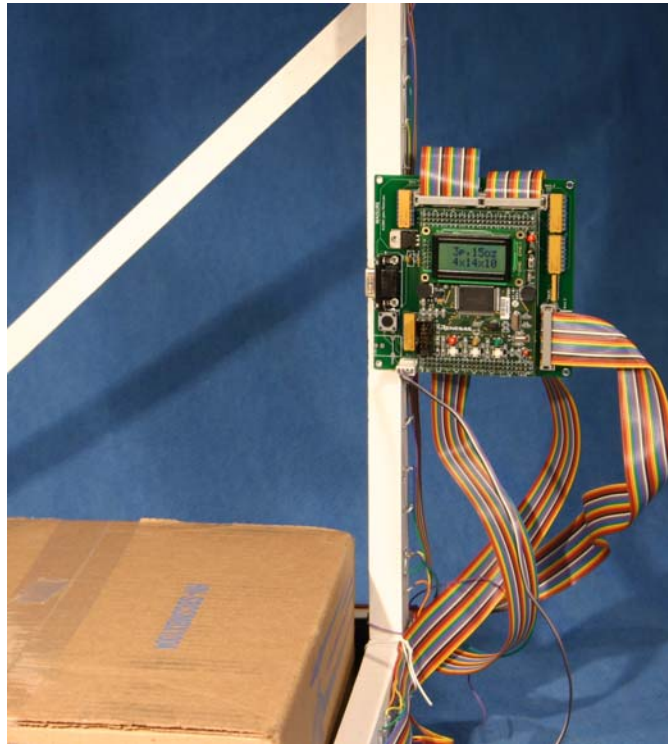


# Weasure

## A Scale that Weighs and Measures



©2005 John W. Peterson  
[www.saccade.com](http://www.saccade.com)

### Introduction

Electronic scales are common today, and many of them easily interface to PCs. However, for most commercial shipping services, the weight alone isn't enough, they also want to know the physical dimensions of the package.

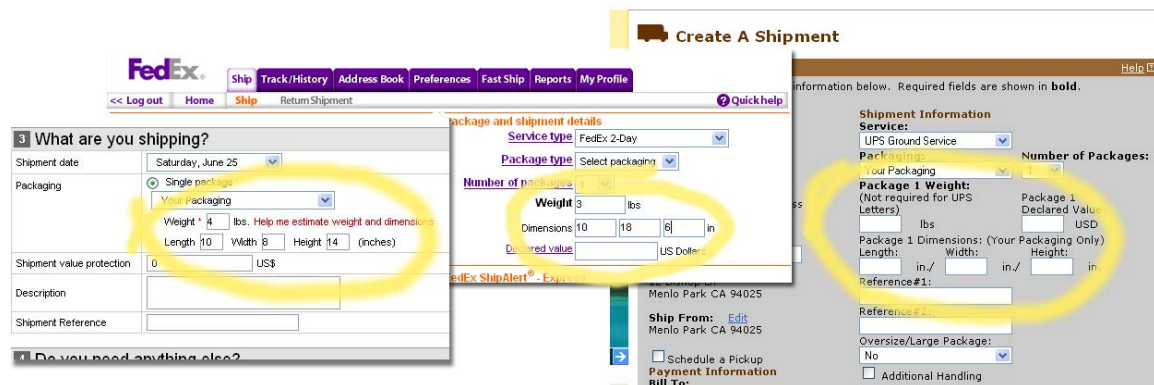


Figure 1 - Screen shots from the DHL, FedEx & UPS web sites

While you can get the weight directly from the scale, measurements are still done by hand with a measuring tape or yardstick. In a high volume shipping room with many different package sizes, measuring and recording the package dimensions adds several steps to the shipping process.

*Weasure* offers a “one-touch” solution to this problem. Simply place the package on *Weasure* and both the weight and dimensions are instantly displayed. By connecting *Weasure* to a PC’s serial port, this data is uploaded directly to the PC, where a shipping application can send the measurements directly to the shipping company’s web site, further streamlining the process.

### *How it Works*

Because *Weasure* is targeted at parcel shipping, there are a couple simplifying factors leveraged in its design. First, shipping companies don’t require very precise measurement data; weights are rounded up the nearest pound, and sizes to the nearest inch. Second, almost everything shipped comes and goes in the ubiquitous cardboard box; so measuring boxes lets you measure most anything that’s shipped.

### Weight measurement

To measure the weight, I started with an existing postal scale. I found a new DigiWeigh DW-36XP on Ebay for about \$10. The scale is very basic, with a simple LCD digital readout. I took apart the scale as soon as it arrived, and started probing for a way to adapt it to this project.

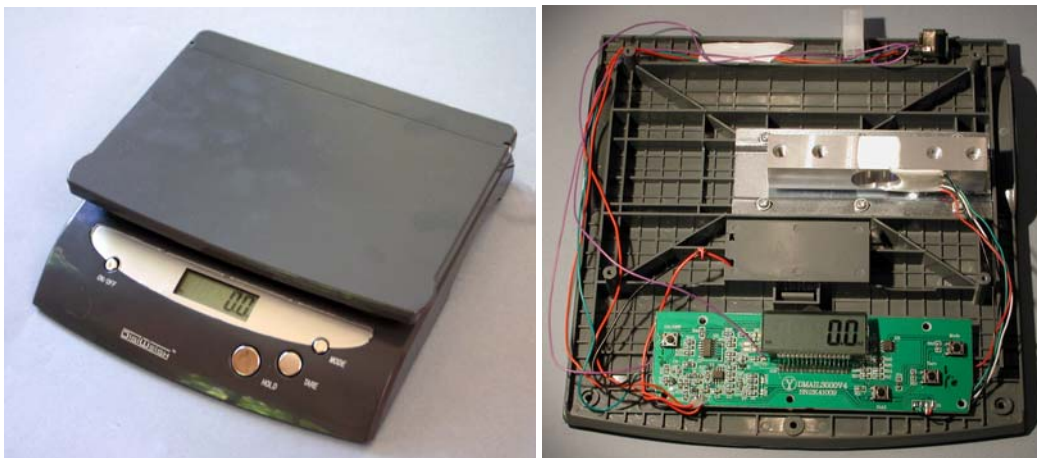


Figure 2 - DigiWeigh scale, opened to reveal strain gage

The measuring component of the scale is a strain gage. This is a metal bar with a flex point (the hole). A resistive material is coated on the bar above and below the hole. As a load is applied, the bar flexes, causing small changes in the resistive characteristics of the bar's coating.

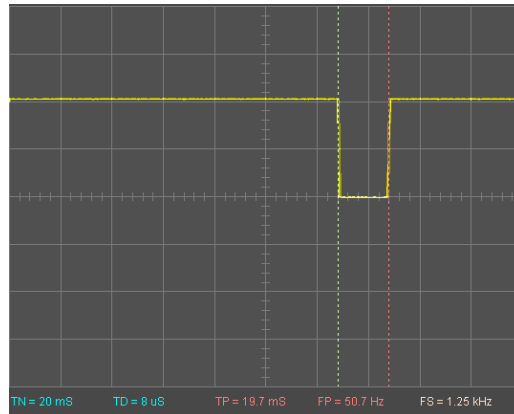


Figure 3 - Scale "T3" output with no load

Since reading the strain gage directly requires some carefully calibrated analog electronics, I looked for a higher-level output I could take from the scale to use for Weasure. The scale does not provide any sort of documented digital output, and uses a proprietary MPU hidden under an epoxy blob. I expected to find an analog value that changed with the force applied. Some probing with an oscilloscope turned up something just as useful though; a PWM (pulse-width modulated) signal with the "low" period directly proportional to the applied weight (this is conveniently labeled "T3" on the DigiWeigh's circuit board). The low pulse is about 20ms wide with no load, and increases by about 2ms per pound of force. This signal is already at the same 5-volt levels used by the M16C and works perfectly with the M16C's built-in PWM timer.

### Size measurement

Since we're measuring cardboard boxes to the nearest inch; it's feasible to have a sensor for each measurement point. To measure the size of the box, a row of inexpensive photocell light sensors spaced one inch apart is placed along the axis for height, width, and depth. When a box is placed on Weasure, examining which photocells show a "dark" reading determines the size.

This scheme works particularly well with the M16C's generous helping of parallel I/O. Even by using only "leftover" I/O ports not already used by the SKP board or the other Weasure functions, there were 62 I/O lines left. This is enough to measure boxes up to 24x20x19" in size.

The photocells I used have a resistance of about 2K $\Omega$  or less when well illuminated, and at least 20K $\Omega$  or more when dark. By connecting the photocell

in series with 10K $\Omega$  pull-up, the photocell pulls the signal line to ground when it's illuminated (generating a "0") and let it go to "1" when it's covered up by the box. These signals are connected directly to the M16C input lines.



Figure 4 - Photosensor schematic, photosensors on the measurement frame

## Construction

### Motherboard design

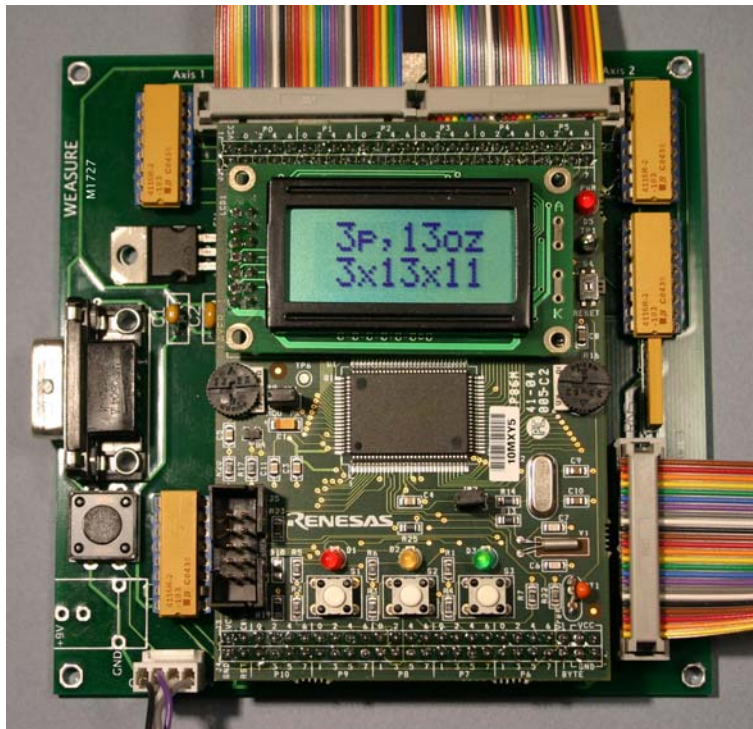
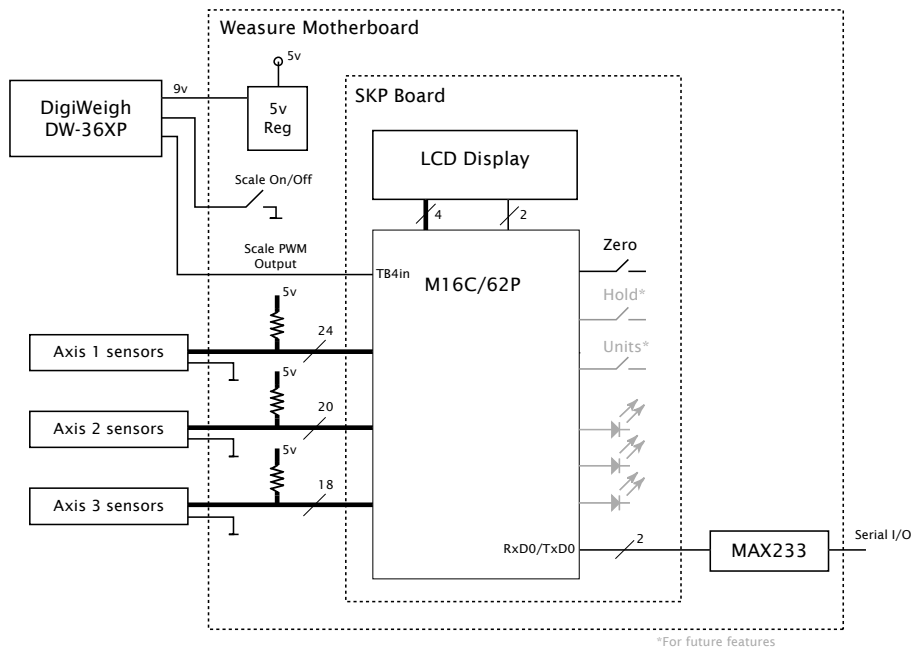


Figure 5 - Weasure board with SKP attached (the MAX233 is under the SKP)

The Weasure Prototype uses the SKP16C62P evaluation board “as is”, since the peripherals on it (in particular the display, buttons and LEDs) are useful for development and the finished product.

To interface the board with the scale, I designed a motherboard PCB. The motherboard’s sockets line up with the interface pins on the SKP board. This board has the resistor networks and connectors for the size measurement photocells, a connector for the scale, and a MAX233 RS-232 serial interface chip to provide the proper signal levels for talking to a PC’s serial port.



**Figure 6 - Weasure Block Diagram**

The DigiWeigh scale runs off of a 9v battery or an optional wall-wart power supply. The 9v supply from the scale is connected to the Weasure motherboard, where a 7805 regulator supplies the 5v supply for the SKP board, the MAX233 and the resistor networks. The PWM output from the scale is connected to the Timer B4 input pin of the M16C/62P.

Since the DigiWeigh scale is obscured by the Weasure measurement platform, the controls and display on its front panel are replicated on the Weasure board. A switch on the motherboard duplicates the On/Off switch, and the “Zero” function is duplicated by S1 on the SKP board. The zero button resets the scale’s zero point. This, like the other functions on the DigiWeigh’s front panel, is easily duplicated in software.

The DigiWeigh scale was modified with a connector for the 9v supply, on/off and PWM signal outputs. I tack-soldered wires onto the scale's control board and routed them to the connector. Conveniently, a cutout on the back of the scale was easily adapted to a connector for attaching it to the Weasure motherboard.

### Measuring frame

The measuring frame for this prototype is constructed out of wood. Holes are drilled at 1" intervals along the measuring surfaces at a angle up towards the light. These surfaces are painted matte black to minimize stray reflections into the photocells.

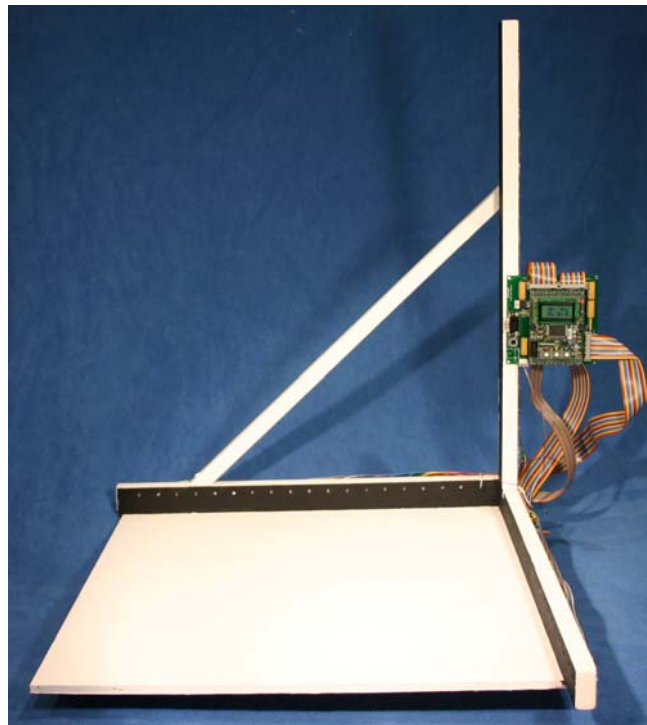


Figure 7 - Measuring frame

One side of each of the photocells is connected to a bare wire to ground. The other side is connected to a special ribbon cable that attaches to the Weasure motherboard. In retrospect, creating circuit boards for mounting the photocells would have been a major timesaver. Manually wiring and soldering all those connections took the better part of a day!

### *Firmware*

The firmware for Weasure is developed with the excellent High-performance Embedded Workshop (HEW) supplied with the SKP. Several code samples from

the kit proved very useful, in particular code for the serial I/O and LCD display is used directly.

The Weasure firmware first initializes the MPU, the I/O ports used, the UART, and the timers. Then it enters the main loop that updates the display based on input from the size and weight measurement.

The size measurement works with three arrays of port bit IDs, one for each measurement axis. Each array slot has the port number in the top four bits, and the bit of that port in the lower three bits. These slots correspond to the photocells on the measurement axis.

To measure a length, the firmware starts at the end of the array and works its way down, examining the value on the port's bit. When it finds a zero value (indicating dark), the array position indicates the length for that axis.

An ISR for Timer B4 is used to measure the pulse width of the scale's PWM output. The timer is configured to invoke the ISR every time the input changes state. By recording the timer's count on positive transitions, the weight dependent pulse width is measured. This ISR averages every four measurements to filter the values and give a stable result. When an averaged result is complete, a flag is set indicating the weight reading should be updated in the LCD display. If no signal transitions are found (i.e., the timer overflows) then the scale is presumed to be off and "Off" is displayed.

Finally, another ISR listens for data on the serial port. This feeds back the Weasure data according to the following simple protocol:

Command byte	Response	Description
s	HH,WW,DD	Returns height, width, depth, in inches
w	PP,OO	Returns weight in lbs and ounces
c	PP,OO,CCCCC	Returns weight and raw PWM count

This protocol makes it easy for a PC to query the Weasure for automated shipping applications.

The excellent Renesas HEW tools made the software development very straightforward. It was easy to start with the supplied examples and modify them to fit this application. The interactive debugger worked great too. I had no difficulties implementing the firmware.

### *Conclusion*

Weasure provides a complete, “one touch” solution for getting all of the measurement information about a package required to ship it. By connecting it to a PC, it’s possible to create shipping applications that streamline shipping by eliminating a separate measurement step.

The M16C/62P was an excellent fit for this application. It has plenty of I/O to implement the photocell measurement scheme without additional hardware, and all of the other peripherals (timers, serial I/O) are available directly on the chip.