A previous Design Idea described a single-wire-plus-ground-return, keypad-to-microcontroller interface in which a single pulse represents each keystroke (Reference 1). The pulse's width is proportional to the key's numerical value, and the microcontroller identifies the pressed key by measuring the pulse width. Component tolerances and the accuracy of the microcontroller's internal oscillator limit the original design to keypads with 16 or fewer keys—that is, four rows by four columns or smaller crosspoint-key matrix. This Design Idea illustrates how a relatively simple modification applies the method to much larger keypads. (The following description omits a few details from the original Design Idea, which you can find here.)

You can divide a large keypad or keyboard into sections of 12 keys each (Figure 1). Each section connects to a separate comparator circuit, which detects a keystroke and generates a trigger pulse using the monostable circuit of IC₄, and you can add more sections in the same manner. Diodes D₁, D₂, and D₃ couple and isolate the comparators' outputs to Pin 2 of IC₄. Each keypad section connects the same selection of timing-resistor values to the monostable. Therefore, each keypad section's output pulse widths occupy the same range: 110 to 1320 µsec.

For the microcontroller to identify an active keypad section, the circuit generates a single, double, or triple pulse, depending on whether the pressed key resides in section 1, 2, or 3, respectively. The microcontroller identifies the pressed key by measuring the width of the first pulse and identifies the keypad section by counting the number of pulses. Implementing the multiple-pulse-encoding scheme requires no additional active components. Pressing a key in Section 1 generates a single pulse. Pressing a key in Section 2 causes comparator IC₂'s output to go low and triggers IC₄ to generate the first pulse. After feedback through R₁₅ and C₈ produces a 2-msec delay, IC₃ triggers IC₄ to produce the required second pulse. Because the time-constant product of R₁₅×C₈ is much longer than 2 msec, C₈ does not charge significantly during the 2-msec interval. However, C₈ does charge to a full 5V during the interval in which the key is pressed, which allows comparator IC₁ to recover to its steady state.

In a three-section keypad, pressing a key in the third section activates comparator IC₃ which in turn triggers IC₄ to produce the first pulse. After feedback through R₁₇ and C₁₀ produces a 2-msec delay, IC₂ triggers IC₄ to produce a second pulse. After yet another 2-msec delay, IC₁ triggers IC₄ to produce the third and final pulse. Although somewhat arbitrary, a 2-msec delay provides sufficient margin over the maximum key-pressed pulse width of 1.32 msec. The interrupt routine provides additional timing margin by allowing as much as 3 msec between pulses.
Listing 1, represents a modified version of the interrupt routine in Reference 1 and supports the circuit in Figure 1 for any number of keypad sections. For a three-section-keyboard implementation, the routine returns a key number from 1 to 36.

Reference

Click here for more Design Ideas!