January 2008].

Anon. (2008a). Modbus, http://en.wikipedia.org/wiki/Modbus [updated 4 January, accessed 4 January 2008].

Anon. (2008b). Profibus, http://en.wikipedia.org/wiki/Profibus [updated 11 December 2007, accessed 4 January 2008].

Anon. (2008c). USB, http://en.wikipedia.org/wiki/USB [updated 3 January 2008, accessed 4 January 2008].

Anon. (2008d). Using the I2C Bus, http://www.robot-electronics.co.uk/htm/using_the_i2c_bus.htm [accessed 4 January 2008].

Anon. (2008e). What is OPC?, http://www.opcfoundation.org/Default.aspx/01_about/01_whatis.asp, OPC Foundation [accessed 4 January 2008].

Davis L (2007). GPIB Bus, http://www.interfaces.com/Design_Connector_GPIB.html [accessed 4 January 2008.]

Schofield MJ (2006). Controller Area Network-background information, http://www.mjschofield.com/history.htm [accessed 4 January 2008].

Seitz B and Samuelian M (2008). A universal approach for implementing real-time Industrial Ethernet, http://www.industrial-embedded.com/articles/seitz_and_samuelian/ [accessed 10 January 2008].

Wolf A (2008). I2C (Inter-Integrated Circuit) Bus technical overview and frequently asked questions, http://www.esacademy.com/faq/i2c/ [accessed 4 January 2008].