# Voltage-to-pulse-width converter spares microprocessor's resources

James Christensen, Kris Design Co, El Cajon, CA

Although not an ADC in the classic "stream-of-ones-and zeros" sense, this voltage-to-pulse-width converter produces a logic-level output pulse whose variable width represents an analog of the input voltage. Based on Atmel's (www.atmel.com) AT89LP4052 microprocessor, $IC_1$, this circuit makes efficient use of the target microprocessor's limited analog-port pinout and code space by using a modified version of the classic timed-discharge-RC (resistor-capacitor) ADC design.

The timed-RC ADC allows a capacitor to charge through a resistor while the microprocessor increments a counter. When a comparator detects that the capacitor voltage and analog- input voltage are equal, the count terminates, and its stored value represents the ADC's output. However, an RC network's exponential charging characteristic produces a nonlinear conversion. Various software and hardware techniques can partially correct the nonlinearities, but all entail adding code, increasing the circuit's development time, or consuming additional I/O-port pins required for other purposes.

To produce a linear-charging characteristic that needs no correction, the circuit in **Figure 1** uses an LM334 constant-current source, $IC_2$, to drive capacitor $C_2$, which connects to $IC_1$'s $AIN_0$ analog-input port. An internal timer in the microcontroller measures the elapsed time from the charging ramp's start to the instant when the ramp voltage crosses the analog-input-voltage threshold at $IC_1$'s $AIN_1$ port.

In this application, potentiometer $RV_1$ provides an analog-input voltage proportional to its position. The width of the positive-going pulse at the output, P1.5, varies in proportion to the analog-voltage input. Note that I/O-port pin $AIN_1$ serves a dual purpose as an analog input and as an open-drain output that discharges ramp-forming capacitor $C_2$ before the next conversion cycle.

An 8-bit voltage-to-pulse-width-conversion cycle completes in less than 4 msec. The code performs the conversion function and outputs a pulse train at $IC_1$'s port P1.5 (Pin 17) with a period of 100 msec and a positive-going pulse width proportional to the analog-input voltage at Pin 13 ($AIN_1$). Programming connector $J_1$ provides access to $IC_1$ for

uploading the compiled code. The AT89LP4052 microprocessor typically executes one instruction per clock cycle, and a 10-µsec timer routine can perform the required housekeeping functions with plenty of time left over for other program tasks, including a future application that requires a binary-coded analog-to-digital output. You can download **Listing 1**, which is written in C for the Keil Software (www.keil.com) compiler, from the online version of this Design Idea at www.edn.com/061201di1.EDN

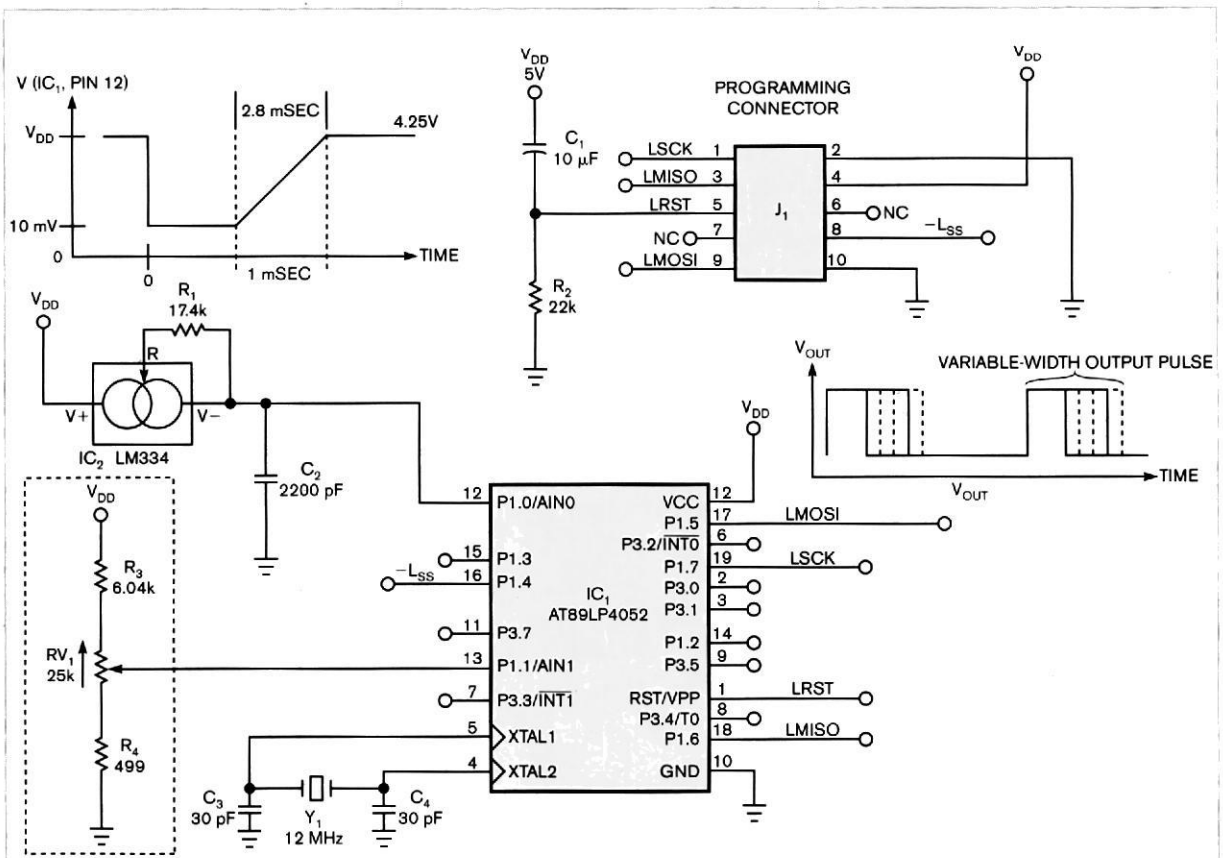


Figure 1 An analog-voltage-to-pulse-width converter features minimal parts. Subgraphs show timing-network and output-voltage waveforms. $IC_1$'s unlabeled pins are available for user functions.