Real-Time Data Collection with SAS® System Applications

Bradley W. Klenz, Meridian Software, Inc.

Introduction

More and more often, the SAS System is being chosen as the solution for real-time data collection applications. The same features that make the SAS System the preferred tool for so many other types of applications mean that SAS software products are candidates in real-time environments: power, flexibility, breadth of functionality, integration, open architecture… we all know the strengths of the SAS System. However, developing SAS applications for real-time environments requires some skills that go beyond those required for other types of SAS applications. Application requirements include

- developing an interface between the SAS System and data collection devices
- creating methods for managing multi-user contention asynchronously.

This paper is intended for the SAS user who wants to establish a more direct link between the real-time data being collected at his or her site and the SAS System. The paper addresses the special considerations for developing a real-time data collection application with the SAS System. With this checklist for guidance, you will be well on your way to creating a real-time SAS data collection application.

Definition of a Real-Time Data Collection Application

Real-time data collection applications are typically seen in manufacturing and laboratory settings. In this paper, the focus is on manufacturing applications where specialized hardware devices take measurements of the manufactured product. These measurements are usually collected by an application and available for Statistical Process Control (SPC) analyses. In this scenario, the application collecting data and performing SPC analyses is a SAS application.

The components of the application include

- a process that manages the data collection
- a system for performing the desired SPC analyses.

The process that manages the collection of data measurement values is responsible for communicating with the measurement hardware, coordinating the collection of data from multiple devices simultaneously, and making the collected data available to other applications.

The second component of the real-time data collection application is a system that performs SPC analyses on the collected measurement data. This component is usually implemented as a SAS/AF® application or a series of SAS programs that execute the applicable SAS/QC® procedures. Because this component uses more typical SAS programming techniques, it is not discussed in detail in this paper.

Another point relevant to this discussion of real-time data collection applications is the interaction between the application and the users (or devices) that are collecting the data measurements. In the simplest case, this interaction may be essentially one-way. The device takes the measurement and the only response from the application is that the measurement was received. A more complex application might allow a user to take a measurement using a hardware device, then receive some feedback regarding the quality of that measurement. The application might then enable the user to respond to the feedback from the initial measurement.

Types of Data Collection Devices

There are several types of data collection devices that can be used in a real-time data collection application. Some data collection devices simply take and record individual data measurements. Another type of device provides the capability to interact with the user to provide feedback or query for additional information in addition to allowing data collection. It is also possible to combine data collection devices with a workstation to provide a more comprehensive data collection station.

In the most straightforward, but not necessarily the simplest case, a data collection device can be a specialized piece of hardware that takes a certain type of measurement and transfers the measurement electronically to the application. Some examples of this type of device include a Mettler® balance, which weighs items and sends the measurement through a cable to the serial port of a PC. Another example is a Mitutoyo® indicator gauge, which measures a physical dimension of an item and transfers the measurement by cable to a specialized protocol device that makes that value available to a connected computer.

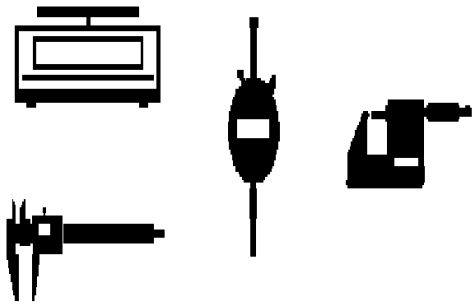

Figure 1 Data Collection Devices

More complex data collection devices are available that allow not only the collection of data values, but provide the ability to give feedback to the user collecting the data measurements, and even prompt the user for additional information depending on the quality of the measurement. These data collection devices are sometimes referred to as remote data entry devices. They are independent hand-held data terminals that connect with the application through either an infrequent cable connection or a continuous wireless connection. The devices can be used by technicians roaming to different locations on the shop floor or by engineers in the field. An example is a Telxon® Portable Tele-Transaction Computer, which has a small display, keypad, barcode scanner, and wireless interface.

Figure 2 Telxon Portable Computer

The availability of industrial grade PCs allows the use of data collection workstations. Standard data collection devices can be connected to the workstation and controlled by a custom application. The user collecting the data can get feedback on a measurement or a set of measurements. The feedback can be text messages or even SPC-oriented charts.

Connecting the Data Collection Device to the Application

When the type of data collection device needed for the application is determined, an interface must be developed between the SAS application and the chosen data collection device. This interface consists of a communication method and software to manage the communication between the device and the application. In this paper, this software is referred to as 'interface bridge software'. The choices involved in developing the interface between the device and the application are described in the next sections. The choices are influenced by the

- data collection device chosen
- communications methods available for the selected operating system
- skills available to create interface bridge software
- degree of multi-user contention that must be handled.

Selecting the Communications Method

One of the major issues that needs to be resolved when building a real-time data collection application is the actual communications between the data collection device and the application. These data communications tend to be operating-system specific. Real-time data collection applications typically run on a UNIX system or PC-based system, so connections available for these environments are discussed below.

First, the physical connection to the device must be handled. Many devices use an RS232 connection directly to a serial port. For some devices a special protocol adapter is needed to connect the device to the serial port on the computer. There are even some devices that have a network connection built into the device.

But the physical connection is only part of the challenge; you must also decide how messages will be sent from data collection devices to the SAS application. The types of communication methods available include:

- Serial port communications (Windows® and OS/2®)
- Pipes and Named Pipes (UNIX and OS/2)
- Shared Memory (OS/2)
- TCP/IP (UNIX, Windows, and OS/2)
- Message queues (OS/2)
- Dynamic Data Exchange (DDE) (Windows and OS/2)

The choice of a communication method is constrained by the

- methods that are supported by the data collection devices
- operating system the application is running under
- functionality needed.

The following paragraphs outline some of the differences between the communication methods. The communication method used in an application is closely tied to the interface bridge software and the multi-user functionality.

Serial port communications are accomplished by having a part of the application read the data values directly from the serial port. This part of the application can be written in SAS using the COMMPORT device type in the FILENAME statement. Another alternative is to have a driver program that reads from the serial port and then sends the measurement values (using a different communications method) to the SAS application. The serial port communications method by itself does not allow flexibility to handle multiple devices concurrently because a direct continuous link is needed between the serial port and the application.

The remaining communication methods are intended to allow two or more processes in an application to communicate with each other. In some cases, such as pipes, a direct connection is established between the two processes. No other processes can be involved in the connection. If either process stops and restarts, and the prior connection cannot be re-established, a new connection must be made.

Using a communication method like DDE allows one process to be designated as a server with multiple processes connected as clients. The clients can disconnect and reconnect later to continue with the data communication.

When using a communication method that is intended to communicate between processes, there is usually a small process dedicated to each measurement device. This process then connects and communicates with a process that receives the measurement values into the application. The second process may be the SAS application itself, but in many applications the second process is dedicated to coordinating the measurement values being received from multiple devices at one time. Later in this paper techniques are discussed on how to handle multi-user contention.

Interface Bridge Software

In some cases, you can use the SAS System directly to handle getting information from the data collection device to the SAS application via your chosen communications method. It is more likely, though, that you will need some type of software bridge to complete the communications link. A software bridge can be developed exclusively for your application, or you may be able to use something "off the shelf" from a third-party vendor.

The SAS System provides some support for interfacing directly to data collection devices. This support includes the FILENAME interface with the COMMPORT device type, the DataMyte® CALL routines available with SAS under Windows and OS/2, and access to pipes and named pipes.

When using the COMMPORT device interface or the pipe interface, you would typically have a SAS session dedicated to reading measurements from a single device. This SAS session can be the main SAS application if multi-user data sharing is not required.

If multiple devices are needed to collect all the data measurements for the application, the main SAS application can be a separate process. In this case the main SAS application would receive the measurements from the device-dedicated SAS sessions using a process-to-process communication method, or possibly SAS/SHARE®. Refer to the SAS operating system companion documentation for additional information on these interfaces.

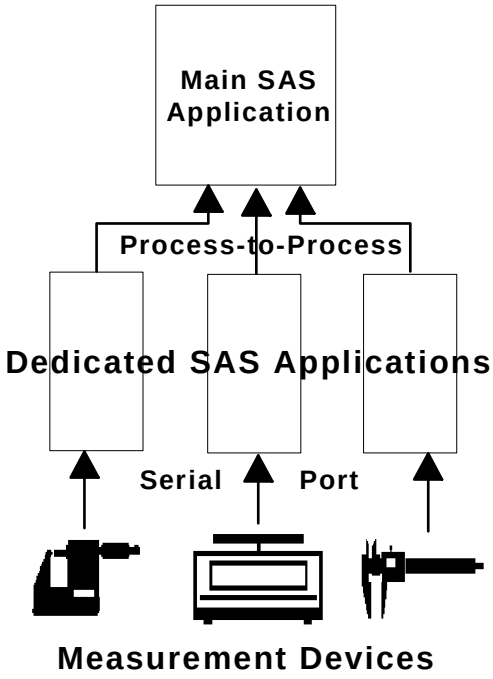

Figure 3 Serial Port Communication

SAS Institute also has a real-time data collection server that runs independently of the SAS System. The server is available under UNIX and uses TCP/IP to communicate with the data collection devices. Although the server itself is not written in SAS, it is specifically designed to provide real-time data collection functionality for SAS applications. The data server has the capability to receive multiple device measurements simultaneously and funnel the measurements into a single input stream for the SAS application. The Institute's server also allows for user-written device drivers to support other communications methods. See the SUGI 19 paper presented by Amitava Ghosh for additional information.

You may also be able to use software available from a third-party vendor, such as the Software Wedge™ from TAL Enterprises. The Software Wedge product takes data collection messages from PC serial ports and makes the messages available using DDE. This product can automatically configure and communicate with the PC serial ports. It also allows for filtering and translation of values received. Using these features, a more consistent stream of data measurements can be sent to the main application. The support for a process-to-process communication method allows for a more reliable connection, which can be disconnected and reestablished as needed.

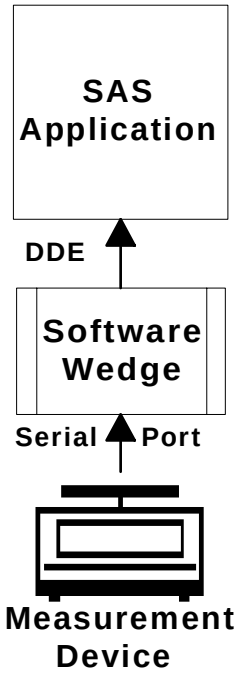

Figure 4 Using Third-Party Software

Another interface option requires creating custom software to bridge the gap between the data collection devices and your SAS application. A custom bridge could be a user-written SAS procedure using SAS/TOOLKIT® or the SAS external MODULE capability. You can also write a routine that translates the messages from the data collection device into a communication method that the SAS System supports directly, such as pipes or DDE.

Event-Driven Program Flow

Another special requirement in a SAS application for real-time data collection is event-driven programming. Typically, the SAS System looks for input from only one source at any particular moment. In a real-time application, you may have data input from the collection devices at random intervals.

You may also need part of your application available for user interaction, such as administrator control for starting and stopping the application. This creates a demand for handling events from different sources, which can be accomplished by having multiple SAS sessions, each receiving input from a different source. One SAS session can be a data server looking for input from the data collection devices, while a second SAS session provides a SAS/AF interface for user access to the collected data and for system control tasks.

In the most complex situation, the application is controlled by a SAS application running under a separate session that handles requests from other processes that connect to it. This main server application normally waits for a client application to connect to it using one of the process-to-process communication methods.

When a client application connects and sends a request, the request is acted on and an acknowledgment is sent if needed. The client application can receive data measurements or input via a user interface. Note that the server application is only looking for one type of input: a request from a client application connection. The client applications can receive input from whatever type is needed for that function of the application.

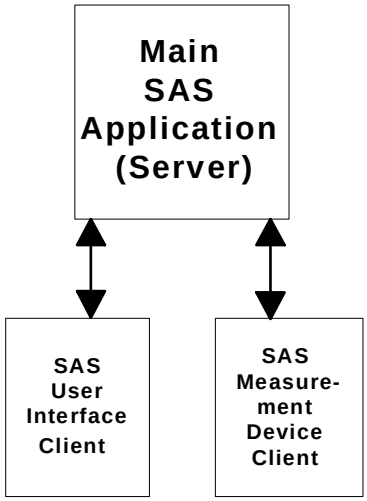

Figure 5 Main Application As a Server

Handling Multi-User Contention

In most cases, a real-time application must allow for input from a number of data collection devices operating at the same time. This means that you have to build the application to handle multi-user contention. You need to ensure that two data collection messages generated at the same time are both received and recorded. One place where the contention can be handled is in the interface bridge software. Many of the communication methods have routines available to queue messages that are received. The interface bridge software can use these routines to create a single data stream that can be read by the SAS application.

One method is to write a SAS application that runs in a separate SAS session to act as the data measurement server. It is usually preferable to use SAS/AF to write such an application because SCL has the flexibility to connect and disconnect to the other processes that are collecting the measurements. A process-to-process communication method must be used to receive the measurement values. The data measurement server must be written carefully because SAS does not provide an automatic mechanism to queue up new measurements that arrive while a previous measurement is being processed.

A good strategy for handling this situation is to write the process that controls the data collection device so that it does not continue with another measurement until it gets an acknowledgment that the current measurement has been saved. If the data collection process receives a return code that indicates that the server is busy, it should retry the connection to the server until the measurement can be sent and the acknowledgment received. This method has the major advantage of allowing each new measurement to be processed by the main SAS application as the measurement is collected. After the new measurement is processed, a message or signal can be sent back to the data collection process in the acknowledgment.

Another method to handle multi-user contention is to have a separate process for each data collection device and use the routines available with a database server to coordinate data flow. This database server can be SAS/SHARE or some other database package. With SAS/SHARE your application could receive a measurement from a data collection device and use PROC APPEND or the SCL APPEND function to add the measurement to a central data set. The SCL APPEND function does allow for easier return code checking to

make sure the observation was added correctly. One disadvantage of this method is that it does not have a mechanism for signaling the main application that a new data measurement has been added. Another method would be advisable if the main application needs to do some additional processing as measurements are added to the database.

You also have to consider other features of the application and how they affect multi-user contention. For example, the application may receive a message from a device and need to query the user for input based on the message received. This functionality should be written to allow other devices to send messages while waiting for the first user to respond to the query. For example:

1. A new data measurement is received by the main server application from user A.
2. The main server determines the measurement from user A is outside a calculated limit and sends a prompt back to user A to verify the measurement is correct. (It is not known how long it will take user A to respond to this prompt.)
3. The application should disconnect from user A's data collection process and be available to receive new measurements from user B.
4. When user A responds to the prompt, the data collection process can reconnect to the main server application and transmit the response.

Tools

There are a number of tools available to create a real-time data collection application. SAS Institute provides products such as SAS/TOOLKIT, the real-time data collection server, and the external MODULE routines. The tools from the Institute can provide the way to write a custom interface that allows your data collection devices to interact directly with the SAS System.

There are also control and configuration packages made for use with data collection devices. Some of these packages are proprietary for specific devices. One example here is the RFXpress from Telxon. This package is a data measurement server that can be used to control Telxon data collection devices only.

Other packages are independent of the measurement device vendor and can support devices from a variety of vendors. The GPWorks/GagePort package from GageTalker® Corporation can connect a number of different devices to a PC serial port. This package allows multiple devices to be attached to one serial port.

Interface bridging can be accomplished using an off-the-shelf package like Software Wedge. This package lets the developer specify how the device and the application communicate through a GUI so that no additional programming is required. For a connection requirement that is not available from a standard package, the software bridge can be custom written using any standard C compiler.

Conclusion

Although real-time data collection applications require development skills that are not typical for SAS applications, the SAS System is a viable option for these types of applications. There are a number of alternatives to consider when designing the data collection application, but first a definite set of issues must be resolved. This paper has outlined the need to address the following design issues:

- Decide what type of data collection device is appropriate for the data collection task.
- Determine which communication method should be used.
- Select the interface bridge software to connect the measurement device to the application.

- Evaluate the need for the application to handle multi-user contention.

This paper does not detail any specific implementations for a proposed application. Please see some of the references listed below for descriptions of actual systems that were implemented using the proposed methods.

Feel free to contact the author for more information on any items presented or to discuss any data collection applications that you have encountered.

The contact address is:

Brad Klenz
Meridian Software, Inc.
12204 Old Creedmoor Rd.
Raleigh, NC 27613

Phone: 919/518-1070

Fax: 919/518-1170

Email: merbwk@interpath.com

References

Here are additional sources providing information for writing real-time data collection applications using the SAS System:

- Donna Fulenwider, Bradley Klenz, and Lorraine Schacht (1995) "Implementation of the FREOP System, The Quality Solution Provided for Northrop Grumman Corporation" Proceedings of the Twentieth Annual SAS Users Group International Conference
- Amitava Ghosh (1994) "Enhancements to the SAS System to Support Data Acquisition in a Real-Time Environment" Proceedings of the Nineteenth Annual SAS Users Group International Conference
- Amitava Ghosh (1995) "XX Access Methods" Proceedings of the Twentieth Annual SAS Users Group International Conference
- Richard Langston (1995) "Examples Using the MODULE Routines In PC Environments" Proceedings of the Twentieth Annual SAS Users Group International Conference

Windows is a registered trademark of Microsoft Corporation.