Microcontroller inputs parallel data using one pin

Rex Niven, Forty Trout Electronics, Eltham, Victoria, Australia

Inputting multiple bits of information using a single entry pin of a microcontroller without the complexity of UARTs can prove useful. Such a scheme could allow scanning of a keyboard, mode switches, or any relatively slowly changing digital data. **Reference 1** details a technique for outputting signals with a single pin. The data from switch bank S_1 first presents itself to IC_3, a 74HC165 parallel-to-serial converter from NXP Semiconductors (www.nxp.com, **Figure 1**). Loading the data into the shift register requires a pulse on the PL line (Pin 1). Line CK accomplishes this pulse by sending as output a long pulse on the microcontroller-pin line. R_2 and C_2 introduce a delay, and, once the pulse exceeds that delay, the PL line goes low, and the data loads.

After the PL signal rises, shorter pulses on the microcontroller's I/O port generate pulses at the shift register's clock input, CP, but not at the PL input. The duration of these clock pulses must be long enough to exceed delay R_1C_1 but not R_2C_2. These clock

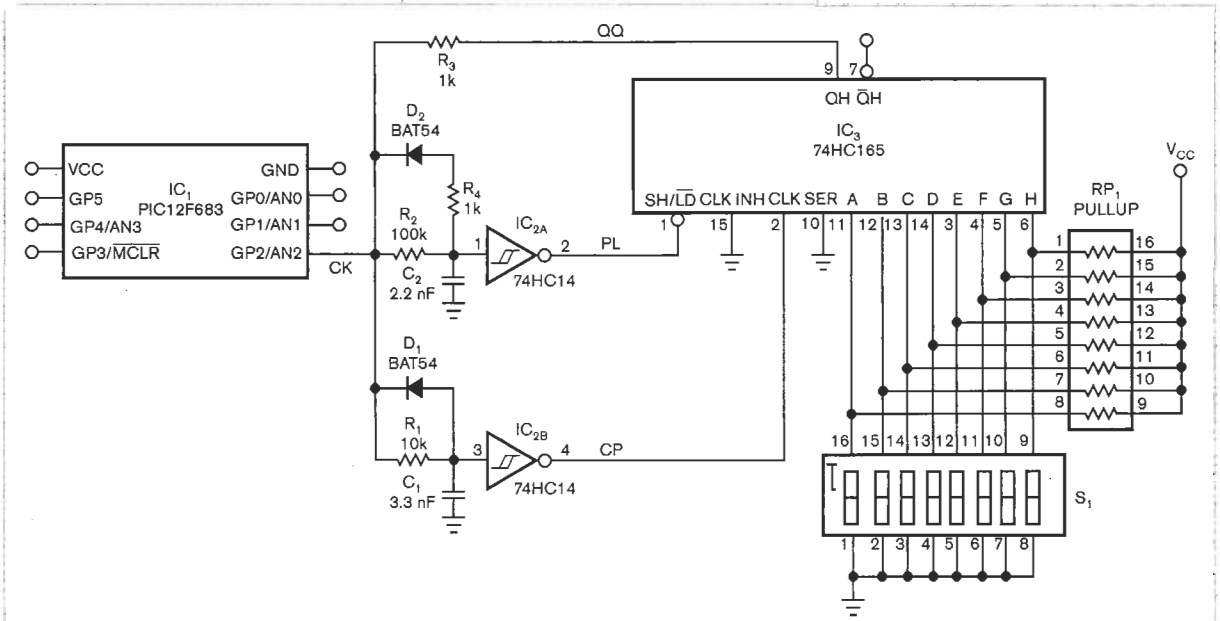

Figure 1 Careful adjustment of the RC time constants allows a microcontroller to input a serial-data stream using a single I/O pin.

pulses shift the data so that the 8 bits appear in sequence at the shift-register output, QQ.

If the microcontroller's data direction briefly changes to input with high impedance, this shift-register data dominates because of the relative values of R_1, R_2, and R_3, with R_3 being a much lower value. The high-impedance state must exist only for a time less than the R_1C_1 time constant (**Figure 2**). The microcontroller now reads the single bit of data. The action of three differing periods generates three functions: load, clock, and data read. The time the microcontrollers need to change port direction, read the pin data, and reset the pin's direction to output determines the timing. For example, a 1-µsec microcontroller requires 10 µsec.

To avoid spurious CP pulses, this time constant must be less than $0.33R_1C_1$, so R_1C_1 could be 30 µsec and R_2C_2 could be 200 µsec. These settings would allow a complete 8-bit read in about 1 msec. To achieve faster operation, replace the RC delays with a precision retriggerable monostable multivibrator, such as NXP's 74HC123, and logic gates. You can expand the scheme with more shift registers to read dozens of signals.

Note that internal logic in the 74HC165 shift register prevents the CP signal from shifting data when LD is active. Resistor R_4 ensures the correct sequencing of LD and CP. Diodes D_1 and D_2 quickly discharge the capacitors to "reset" the delay function of R_1C_1 and R_2C_2.EDN

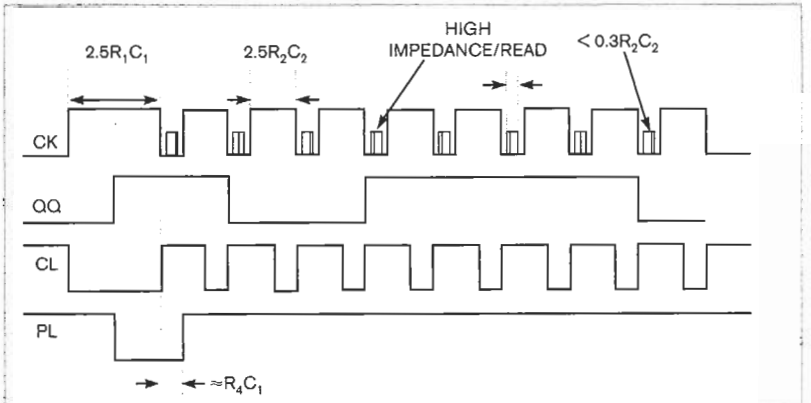

Figure 2 The high-impedance state must exist only for a time less than the R_1C_1 time constant.

REFERENCE

Niven, Rex, "RC lowpass filter expands microcomputer's output port," *EDN*, June 21, 2007, pg 74, www.edn.com/article/CA6451248.