FEATURE ARTICLE

Design2K Contest WINNER

Paul Kiedrowski

The QuizWiz

A Hand-Held Scoring Device

When it comes to making the grade, Paul's Design2K project passed the test with flying colors and won a grand prize. With the QuizWiz, a teacher has a simple and cost-effective tool that can reduce the time spent grading quiz forms.

For automatic scoring of multiple choice tests, many schools use a commercially available system based on a desktop card reader machine, which requires that students mark their answers on preprinted forms of specific size and layout. This method is expensive because of the equipment and score cards, therefore, usually it's used only for critical testing.

In most cases, because only one centralized scoring machine is available to teachers, it is not located in the classroom where it would offer the most convenience. Perhaps more importantly, the most useful time to evaluate test results is immediately after a test so that feedback can be given and the missed questions discussed promptly. This is especially desirable for periodic quizzes where the intent is to allow the teacher to quickly gauge the classroom's learning progress. What is needed is a better, lower cost, convenient way of quickly scoring quizzes.

INTRODUCING THE QUIZWIZ

To answer these needs, I developed a hand-held scoring device based on a Philips 8-bit microprocessor. The 87LPC764 is a new 20-pin offering that combines fast speed, 8051 code compatibility, and low cost. This processor is ideally suited for the scoring device project because of its 4-KB code space, power-saving modes, serial port, and remaining 16 I/O pins. My objective was to create a device that is simple to operate and affordable enough that every teacher could own one (see Photo 1).

The QuizWiz has many features that make the teacher's ability to score multiple-choice quizzes fast and easy. It reduces scoring time to only 10 to 15 s per page. It is capable of scoring tests printed on standard paper without preprinted forms, using a word-processing template.

The QuizWiz does not require machine-assisted paper handling mechanisms. Also it operates on three AAA batteries. The device is inexpensive ($30 for parts) and measures only 6″ × 1.7″ × 1″. Simple to learn, the new teacher's aid requires only two buttons to operate.

A 2 × 8 LCD shows the status and results, and a buzzer provides distinctive audio feedback. Automatic power

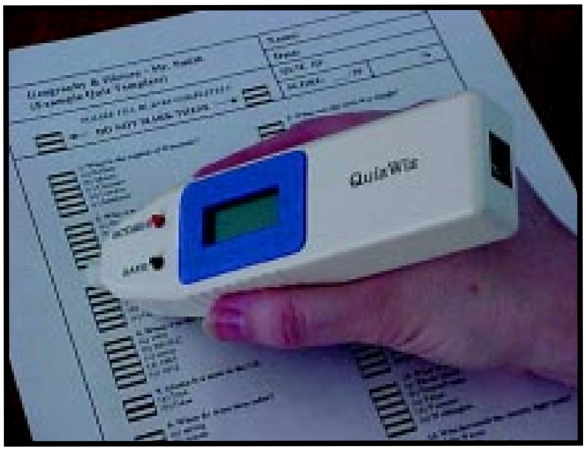

Photo 1—*The prototype QuizWiz sports a 2 × 8 character LCD display and just two operating switches labeled "scores" and "save." The quiz format can be seen here, requiring a starting sync section, dark areas between answer selections, and a minimum amount of white space between questions. A mini-DIN connector is on one end to provide an optional serial port interface.*

shutdown during idle periods provides long life. The flexible quiz format allows multiple columns and pages.

For convenience, there is temporary memory storage of results during power shutdowns. Totals, per question and per quiz, are available. The QuizWiz can handle eight choices per question, four columns of questions, and 32 questions per quiz. The processor provides an RS-232 serial connection for uploading results to a PC, which allows tracking of which questions were missed per student.

CIRCUIT DESCRIPTION

Figure 1 shows the circuitry has been partitioned by the two chassis sections. A PCB in the lower half contains most of the components, whereas the sensor tip, LCD, and battery compartment are in the upper half. The processor was DIP-socketed for the development phase (see Photo 2).

A single reflective optosensor was chosen to perform the scanning detection process, with an optimum sensor-to-paper distance of 1 mm. To preserve battery power, the optosensor LED is only activated when the QuizWiz is pressed against the paper, depressing a mechanical switch located in the tip. An alternate scheme that I initially considered required two sensors, the second one used for scanning a parallel column of markings intended only to synchronize the scan position. The additional LED would have significantly increased the power consumption, however.

Normal battery operating current is approximately 25 mA when all circuits are operating, 15 mA when not scanning, and 20 µA during shutdown. Using three AAA batteries in series with a typical capacity of 1000 mAh, the teacher can score approximately 100 quizzes for 30 students before replacing the batteries.

The QuizWiz uses a simple three-chip design consisting of the processor, a 5-V DC/DC converter, and an RS-232 three-wire interface. The 87LPC764 is a good match for the required features, because all of its pins and most of its features are used in this project. The only features not used are the I²C interface and analog comparators. To minimize cost further, no external crystal is required, because the processor conveniently includes an internal 6-MHz RC oscillator.

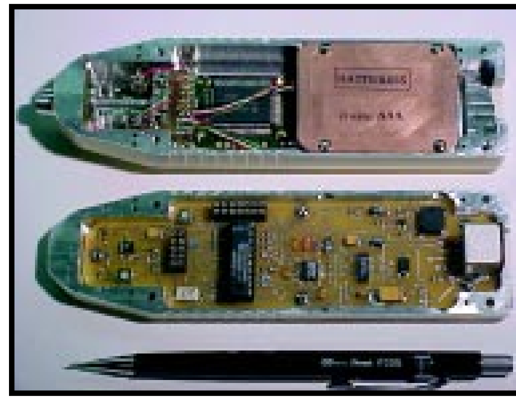

Photo 2—*A design-for-manufacturing approach was taken for the construction of the aluminum-chassis prototype. The main two-sided PCB has ample room for parts, mainly because of the housing size needed to fit the AAA battery pack and LCD display.*

OPERATION

Photo 1 exemplifies how the quiz format is arranged to aid in the scanning process. The format requires questions to be arranged as usual in columns, with three additional constraints. These constraints include a short sync marking at the top of each column to aid in establishing the scan rate, a black or darkened area between each of the answer selections, and a minimum amount of white space between each question to distinguish them.

To use the device, teachers simply place the QuizWiz on the paper and slide it down along the check boxes used for multiple-choice selection by the students. The

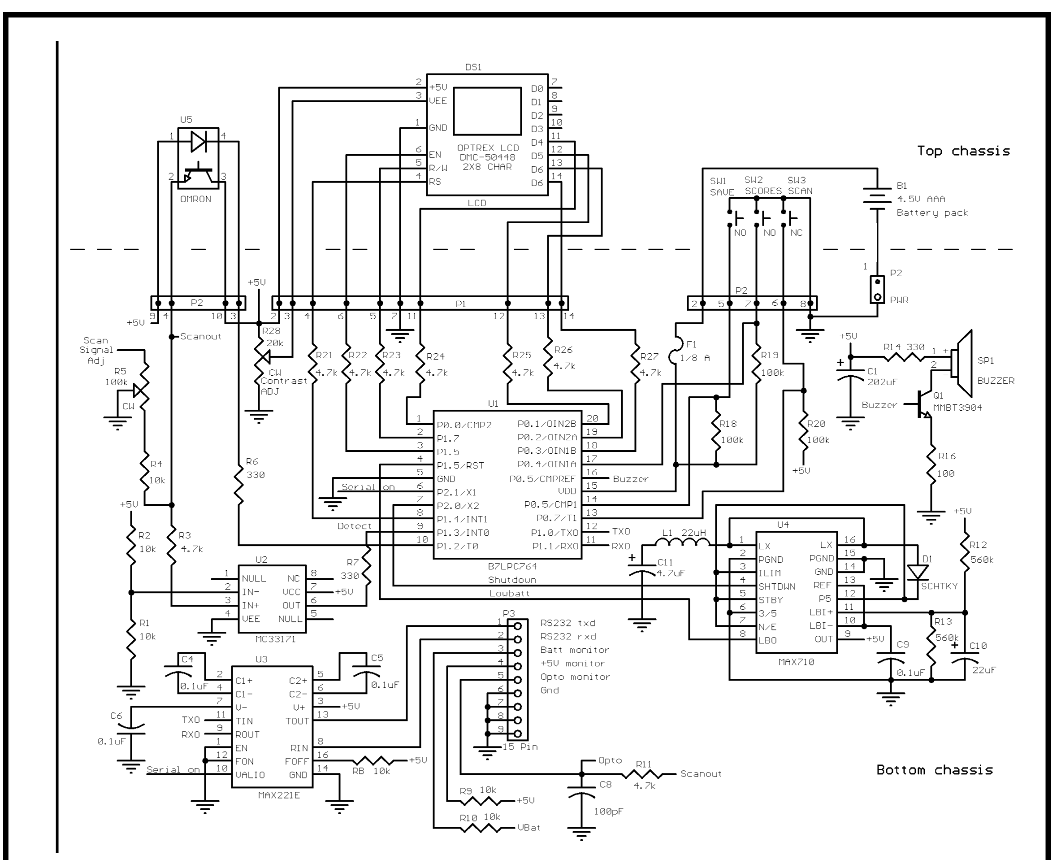

Figure 1—*Here you can see the 87LPC764 processor, MAX221 RS-232 interface, and MAX710 DC/DC converter. The device supplies 5 V to the LCD, which uses a 4-bit interface to save I/O pins.*

device scans the selections made and compares the results to the correct answers that were previously scanned in via a master quiz. After each quiz is scanned, the score is displayed as both the number of correct answers and a percentage. The teacher also can select the current cumulative scoring statistic for each question or the overall quiz, so the appropriate review focus can be established.

As stated earlier, the QuizWiz includes a sounding device to provide audio feedback. A short beep will be heard if a column is scanned properly, a longer beep when all columns of the quiz are correctly scanned, or various alarm beeps if an error is detected. After each successful quiz scan, the teacher presses Save or scans again.

At any time after a master quiz is scanned, the teacher can press the scores button for results of each question, as well as for other information. Simply continue to press the button to advance to the results for the next question. Note that holding down the scores button for at least 1 s resets the display and shows additional pertinent data.

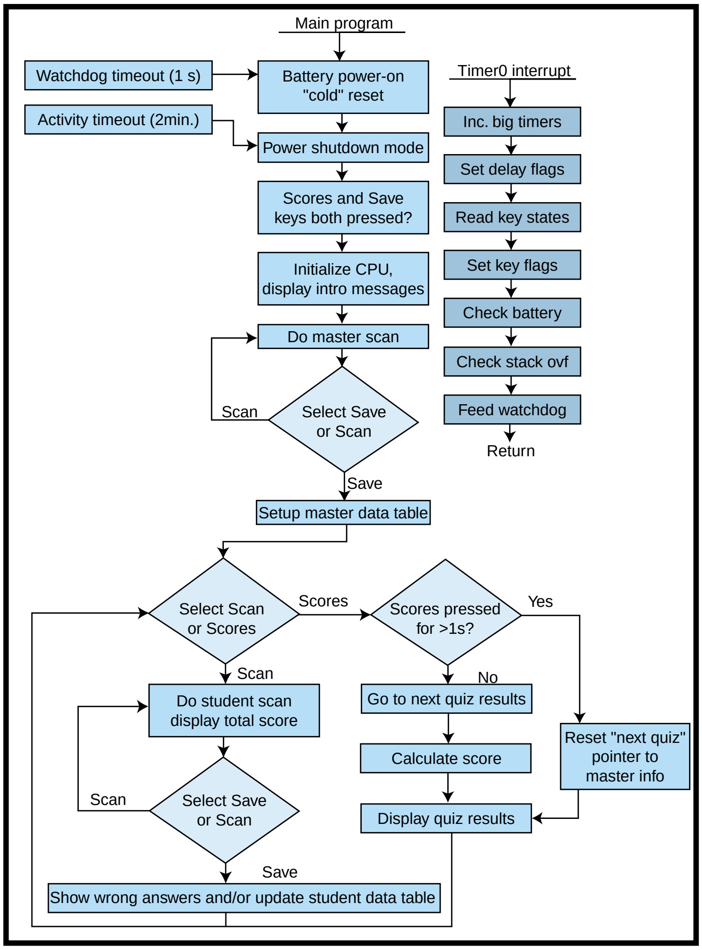

Figure 2—*In the operation flow diagram, notice that the scan switch is depressed whenever the tip is pressed to the paper. This approach minimizes power consumption by allowing the device to use only a single optosensor.*

Results are stored in memory unless battery power is removed or both the scores and save buttons are held down for more than 5 s. Then, the QuizWiz will erase all results and start over, directing the teacher that the next quiz scanned must be the master.

The QuizWiz goes into low-power shutdown mode after 2 min. of inactivity and is returned to normal operation when both scores and save are pressed together. A flow diagram of the operation is shown in Figure 2.

For access to quiz scoring details as they are scanned, you may connect a PC to the QuizWiz using a standard RS-232 serial port connection at 9600 bps. The QuizWiz automatically detects the presence of the serial port connection, and power usage is reduced when not connected.

ADDITIONAL FEATURES

There is no main switch needed to disconnect electronics from the battery, and existing data is preserved in RAM during shutdown. Although the processor has keyboard interrupt capability, a polling technique is used instead, because a main ISR timing loop is required. The circuit contains provisions to detect a low-battery condition and displays an appropriate message, then shuts down if necessary.

The device must not only correctly identify marked answers, but also be capable of reading multiple columns of questions and ensure that the total question count is the same as that of the master quiz.

Because the LCD used was chosen for its minimal cost and small size, it was important to carefully choose the formatting of the displayed results. For example, pressing the scores button presents the following sequence:

SCORING RESULTS:
TOTAL # QUES = XX
TOTAL # COLMN = X
TOTAL # QUIZ = XX
AVERAGE SCOR = XXX
QUES #01 SCOR = XXX
QUES #02 SCOR = XXX
(And so on, until all questions are listed.)
QUIZ #01, SCOR = XXX
QUIZ #02, SCOR = XXX
(And so on, until all quizzes are reported.)
NO MORE QUIZZES
(Then it starts again.)

HARDWARE ANALYSIS

The microprocessor has the capability to monitor keyboard interrupts, but cannot distinguish key states or releases. The keyboard interrupt is level sensitive and will continue generating interrupts until the key is released. Therefore, polling must be used to determine when multiple keys are pressed as well as for debouncing. For these reasons, it is simpler to use the Timer0 interrupt to monitor key states and actions, and only enable the keyboard interrupts for wake up from power shutdown. The nominal Timer0 interrupt rate is:

$$\frac{\left(\frac{6\text{ MHz}}{6}\right)}{65536} = 15.26\text{ Hz, or } 65.5\text{ms}$$

The behavior of the device during a scan is detailed in Figure 3. Assuming a 10″ column of 0.1″ markings per answer (dark plus light), scanned in 0.5 s, the minimum value of TSYNC is:

$$\frac{0.1}{10 \times 0.5} = 5\text{ ms}$$

Assuming markings are .25 and the column scanned in 3 s, the maximum TSYNC is:

$$\frac{0.25}{10 \times 3} = 75\text{ ms}$$

So, during scans, if you use Timer1 in 16-bit mode, it has a range of 0 to

65.5 µs, which is close enough for starters. You can improve the range by independently measuring TDARK and TLIGHT. To conserve memory usage, if you only store the upper eight bits of Timer1 when measuring, the precision will be 256 µs. That's good enough, because TDARK won't be less then 2 µs, and this error does not accumulate during the scan.

The Timer0 interrupt latency is not critical because the fastest operation you are trying to measure is TDARK to within 1/10 of its period, or approximately:

$$0.1 \times \left(\frac{5\text{ms}}{2}\right) = 250\mu\text{s}.$$

Therefore, you can either disable timer0 interrupts during this critical time (with an insignificant effect) or try to keep the timer0 ISR under 250 µs. If enabled, Timer1 should be set to a higher priority than Timer0.

The "big timers" are all 8-bit counts (reset at different times) of Timer0 interrupts, so the maximum range is 255 × 65.5 ms = 16.7 s. There is a 3-bit additional counter used for the activity timeout, which is 2 min.

The scan key is usually closed, so it draws current normally. It is biased through the 5-V line that is turned off during shutdown, and its keyboard interrupt doesn't need to be enabled for wake up. The Save and Scores keys are biased to battery because they need to be monitored for wake up detection. Either can cause wake up, but polling must be done to ensure that both are pressed. This can be done without activating the 5-V line.

In addition, this can help ignore false key presses. Key press states are determined when two successive key state reads are the same in the Timer0 ISR. Because you need to determine how long a key is pressed as well as if

Listing 1—*Take a look at how the ISR functions.*

```
Timer0_ISR:
;ISR must clear key timers & errorflags when shutdown bit
  is cleared
disable all interrupts (assumes individual int enables
  set elsewhere)
        push A, R0, etc
        clear timer0 interrupt
        increment all active timers
        set timer overflow flags
        read keystates
        set key flags
        check serial port, enable int, etc if enabled
        setup serial port:
ISR detects that serial cable is attached for 2 tmr0 ints
        if so set bit SerialDetect
        then check SerialEn bit
        if enabled then setup port and set bit
  SerialActive
        if not enabled then turn off serial port if al-
  ready active
        check battery:
if not LowBatterySignal then
clear LowBattTimer
clear LowBattery flag
break
else inc LowBattTimer
if LowBattTimer > 5 sec then set LowBattery flag
        check stack overflow:
        if StackVerify <> 7Eh then set StackError flag
        feed watchdog
        pop A,R0, etc
        enable all interrupts
        return from int
```

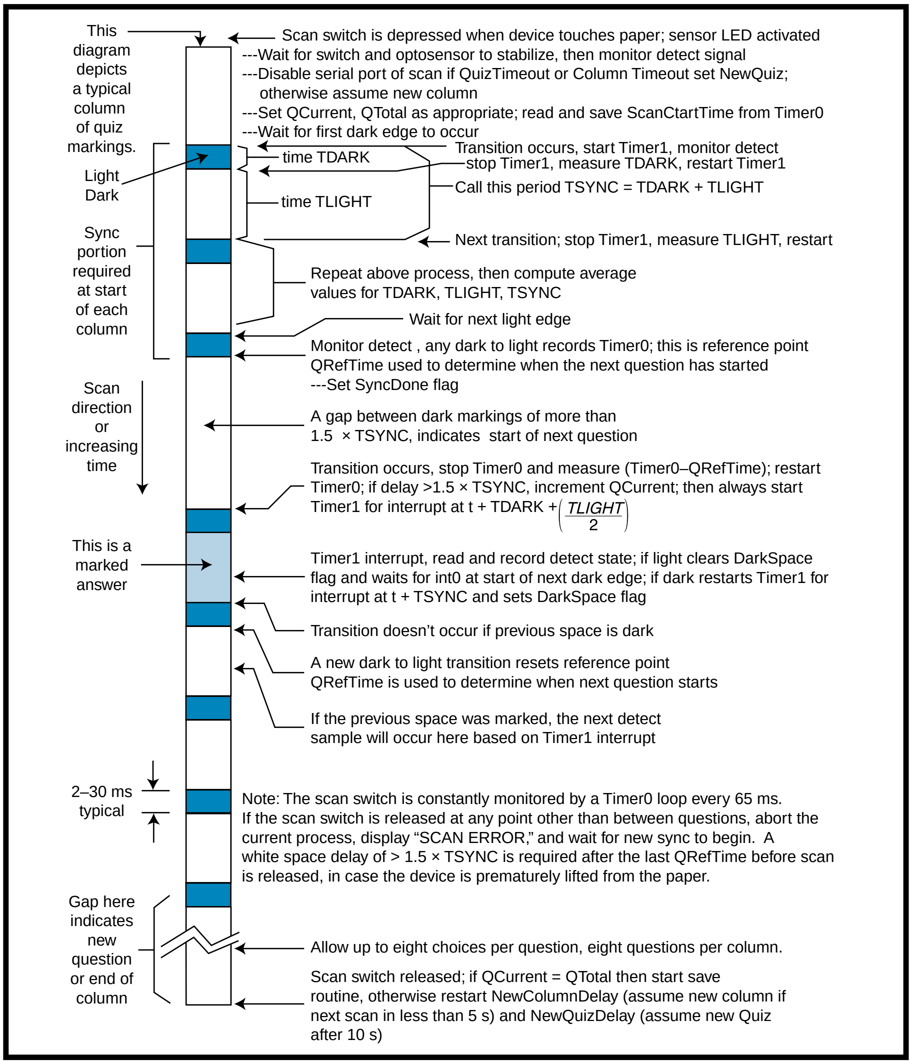

Figure 3—*The scanning process diagram shows how the quiz markings are interpreted and how timing is used to detect a marked answer. The sync portion provides an estimate of the scanning speed.*

more than one is pressed, no action is taken on key presses until after they are released.

The buzzer and sensor LED can be driven directly from the port pins (20 mA maximum), however, the circuit uses a 2N3904 NPN transistor buffer to provide noise isolation.

The battery voltage was selected to be 4.5 V (i.e., three AAA cells in series) because the LCD's high state output voltage must be within 0.5 V of the processor's supply voltage. Using three AAA cells instead of two is allowed because of the height of the LCD and because the 5-V converter has better efficiency there. A standard AAA alkaline battery (0.41″ × 1.75″) is typically rated for approximately 1000 mAh, or 40 h at 25 mA of continuous usage.

By contrast, you could eliminate the converter by using a standard 9-V battery and a linear regulator. However, that would be less efficient. Furthermore, with a capacity of approximately 500 mAh, it would yield only a 20-h lifetime.

By driving the LCD in 4-bit mode instead of 8, you free an additional four I/O lines, which is convenient and allows you to add power shutdown, battery monitor, buzzer drive, and serial port detection features.

The scan switch could be conveniently replaced by another optosensor, but it would draw too much current. Another alternative is to periodically pulse the scan optosensor (i.e., turn on the LED) and check for a high detect level, indicating that it was sensing a reflective surface. Then, if synchronization wasn't found within a prescribed time, you would turn it off again. However, this also would draw

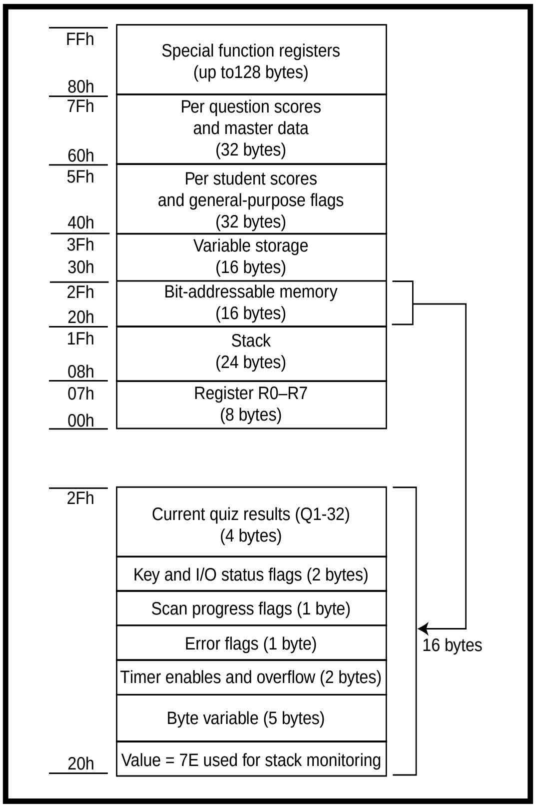

Figure 4—*The internal RAM memory map shows that all of the 87LPC764 memory is used.*

too much current because the sampling would have to be done even if there was no intention of using the device.

The best solution is to embed the scan switch into the tip of the device so that it gets pressed automatically during use. Note that a small lever-activated micro-switch is well suited for this purpose.

Optosensor testing revealed that for the best rail-to-rail output signal, a simple op-amp comparator is needed. The processor's fixed internal voltage reference of 1.28 V is not well centered for this application (2.5 V is desired), and using an external reference would use another I/O pin, so an external op-amp is used instead of one of the processor's comparators. An external op-amp also allows buffering closer to the sensor and away from digital devices for noise protection.

The external op-amp chosen is MC33171, designed for single-supply operation, low 180-µA supply drain, and 1.8-MHz bandwidth. In addition, pin INT0/P1.3 is used for the detect signal, so scanning can be interrupt-driven. This pin is a Schmitt trigger input, which improves noise rejection.

The detect sensor signal must be such that moving the sensor from a light to dark marking causes the detect line to go from high to low, so Interrupt0 can be used (set for edge triggering). When the device is lifted from the paper, there is no reflection and the detect line is low, similar to that of a dark marking. This is not a problem, because the scan switch is depressed first, followed by a short wait before the sensor is read. At this point, detect should be high because of the paper's light area reflections.

SOFTWARE

Results for the current quiz scan are maintained in four bytes, where each bit indicates right or wrong for a particular question (see Figure 4). The cumulative scoring for up to 32 questions as well as the master results require 32 bytes.

An external hardware reset signal is not needed because the QuizWiz is using the watchdog. Whenever a reset occurs, the WatchdogOverflow bit is checked to determine if the watchdog caused it. Because the watchdog timer can't be turned off by the software when it enters shutdown mode, the device must expend a small amount of power to refresh it. Alternatively, the WD timer can be disabled completely at startup and no monitoring is needed during shutdown.

The main ISR loop uses Timer0 in the 16-bit mode 1 and always runs (see Listing 1). The stack will be used for three levels of subroutines. Interrupts will be disabled as needed to ensure this. Assuming that a single subroutine will need to push PC, DPTR, ACC, B, R0, and R1 (worst case), this will require 8 bytes. Therefore the 24 bytes from 08h to 1Fh are reserved for the stack. Store 07Eh at location 20h at the bottom of the bit-addressable memory to monitor stack overflow. This value will be checked during each Timer0 interrupt, and if changed, it will cause a software reset.

LAST THOUGHTS

The most challenging aspect may have been cramming all of the circuitry into the smallest possible package. Several iterations were required to optimize mechanical constraints with battery size, circuit layout, and assembly methods.

The software was limited mainly by the amount of available RAM, but the 8051 architecture proved adequate because of bit-flag memory. If even more RAM was available, then additional scoring details could be stored or perhaps more complex grading results. And, if more I/O pins were available, then you might consider using a full 8-bit LCD interface and a larger character display.

Another improvement would be if the LCD could run directly off battery voltage, which of course varies over time. This might lower current drain and eliminate the need for a 5-V regulator. However, a brief search for such a display was not successful.

The scan timing currently is determined by the use of simple sync marks at the top of each column and, therefore, depends on a uniform scan speed. In a more elaborate approach, the device would determine the scan rate from the timing of the markings themselves and adapt to any changes as the scan progresses.

But, the difficulty is that the processor must store a significant amount of scan data and post-process it to evaluate the variable scan rate and take into account marked spaces. ■

Paul Kiedrowski obtained a BSEE from Rose-Hulman Institute in 1982 and has worked in the RF, microwave, and communications industries since then. Currently, he is senior staff engineer at Motorola in Fort Worth, Texas, developing base station infrastructure equipment. You can reach him at kiedro@swbell.net.

SOURCE

87LPC764
Philips Electronics N.V.
+31 40 27 34184
Fax: +31 40 27 32250
www.philips.com