CIRCUIT CELLAR INK®
THE COMPUTER APPLICATIONS JOURNAL

PC/104 QUARTER

David Chanoux

Embedded PCs

Embedded Systems for Weight and Force Measurement Applications

Until recently, there has been no tool to integrate weighing sensors with embedded PCs. David lets us in on some recent improvements in the link between the PC/104 standard and load-cell technology.

Suppliers of real-time weighing and force measurement systems now have an integration tool to connect weighing sensors to embedded PCs. Three criteria have been met, paving the way for PC-based weighing applications.

First, good-quality, high-performance, ADCs are changing weighing limits. Applications can be implemented differently.

Second, microcontrollers with easy-to-use bus interfaces allow weighing applications to be segmented, using the vast processing, graphics, storage capabilities, and presence of today's PCs to advantage.

And third, the emerging embedded-PC bus standard, PC/104, enables effective industrial packaging, so the PC is installed where weighing takes place.

The Scanning Devices PC/104-Compliant Load-Cell Controller shown in Photo 1 uses these new developments to enable PC-based weighing systems.

This article describes weight- and force-measurement applications, tracing their evolution to today's microcontroller-based systems, and projects the effect of these developments on tomorrow's embedded PC-based weighing systems.

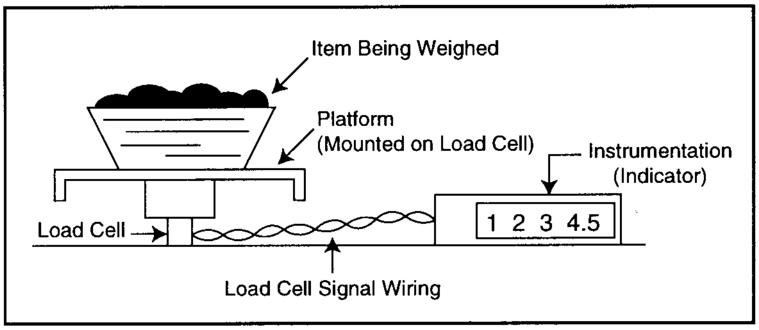

***Figure 1:** The load cell supports a platform and the weight being measured. The load cell is connected to an indicator which provides bridge excitation and converts the signal to weight. Notice the load cell is down under the weight!*

TODAY'S PC IN WEIGHING SYSTEMS

Until recently, electronic weight- and force-measurement systems have been the domain of microcontrollers. The PC has been kept at arm's length.

The highly valued portion of a weighing instrument is its ability to precisely measure a relatively small analog voltage signal representing the weight or force. The serial link to a PC for data or process parameters used for measurement was of less interest and often ended up being one-way. From the PC's point of view, it was input only!

System builders requiring the capabilities of both PCs and weighing indicators had no choice but to connect them via either EIA (RS-232, RS-422) or 20-mA current loop interfaces, with the PC system designer accepting the limited data-exchange capabilities found in most digital indicators. (The weighing industry uses "indicator" to describe an instrument working with a load cell to measure and display weight.)

But first, let's get some background on weight and force measurement.

MECHANICAL WEIGHING

Mechanical weighing systems are typically spring-based or lever-based instruments. A bathroom or produce scale is a common spring-based scale which deflects a spring to rotate or displace a dial, thereby displaying the weight. These scales are notoriously inaccurate.

Lever-based instruments balance the unknown weight with a known weight on the end of a lever arm and are accurate enough for legal-for-trade scales. However, reading and recording the weight is left to the viewer's interpretation.

ELECTRONIC WEIGHING

Electronic weighing centers around strain-gauge transducers or load cells, sensing instruments which convert applied force to a resistance change. Figure 1 illustrates a typical electronic weighing system consisting of platform, load-cell transducer, and indicator.

The transducer is configured as a Wheatstone bridge made up of four resistive elements. Two of these strain gauges are mechanically react positively to applied force, while the remaining two react negatively to the same force. An excitation voltage applied to one pair of bridge terminals transforms to a measurement-signal voltage on the other.

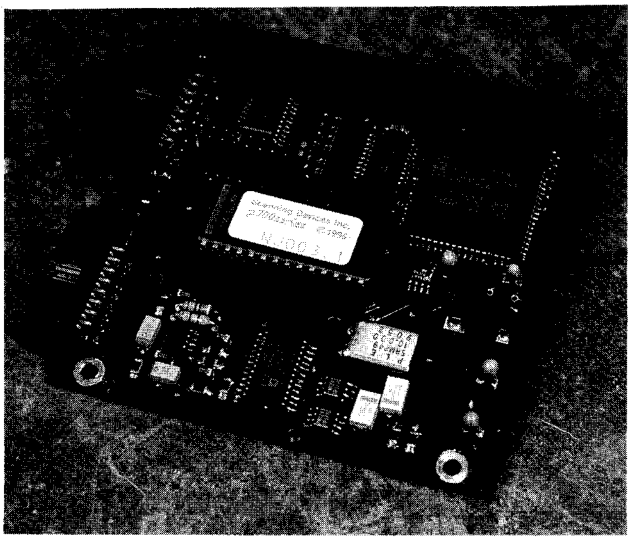

Photo 1: Specialized integrated circuits—selected and configured for by application—are controlled with specialized firmware. They keep the part count at a minimum and allow implementation in the PC/104 compact format.

With no force or weight applied, the bridge is balanced and the signal voltage is zero. As force is applied, two strain gauges are put in tension, the other two in compression: their effective resistance changes in opposite directions. The Wheatstone bridge produces a signal voltage proportional to the applied force.

Load-cell specifications include capacity (the full load, such as 100 pounds) and output voltage ratio. The output is typically 2 mV/V (i.e., 2 mV of signal voltage per volt of excitation at capacity). If the excitation is 10 V, the load-cell signal is 20 mV with 100 pounds applied.

The many capacities, mechanical configurations, and special features available with load cells provide almost unlimited applications. Photo 2 shows a typical $350 beam-style load cell applicable to low-profile scales. Excitation and the corresponding measurement-signal voltages are AC or DC, giving further design discretion to the transducer engineer.

By nature, the Wheatstone bridge enables load cells to be used individually or combined for additive weighing. For example, a platform may be supported on four load cells which are electrically connected so that their signals sum to one equivalent load cell. The single measurement signal represents the total weight on the platform, regardless of weight distribution.

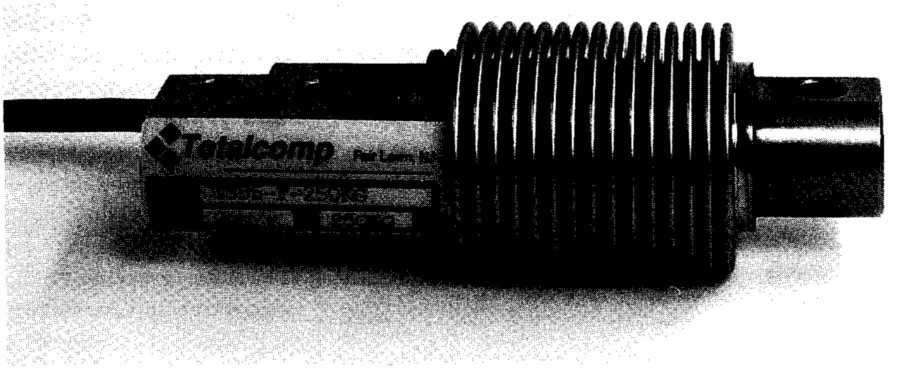

Photo 2: The load cell converts force to an electrical signal. This sensor is the heart of electronic weighing systems. It consists of a strain-gauge transducer designed to produce a resistive change with applied force. The load cell is configured as a Wheatstone Bridge, allowing the resistive change to be measured precisely.

THE CHALLENGE

At first glance, the application seems straightforward: pass the signal through an ADC, do some math, drive a display or a printer, and if communications are needed, write to the serial port. Any PC with a general-purpose analog-input channel suffices. Anybody could do this, you say.

Not so fast! Let's introduce some real-world requirements and complications to illustrate why the PC–load-cell combination is not installed in every scale. Let me also describe how a functional indicator deals with these complications. The Scanning Device Load-Cell Controller illustrated in Figure 2 shows the required components. Each has a purpose—bypassing any one compromises performance.

THE REAL ISSUES

The signal is differential and its magnitude is typically measured in millivolts. A typical analog-input channel in data-acquisition systems delivers 12 bits of signal precision, measured with the assumption that the input signal spans a 0–5-V range. That is, the digital conversion is a 12-bit representation of a 5-V input.

If the input is only 5 mV, (0.001 of the ADC's input range, but not atypical of a load-cell signal), the 12-bit conversion result would be ten bits equal to zero followed by two bits of significant data.

Clearly, an input-amplifier stage is required. An input-amplifier stage consisting of an instrumentation amplifier converts the small differential input into a range acceptable to an ADC. The amplifier also provides differential input and gain set by a single external resistor. Its offset control puts the amplifier output in the right range.

The Analog Devices Series 620 is an effective instrumentation amplifier for this application. It is functionally equivalent to the three-amplifier configuration on the right in Figure 3. Fewer parts, reduced noise, and controlled gain favor the instrumentation amplifier most times.

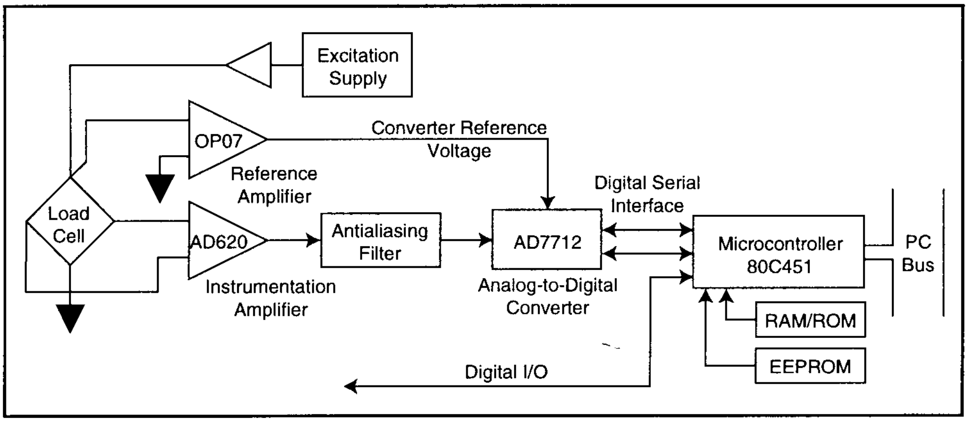

Figure 2: Key components of the Scanning Devices Load-Cell Controller generate load-cell bridge excitation, condition the ADC with signal and reference, process the digital measurement signal, and respond to the PC via PC/104 bus. Onboard processing and memory enable real-time measurement and control under PC supervision.

Traditional electronic weighing systems offer input-amplifier stages with dual adjustments. A technician can set the operating points of the input-amplifier circuitry—zero and span—via some means.

The zero adjustment removes dead weight—that not significant for the measurement—from the hardware. The zero adjustment sets the input amplifier to produce a minimum voltage when only the dead weight is applied.

The span adjustment sets the gain of the input amplifier so that the maximum voltage is generated when maximum weight of interest is applied.

But if you are designing an embedded system, you don't want to provide adjustments. Why are zero and span necessary? Precision.

Quality load cells achieve precision to 1 part in 5,000. Good ones do even better. To put that in the perspective of ADC specifications, that's 14 bits of precision in the result. That statistic assumes the load-cell signal spans the full input range of the ADC. In most cases it doesn't.

Minimally, an input-amplifier design must allow for overrange voltages and negative measurements while producing a signal so a converter can achieve 16-bit precision. The high-speed 12-bit sample-and-hold converter common on many computer I/O modules is just not the right tool for this task. What should we use?

ADCs using the sigma-delta technique are capable of up to 24-bit precision (e.g., the Analog Devices AD7710 Series converters). They use successive approximation to achieve low-frequency measurements with high precision, exactly the characteristics needed for weighing.

Conversion rates of 50 Hz are often adequate in static and quasistatic weighing. Giving up speed in return for precision is clearly the right choice for weighing applications.

The Analog Devices AD7712 Sigma-Delta Converter (see Figure 4) with serial digital interface lets you configure performance for applications. Surrounding the converter is linear-input circuitry tuned to achieve high precision with the acceptable speed and digital processing to interpret the conversion results.

Isn't 24 bits overkill? No. Let's examine how we can use the precision.

Scanning Devices tested many good-quality load cells to determine how much precision can reasonably be expected. Without presenting data, let's assume that a good-quality load cell delivers 18 bits of significant data (i.e., 1 part in 262,000).

With a 24-bit conversion on a full-scale signal, we find 18 bits of significant data and 6 bits of random noise. We can argue at length about the data, how much significance can be extracted, and by what means. But, let's assume 18 bits and address the zero and span adjustments.

Making the weighing range span the converter's input-voltage range assumes the converter is the limiting factor in measurement precision. If the weight range of interest spans 4 V (e.g., 0.5–4.5 V at the input to the converter), the converter produces a result with some significance.

If the same weight range of interest spanned 1 V at the input to the converter, the result has only one-fourth the significance. The result is equivalent to taking the conversion from the 4-V case and shifting it two bits to the right, causing the two least significant bits to be lost. Assuming the converter is the limiting factor, the zero and span adjustments prevent this loss of significance, maximizing measurement precision.

But if we have more precision than necessary, why expose the application to adjustments? Load cells are specified so the input range can be determined for a given load cell, .

Let's make the input amplifier fixed so it

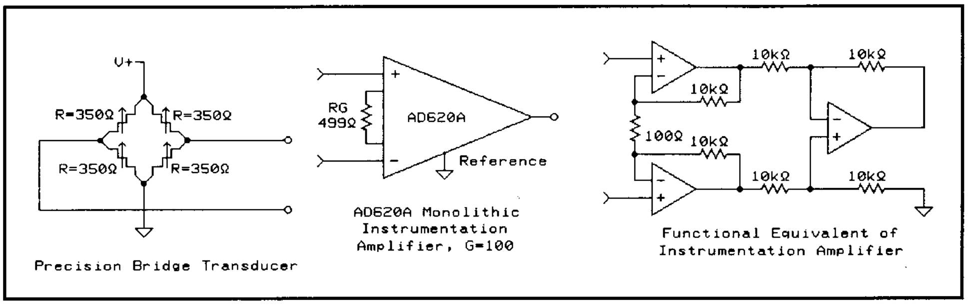

Figure 3: The instrumentation amplifier provides differential input, controlled gain, and offset. It is functionally equivalent to the three-amplifier configuration shown. Performance and cost issues guide selection of the best input-amplifier configuration.

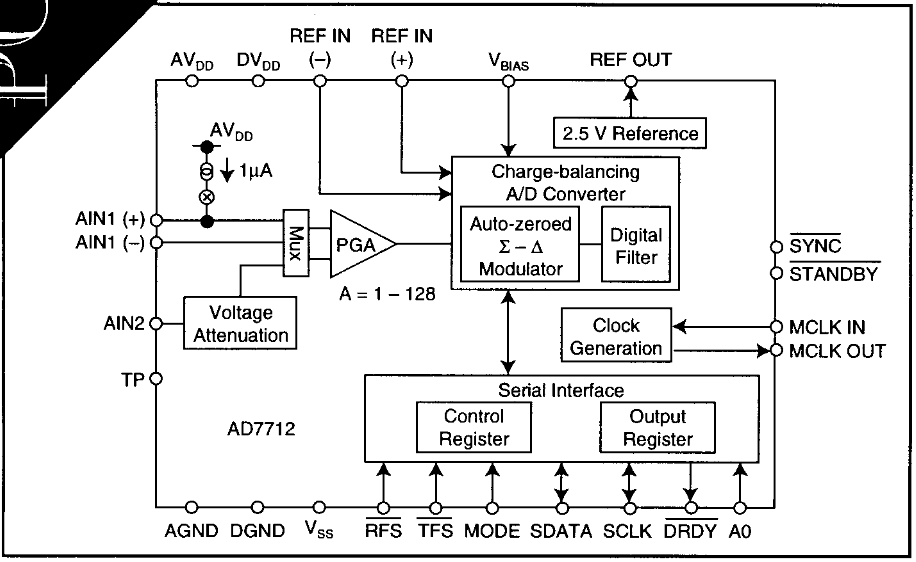

***Figure 4:** The Sigma-Delta converter combines a single-bit modulator and DSP filter to achieve very accurate results. The microcontroller interface is serial, consisting of four control lines in addition to clock and data.*

produces a reasonable span over the full range of the load cell. Also allow for both overrange and negative signals. The application's range of interest may be small, one fourth, one eighth, or even less of the load-cell capacity.

We effectively right-shifted the converter result by two, three, or however many bits represent the unused converter input range. But if the least significant six bits are random noise, no significant data is lost.

We used the extra precision in the ADC to eliminate the necessity for input-amplifier adjustments. We can put away the little screwdrivers and treat the load cell like a real computer peripheral!

Except—the measurement signal is at the load cell. The excitation-voltage and millivolt-level measurement signals must be routed to and from the measurement point, which might be some distance and through unfriendly environments. We don't have digital signals at the load cell to insulate the measurement from these conditions. Instead, load cells use excitation sensing.

To compensate for possible voltage drops in the excitation wiring, load cells often have "sense" wires, connections to the excitation terminals which allow the weighing system to measure the applied excitation voltage at the load cell as well as the signal weight.

Remember, the load cell signal is specified as millivolts per volt of excitation. If the excitation voltage changes due to noise or other causes, voltage drops in the cable, the measurement signal changes in proportion even if the applied weight does not.

How to compensate for excitation variation? The sense signal is brought into the measurement system via high-impedance inputs so that little current flows in the sense leads. The sense signal creates a reference for the ADC, which in turn defines the ADC's unit of measure. Think of the conversion as a digital number times the unit of measure defined by the reference. Changes in the excitation are compensated by equivalent changes in the converter's reference.

At last we have a measured signal in digital form. We've used an instrumentation amplifier, high-precision ADC, and excitation-sensing operational amplifiers for the converter's reference. We have converter data in the onboard microcontroller, extracted with the portion of code detailed in Listing 1 and 2.

ENTER THE PC

Traditionally, the process of weighing could be broken down into four more steps: taring, conversion to units, displaying, and controlling. These steps were hard-coded into the indicator with no opportunity for adjustment.

With the entrance of the PC, these steps have become more modular. The developer can now program them to be automated or user directed.

I've broken the steps into more detail so you can see the PC's new, more active role:

- take out the tare

The weighing system usually starts at some nonzero weight: a platform or the load cell's mechanical mount, the box or carton for the item to be weighed, or the first ingredient in a two-ingredient recipe.

The measurement involves taking data from the load cell and storing and computing a tare or zero weight which is subtracted from the gross measurement to obtain the net measurement.

When can the tare be measured? Hav-

***Listing 1:** This subroutine reads from the AD7712, moving three bytes to the address in pointer r0. Note the signal names correspond to pin designators on the AD7712 (see Figure 4). Control signals select the source of the data, synchronize the devices, and clock the data.*

```
; serin()
        .set   .serin,h'08eb

; AD7710 Interface
        .equ   P4,h'c0
        .equ   P4.0,h'c0
        .equ   P4.1,h'c1
        .equ   P4.2,h'c2
        .equ   P4.3,h'c3
        .equ   P4.4,h'c4
        .equ   P4.5,h'c5
        .equ   RFS,P4.0
        .equ   TFS,P4.1
        .equ   DRDY,P4.2
        .equ   A0,P4.3
        .equ   SCLK,P4.4
        .equ   SDATA,P4.5
; operates the AD7712 in external clocking mode, mode=0
; uses r0 as pointer to data memory; r2,r3 as loop counters
; push/pop accumulator
; returns with data in @r0+2, @r0+1 @r0   24 bits
; data transfers MSB first

        .org   .serin
```

(continued)

Listing 1: cintinued

```
serin:
        clr   sclk
        setb  sdata          ; enable data line
        mov   a, r0
        push  acc
        inc   r0
        inc   r0
        mov   r1, #3         ;load byte counter

wait:   jb    drdy,wait      ;make sure data ready is low
        clr   rfs            ;enable
mvbyte: mov   r2,#8          ;load bit counter
mvbit:  mov   c,sdata        ;move data from port
        rlc   a              ;build byte in accumulator
        setb  sclk           ;cycle clock
        clr   sclk
        djnz  r2,mvbit       ;test bit cntr, jump back if not done
        mov   @r0,a          ;store byte
        dec   r0
        djnz  r1,mvbyte      ;test byte cntr, jump back if not done
        setb  rfs
        pop   acc
        ret
```

Listing 2: This subroutine writes to the AD7712, moving three bytes from the address in pointer r0. The write destination is determined by A0, either control register or calibration register.

```
; serout()
        .set  .serout,h'090f

; AD7710 Interface
        .equ  P4,h'c0
        .equ  P4.0,h'c0
        .equ  P4.1,h'c1
        .equ  P4.2,h'c2
        .equ  P4.3,h'c3
        .equ  P4.4,h'c4
        .equ  P4.5,h'c5
        .equ  RFS,P4.0
        .equ  TFS,P4.1
        .equ  DRDY,P4.2
        .equ  A0,P4.3
        .equ  SCLK,P4.4
        .equ  SDATA,P4.5

;r1 is used as a byte counter, 3 bytes in each transfer
;r2 is used as a bit counter
;r0 is loaded with a pointer to the output data prior to
;calling the serout function
;data is tranferred MSB first

        .org  .serout
serout:
        mov   r1,#3          ;load byte counter
        inc   r0             ;point to msb
        inc   r0
        clr   sclk
        clr   tfs            ;enable interface
mvbyte: mov   r2,#8          ;load bit counter
        mov   a,@r0          ;load first byte of data
mvbit:  rlc   a              ;move msb to the carry bit
        mov   sdata,c        ;mov the carry bit to the port
        nop                  ;delay one cycle to avoid race
        setb  sclk           ;cycle the clock
        clr   sclk
        djnz  r2,mvbit       ;test bit counter, jump if not done
        dec   r0             ;decrement data pointer
        djnz  r1,mvbyte      ;test byte counter, jump if not done
        setb  tfs            ;disable interface
        ret
```

ing the user push a button when the platform is empty works for a microcontroller. Just route the button to a port pin and set up a loop to interrogate the button.

However, if the PC generates a tare command, the system has more flexibility. Tare could come from a PC procedure, the user interface, or any other means.

Here, we encounter the first need for bidirectional data exchange. The PC knows or can easily find out when to tare.

- convert to engineering units

The load-cell output is a voltage. The weight or force is in some other unit, perhaps pounds or kilograms. The digital indicator multiplies to generate a weight in units of interest. But how are the calibration factors used in the derived calculation, and where are they stored?

The calibration procedure of an indicator–load-cell combination specifies the calibration constants for calculating the measurement from raw data. The PC can be programmed to take an operator through the procedure and then store the results, download the data on application startup, or take the raw data and compute the weight itself. Suddenly, we have several options.

- display the data

Display is easy for a microcontroller as long as the display is on the board or in the same box with the microcontroller. Many varied digital displays are available with easy-to-program display-driver chips.

Displaying data remotely is a different problem. Let the PC interpret weighing data, add to it date, time, batch number, or whatever else is relevant, and send the data for display, print, or incorporation into a report at another location.

- take a control action

Often, a measurement leads to process action. For example, if the weight is not between 1 and 2, reject mechanism. This kind of control action is easy for micros.

But, where do the setpoint values come from? Are they programmed, stored in a file, or downloaded from the network?

Notice that as we come further down the list, the task extends beyond the traditional microcontroller. The application measures more than just a voltage generated by the load cell. The PC has readily available resources to augment the application while the microcontroller does not.

AN OFF-THE-SHELF SOLUTION

Scanning Devices manufactures a PC/104-compliant load-cell controller that takes advantage of these developments. Model 754PLC3 uses an 8051-derivative microcontroller with a handshaking port as the PC bus interface. A bidirectional 8-bit port for transfers data to and from the PC bus with a 2-bit port-control status register.

The load-cell controller is an 8-bit, stackthrough PC/104 module. The module provides excitation, sense inputs, and measurement inputs for a single load cell. The key to measurement is a high-precision ADC, Analog Devices AD7712. It also offers four digital inputs and four digital outputs for process monitoring and control.

But what takes it a step ahead of the system with a digital indicator installed on the serial port is the rich set of data-exchange and control transactions possible between the PC and load-cell controller. These transactions let the PC control the microcontroller and converter operations so the developer can maximize resources for each part of the application.

In Photo 3, you see the main-menu screen of the Lancelot PC-based demo program. The screen shows buttons for calibration and filter setup, setpoint control, mode selection, digital input and output control, and measurement data transfer and display. All of these can be under user or program control.

Scanning Devices' software runs in the onboard microcontroller to set up and control the ADC, implement postconversion digital filtering, compute weight from raw conversion data, compare weight to four setpoints, monitor four digital inputs, and set four digital outputs. This capability and an onboard EEPROM for permanent storage lets the indicator module run independent of the PC for much of the application.

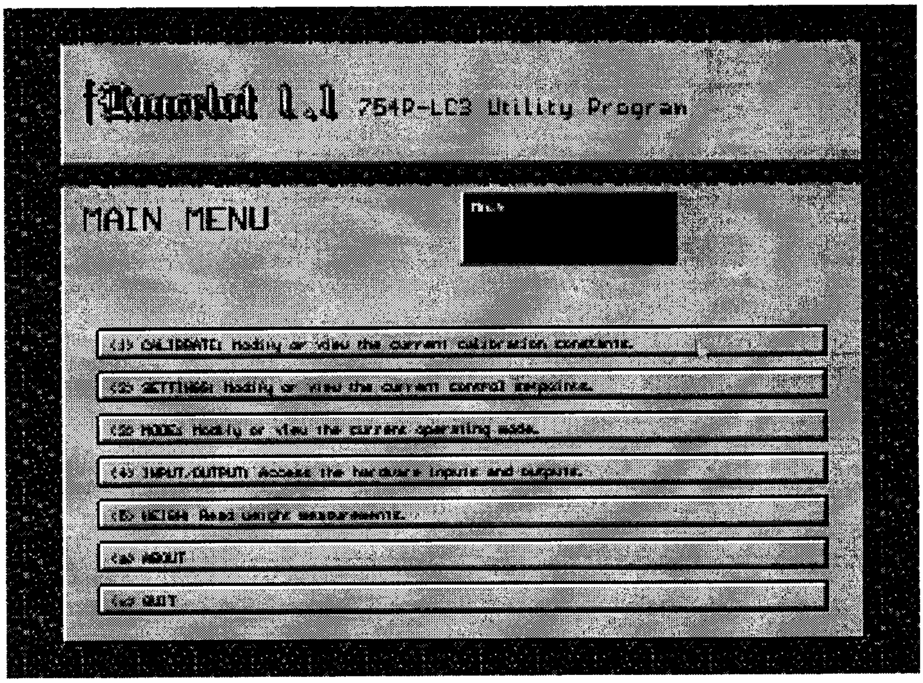

***Photo 3:** Scanning Devices' Lancelot Main Menu shows the functions available to the user on one screen—calibration, setpoint entry, mode selection, digital input and output control, and weight measurement. This C++ demo serves as both a diagnostic installation aid for the controller and a programming tutorial on application techniques. Instead of being user- or menu-driven, these functions are easily implemented as real time.*

However, when the PC writes a command to the port address, it's a different story. The 754P-LC3 is passive regarding the PC bus. All transactions are initiated by the PC. When the PC writes a character to the indicator module's address, the micro interprets the command and either accepts data from or transfers it to the PC.

GetParameters	transmit the current values of the four process setpoints
SetParameters	receive new values for the four process setpoints, store in EEPROM, and use
GetCalibration	transmit the current value of the load-cell calibration constants
SetCalibration	receive new values for the load-cell calibration constants, store in EEPROM, and use
GetMode	transmit the current operating mode
SetMode	receive a new value for the operating mode, store in EPROM, and use
GetWeight	transmit net weight data in engineering units
GetData	transmit raw weight data (conversion output)
GetIOStates	transmit the state of the indicator module's four digital inputs
SetIOStates	receive new values for the indicator module's four digital outputs and set outputs
GetFilter	transmit the characteristics of the onboard digital filter
SetFilter	receive new values for the onboard digital filter
Reset	restart with parameters in EEPROM

***Table 1:** These C++ functions, which are easily integrated into PC applications, give the user control over weighing operations. User-written functions in the language of choice can achieve the same results.*

Table 1 shows the thirteen transactions that are currently implemented, all related to setup and data transfers. Compare this level of PC communications and control to the one-way serial output of traditional indicators. The commands in Table 1 refer to the data variables stored and used in the indicator module (see Table 2).

By making the raw data and weight available, users can calculate weight, introduce filters, and compare more setpoints. By providing access to indicator-module digital inputs and control over the outputs, PC users control the whole process.

So, while the module indicator may run independently of the PC, it may also be configured as a special-purpose data-acquisition module with calculations and logic done completely in the PC. The choice is up to the system designer.

WHY PC/104?

Because of the potential for industrial packaging. Scanning Devices is involved in industrial weighing for process control. Such weighing occurs under the material, not on the desktop. Dirt, moisture, electrical noise, and other irritants make short life of unprotected desktop PCs. Protecting desktop PCs is costly.

The PC/104 consortium has specified a small form factor with relaxed bus-drive

Scanning Devices, Inc.

Sensors, Instruments, Displays, and Controls for Industrial Automation

Scanning Devices, Inc. designs and manufactures sensors, instruments, displays, and controls for industrial automation. The company specializes in electro-optics and microcontroller-based instrumentation.

The following load cell controllers and related software products for PC based weighing systems are currently available from Scanning Devices, Inc. All products are designed to work with resistance Wheatstone bridge transducers such as strain gage load cells and pressure transducers.

754P-LC3 - PC/104 compliant load cell controller. Single load cell input channel with excitation for up to four 350 ohm transducers connected in parallel. Four digital inputs and four digital outputs for control.

754P-LC4 - PC/104 compliant load cell controller. Four independent load cell input channels. Four digital inputs and four digital outputs for control.

754I-SA1- ISA compliant load cell controller for IBM and IBM compatible desktop PCs. Requires one 8 bit slot in PC backplane. Single load cell input channel with excitation for up to four 350 ohm transducers connected in parallel. Four digital inputs and four digital outputs for control.

754P-SW3 - Utility and demo software package for 754P-LC3. Provides C++functions to control the module and diagnostic software. Runs under MS-DOS. One-time purchase with license for incorporation and use in OEM products.

754P-SW4 - Utility and demo software package for 754P-LC4. Provides C++functions to control the module and diagnostic software. Runs under MS-DOS. One-time purchase with license for incorporation and use in OEM products.

754W-IN1 - Windows software for 754I-SA1. Provides a Windows interface in the Accessories Group. Runs in Windows 3.1 and Windows 95.

For further information on these products or customized microcontroller instruments and displays, contact Scanning Devices at the address below.

Scanning
Devices

111 Terrace Hall Ave. Burlington, MA 01803
781-272-5135 www.scanningdevicesinc.com