Analog-to-Digital Conversion Utilizing the AT89CX051 Microcontrollers

8-Bit Microcontroller with Flash

Application Note

The Atmel AT89C1051 and AT89C2051 microcontrollers feature on-chip Flash, low pin count, wide operating voltage range and an integral analog comparator. This application note describes two low-cost analog-to-digital conversion techniques which utilize the analog comparator in the AT89C1051 and AT89C2051 microcontrollers.

RC Analog-to-Digital Converter

This conversion method offers an extremely low component count at the expense of accuracy and conversion time. In the example presented below, resolution is better than 50 millivolts, accuracy is somewhat less than a tenth of a Volt and conversion time is seven milliseconds or less.

As shown in Figure 1, the RC analog-to-digital conversion method requires only two resistors and a capacitor in addition to the AT89CX051 microcontroller. A microcontroller output (pin 11), which swings from approximately ground to V_{CC}, alternately charges and discharges the capacitor connected to the non-inverting input of the internal comparator (pin 12). The microcontroller measures the time required for the voltage on the capacitor to match the unknown voltage applied to the inverting input of the internal comparator (pin 13). The unknown voltage is a function of the measured time.

The HP5082-7300 LED displays shown in Figure 1 are not required for the conversion, but are utilized by the software to implement a simple two-digit voltmeter. The result of the analog-to-digital conversion is displayed in volts and tenths of a volt on the two displays. The voltmeter application does not utilize the full resolution of the RC conversion software, but serves to demonstrate the method as well as providing a tool for debug.

The waveform for a typical capacitor charge/discharge cycle is shown in Figure 2. The discharge portion of the curve is identical to the charge portion rotated about the line $V_C = V_{CC}/2$. The equations and discussion below apply to the charge portion of the cycle, except where indicated.

The voltage on the capacitor as a function of time is given by the exponential equation:

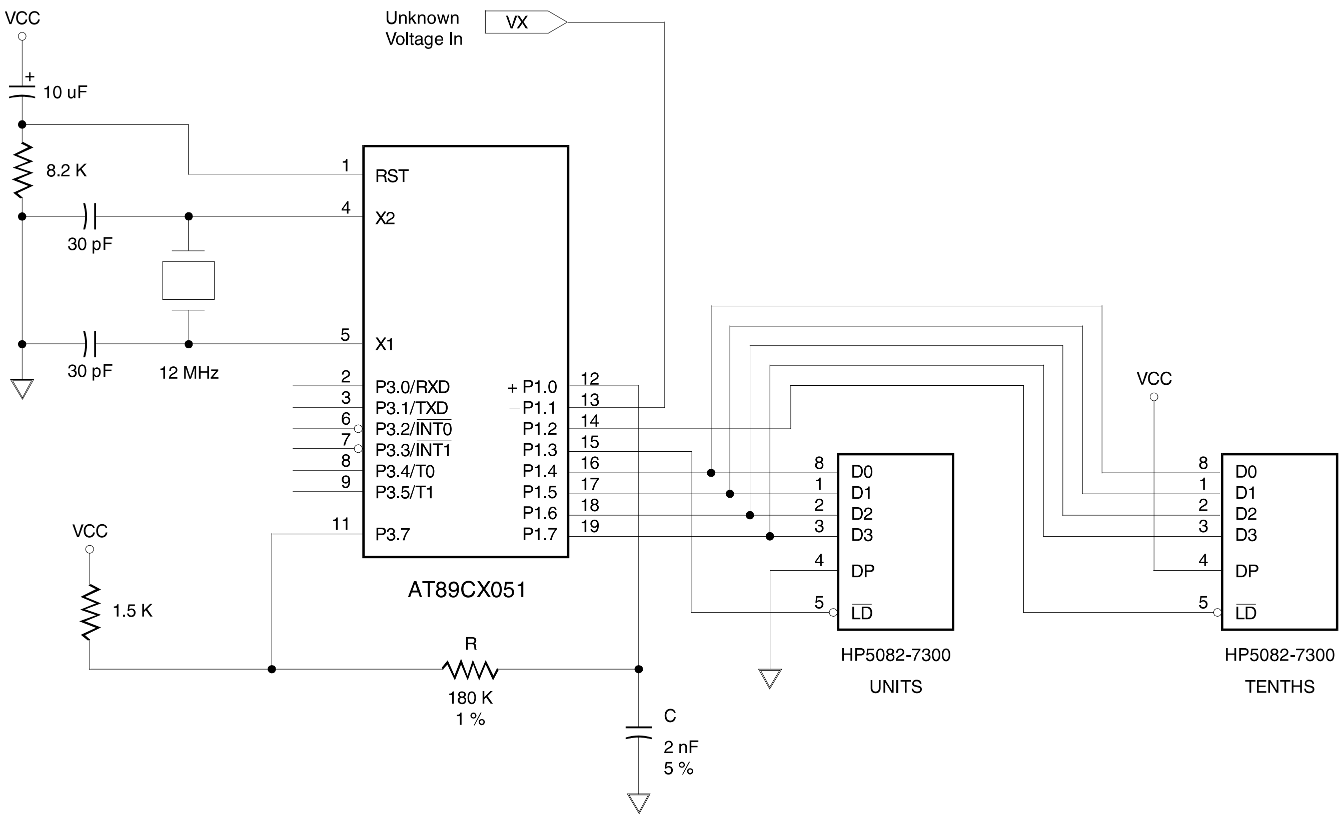
$$V_C = V_{CC} (1-e^{-t/RC}) \qquad (1)$$

where V_C is the voltage on the capacitor at time t, V_{CC} is the supply voltage and RC is the product of the values of the resistor and capacitor. Note that voltage is expressed in Volts, time in seconds, resistance in Ohms and capacitance in Farads. The product RC is also known as the “time constant” of the network and affects the shape of the waveform. The waveform is steepest when capacitor charging or discharging begins and flattens with time.

The first problem with the RC conversion method is the difficulty of solving the exponential equation without utilizing floating point calculations and transcendental functions. On a compressed time scale, the exponential curve appears straight over much of its length, suggesting that it might be approximated by a line. This scheme fails due to the continuous variation in slope over the length of the curve, which produces significant error. It also does not ad-

0524A

Figure 1. Two-Digit Voltmeter



dress the problem where the curve rolls off severely near the asymptote at V_{CC}.

The microcontroller need not solve the exponential equation in real time if a lookup table is used to map pre-calculated values to each sampled time interval. This scheme allows the data to be encoded and formatted as required by the application while simplifying the conversion software. Symmetries in the data may be exploited to reduce the size of the table.

The second problem with the RC conversion method is the substantial error which results from variations in component values. Figure 3 shows an exaggerated view of the variation in the voltage on the capacitor due to variations in the values of the resistor and capacitor. As shown in the figure, the variation in the voltage on the capacitor decreases as the voltage on the capacitor decreases.

The symmetry of the capacitor charge/discharge cycle can be exploited to reduce the effect of variations in component values on conversion accuracy. This is done by utilizing the charge portion of the cycle to measure voltages less than $V_{CC}/2$ and the discharge portion to measure voltages greater than $V_{CC}/2$. The worst case error is reduced to the error at $V_{CC}/2$.

Before component values can be assigned, the time interval at which the comparator output is to be sampled must be determined. The sample interval should be as short as possible to maximize converter resolution and minimize conversion time. The sample interval is limited by the time required to execute the requisite code, which is determined by the clock rate of the microcontroller. In the voltmeter application, the microcontroller operates with a 12 MHz clock, resulting in a sample interval of five microseconds.

The time constant (RC) affects the shape of the capacitor charge/discharge waveform. The value of the time constant must be chosen so that the steepest parts of the waveform are resolvable to the desired resolution. The steepest part of the charge portion of the waveform occurs near the origin, while the steepest part of the discharge portion occurs near V_{CC}. Due to the symmetry of the waveform, the same time constant may be used for measurements made on either portion of the waveform.

Figure 4 shows an expanded view of the relationship between voltage and sample time near the origin. In the figure, ΔV is the desired voltage resolution of the converter and Δt is the sample interval determined previously. The curve labeled 'V_C' represents the voltage on the capacitor, which appears linear at this scale. In the figure, the slope of the curve is ideal, causing sampling to occur near the center of the voltage intervals. The slope of the curve may be less than shown, but may not be greater, or resolution will be lost. Note that the first sample is offset from the origin by 1/2 Δt to center the sample in the first voltage interval.

To obtain the minimum value of the time constant which will produce the required slope at the first sample, solve Equation 1 for RC:

$RC = -t/1n(1-V_C/V_{CC})$ (2)

Then set ΔV to the minimum desired resolution (0.05-volt), Δt to the sample interval determined previously (five microseconds), and calculate RC at the first sample point, where $V_C = 1/2\ \Delta V$ and $t = 1/2\ \Delta t$:

$$RminCmin = \frac{(-1/2)\Delta t}{\ln[1-(1/2(\Delta V))/VCC]} = \frac{-(1/2)(5 \cdot 10^{-6})}{\ln[1-(1/2)(0.05)/VCC]}$$

$$\cong 4.99 \cdot 10^{-4}$$

The product of the values of R and C must not be less than the calculated minimum time constant. Utilizing a resistor with a one percent tolerance and a capacitor with a five percent tolerance:

$(Rnom-1\%)(Cnom-5\%) >= 4.99 \cdot 10^{-4}$.

In the voltmeter application, the selected values of R and C are 267 kilohms and 2 nanofarads, respectively, yielding a minimum time constant of approximately $5.02 \cdot 10^{-4}$.

An additional constraint is placed on the value of R. Referring again to Figure 1, note the 5.1 kilohm pullup resistor connected to pin 11 of the microcontroller. This resistor is present to supplement the microcontroller's weak internal pullup, but has the detrimental effect of changing the time constant of the RC network during the charge portion of the capacitor charge/discharge cycle. This produces an asymmetry in the charge/discharge waveform, which contributes to conversion error. To minimize the effect of differences in the capacitor charge and discharge paths, the value of R should be chosen to be much greater than the value of the pullup resistor. In the voltmeter application, the selected value of R is 267 kilohms, which exceeds the value of the pullup resistor by more than an order of magnitude.

The time constant (RC), which is a function of the desired converter resolution, determines the duration of the capacitor charge/discharge cycle. The more time required for the capacitor to charge and discharge, the greater the number of samples required in the measurement loop and the greater the number of entries in the lookup table.

The time required for the capacitor to charge and discharge is approximated by calculating the maximum time for the voltage on the capacitor to rise to within one half of the smallest resolvable voltage interval from the asymptote. For the charge portion of the waveform, the asymptote is at V_{CC}. Due to the symmetry of the waveform, the determined value applies to both the charge and discharge portions of the cycle.

Solving Equation 1 for time yields:

$t = -RC \cdot \ln(1-V_C/V_{CC})$. (3)

Assuming a resolution of 0.05 Volt, the desired capacitor voltage is:

$V_C = V_{CC}-(1/2)(0.05) = V_{CC}-0.025$.

From Equation 3:

$tmax = -RmaxCmax \cdot \ln(1-(V_{CC}-0.025)/V_{CC})$

$= -(Rnom+1\%)(Cnom+5\%)\ln(0.025/V_{CC})$

$= -(1.01)(267 \cdot 10^3)(1.05)(2 \cdot 10^{-9})\ln(0.025/5.0) \cong 3$ ms

The minimum number of samples required in the measurement loop is determined by calculating the time required for the voltage on the capacitor to reach $V_{CC}/2$ and

Figure 2. Typical Capacitor Charge/Discharge Cycle

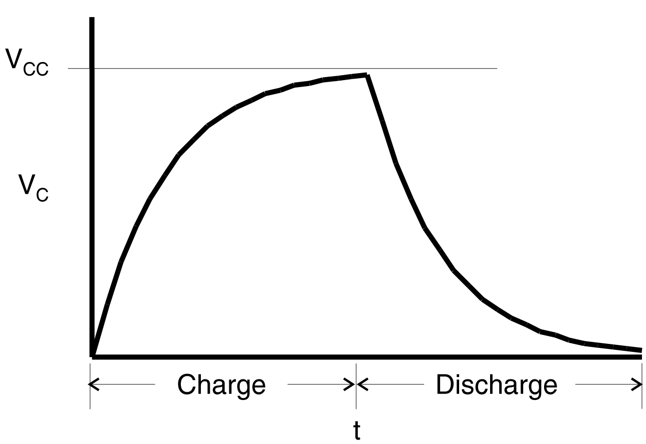
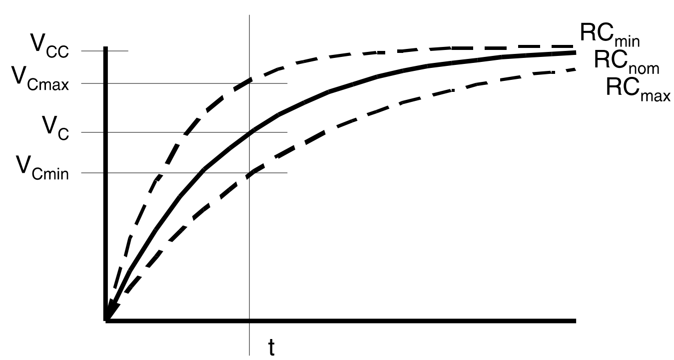


Figure 3. Capacitor Voltage Variation as a Function of RC Variation



dividing the result by the sample interval. The maximum value of the time constant is used in the calculation, since the voltage on the capacitor rises slower when the values of the resistor and capacitor are large. Due to the symmetry of the capacitor charge/discharge waveform, the determined sample count may be used for measurements made during either portion of the cycle.

From Equation 3:

$t_{max} = -R_{max}C_{max} \bullet \ln(1-(1/2)V_{CC}/V_{CC})$

$= -(R_{nom}+1\%)(C_{nom}+5\%)\ln(1/2)$

$= -(1.01)(267 \bullet 10^3)(1.05)(2 \bullet 10^{-9})\ln(1/2)$

$\cong 393\ \mu s.$

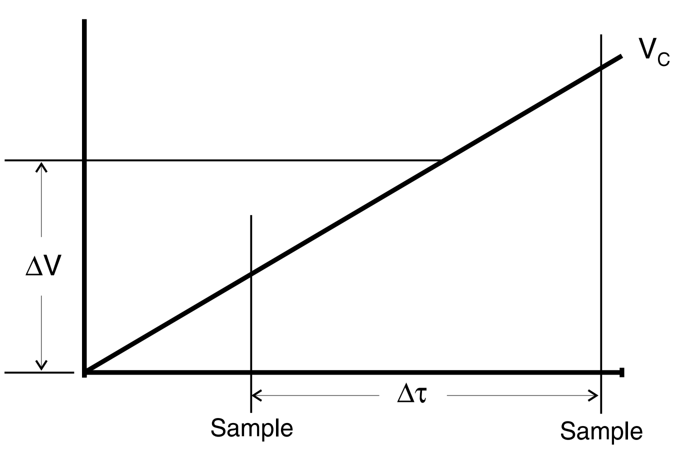
The minimum number of samples for half the cycle is:

$t_{max}/\Delta t = (393 \bullet 10^{-6})/(5 \bullet 10^{-6}) = 79.$

To maximize accuracy, voltages from zero to $V_{CC}/2$ are measured during the charge portion of the capacitor charge/discharge cycle and voltages from V_{CC} to $V_{CC}/2$ are measured during the discharge portion of the cycle. As a result, the total number of entries in the table is twice the number of samples calculated previously for each half cycle.

The lookup table contains application-specific values corresponding to the calculated voltage at each sample. For each half cycle, the Nth entry in the table corresponds to the voltage at t = (N-1) Δt, where Δt is the sample interval determined previously. For the charge half cycle, the voltage at each sample is calculated by solving Equation 1 for the time elapsed since the capacitor began to charge. For the discharge half cycle, the voltage at each sample is calculated by solving the following equation for the time elapsed since the capacitor began to discharge:

Figure 4. The Relationship between Voltage and Sample Time near the Origin



$$V_C = V_{CC} \bullet e^{-t/RC} \qquad (4)$$

The size and contents of the table may vary from application to application depending on the sample interval and conversion resolution. As the resolution increases, the number of entries in the table grows.

In the voltmeter application, with resolution equal to 0.05 Volt, the lookup table contains 158 entries, which is twice the number of samples per half cycle calculated above.

Voltages corresponding to samples taken during the charge half cycle are calculated by replacing 't' with 'N Δt' in Equation 1, where N represents the sample number (0-78). By setting Δt equal to the sample interval of 5 microseconds, R to 267 kilohms, C to 2 nanofarads, and Vcc to 5.00-volts, Equation 1 becomes:

$V = 5(1-e^{-N\,(.0093633)}).$

Voltages corresponding to samples taken during the discharge half cycle are calculated by replacing 't' with 'N Δt' in Equation 4, where N represents the sample number (0-78). Using the same values as for the charge half cycle, Equation 4 becomes:

$V = 5 \bullet e^{-N(.0093633)}).$

An abbreviated list of the voltages calculated for the capacitor charge/discharge cycle is shown below. The ordering of the voltages, increasing in the first half, decreasing in the second, tracks the voltage on the capacitor and defines the ordering of the table entries.

N = 0	V= 0.000
N = 1	V= 0.047
.	.
.	.
.	.
N = 74	V= 2.499
N = 75	V= 2.523
N = 76	V= 2.546
N = 77	V= 2.569
N = 78	V= 2.591

N = 0	V= 5.000
N = 1	V= 4.953
.	.
.	.
.	.
N = 74	V= 2.501
N = 75	V= 2.477
N = 76	V= 2.454
N = 77	V= 2.431
N = 78	V= 2.409

As shown by the list, the number of samples in each half cycle is greater than required to reach the midrange value of 2.500 Volts. This allows for "fast" cycles which overshoot the nominal midrange value before the last sample is taken in each half cycle. Note that the difference between the calculated voltages at samples N=0 and N=1 is within the desired resolution of 0.050-volt, but the difference in voltage between adjacent samples decreases as N increases. This reflects the non-linear relationship between voltage and time in the circuit.

The calculated voltages shown in the list are not entered into the lookup table, but are used to determine the values of the table entries. In the voltmeter application, the calculated voltages are rounded to tenths of a volt and the result stored in the table in packed-BCD form, two digits per byte. Example: the table entry corresponding to 2.523-volts is 25 hex, which displays as
2.5-volts.

The voltmeter prototype demonstrated accuracy of +/- one count (0.1 Volt), but accuracy of somewhat less than a tenth of a Volt is about the best that can be expected from the RC analog-to-digital conversion method. Even using precision components, variations in component values may contribute an error of +/- 0.104-volt, as shown below.

To calculate the worst case error at V_C = 2.5-volts, first determine the corresponding t at the nominal values of R and C using Equation 3:

$t = -R_{nom}C_{nom} \bullet \ln(1-V_C/V_{CC})$
$= -R_{nom}C_{nom} \bullet \ln(1-2.5/5.0)$
$= -R_{nom}C_{nom} \bullet \ln(0.5)$.

Substitute for t in Equation 1 to get minimum Vc:

$V_C\text{min} = V_{CC}\,(1-e^{-t/(R_{max}\,C_{max})})$
$= V_{CC}\,(1-e^{(R_{nom}\,C_{nom}/R_{max}\,C_{max})\ln(0.5)})$
$= 5\,(1-e^{\ln(0.5)/(1.01 \bullet 1.05)})$
$\cong 2.399$ V

Again, for maximum Vc:

$V_C\text{max} = V_{CC}\,(1-e^{-t/(R_{min}\,C_{min})})$
$= V_{CC}\,(1-e^{(R_{nom}\,C_{nom}/R_{min}\,C_{min})\ln(0.5)})$
$= 5\,(1-e^{\ln(0.5)/(0.99 \bullet 0.95)})$
$\cong 2.607$ V

The results show a variation of 0.208-volts at 2.5-volts, or a worst case error of +/- 0.104-volts. The worst case conversion error may be further reduced by utilizing components with tighter tolerances. Conversion accuracy and linearity are also affected by the characteristics of the capacitor. The capacitor used in the voltmeter prototype is a polystyrene film type, which not only provides good accuracy, but minimizes error due to dielectric absorption and other effects.

Error sources which have not been examined include: comparator limitations; asymmetries between the charge and discharge portions of the cycle; failure of the voltage on the capacitor to reach ground or V_{CC}; variations in V_{CC}. The contributions to conversion error made by these sources can be expected to increase error to somewhat more than the value due to component tolerances alone.

Successive Approximation Analog-to-Digital Converter

This conversion method offers good resolution and accuracy and a short conversion time at the expense of increased component count.

Successive approximation (SA) ADCs incorporate a digital-to-analog converter (DAC), a comparator and a successive approximation register (SAR). The SAR controls the conversion by performing a search for the binary code which, when fed to the DAC, will produce an output matching the voltage to be converted. The comparator compares the DAC output to the unknown voltage and returns the result to the SAR.

The SAR begins the search with the most significant DAC bit, which controls the widest output variation, and moves toward the least significant bit, causing the DAC output to “zero in” on the unknown value. The result of the trial is the binary code corresponding to the unknown value. In an eight-bit SA converter, only eight iterations are required to find the correct binary code, resulting in relatively fast conversions.

In this application (Figure 5), an AT89CX051 microcontroller with an integral analog comparator performs the SAR function in software, reducing the component count. The DAC selected for the application is an MC1408-8, eight-bit, current output type chosen for its low cost. Seven- and six-bit versions are available as the MC1408-7 and MC1408-6, respectively. The MC1408 series is guaranteed accurate to within +/- 1/2 LSB at 25 degrees C at a full scale output current of 1.992 milliamps. The relative accuracy of the MC1408-8 is better than 0.19%, assuring eight-bit monotonicity and linearity. The DAC has an output settling time of 300 nanoseconds.

The DAC contains binary-weighted, current-steering switches which scale an input current by the applied binary code. The input current is derived from an LM336-2.5 precision voltage reference and a series resistor. The scaled current output is converted to a voltage by an LF355B operational amplifier wired as a current-to-voltage (I/V) converter. The LF355B op amp was selected for the I/V converter because of its low input offset voltage and high output slew rate. The voltage output of the I/V converter is fed into the AT89CX051 comparator, where it is compared to the unknown voltage. When the programmed voltage exceeds the unknown voltage the output of the comparator goes high, which is detected by software. A second op amp, wired as a non-inverting, unity gain buffer may be inserted between the unknown voltage source and the input to the AT89CX051 comparator to provide isolation.

The LM336-2.5 reference provides a nominal 2.490-volt output (Vref). The actual voltage may vary from 2.390-volts to 2.590-volts. The reference voltage and temperature coefficient may be trimmed using the method indicated in the LM336-2.5 data sheet. The nominal value of the current reference resistor (Rref) connected to pin 14 of the DAC is 1240 Ohms, yielding a reference current (Iref) of 2.490 V / 1240 Ohms (Vref/Rref) = 2.008 milliamps. The eight-bit binary code applied to the DAC scales Iref by from 0/256 to 255/256, resulting in a current output (Io) of from zero (Iref•0/256) to 2.000 milliamps (Iref •255/256) full scale. Note that the sign of the DAC output current is opposite the sign of the reference (input) current. The output voltage is determined by multiplying the DAC output current (Io) by the value of the I/V converter gain resistor (Ro). Nominal full scale output voltage is 2.000 mA•2500 Ohms (Io F.S.•Ro) = 5.000-volts.

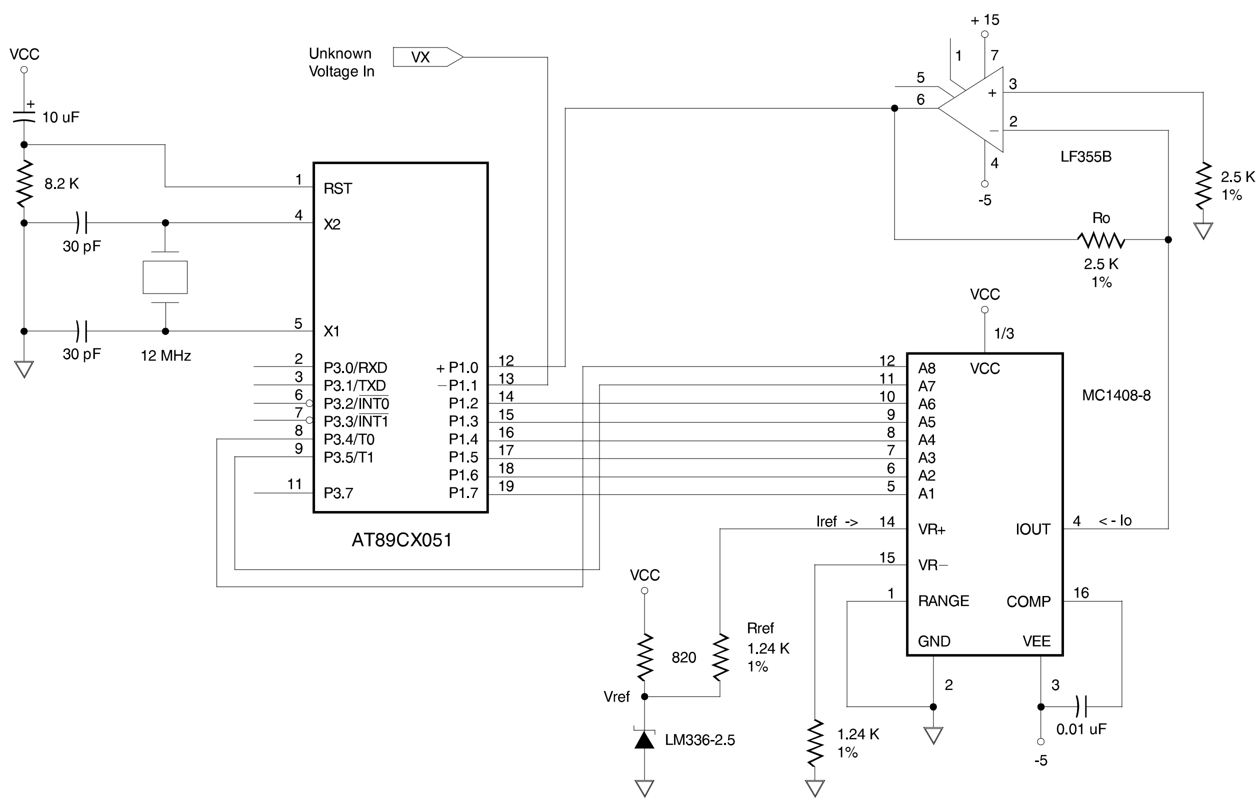
The circuit does not provide adjustments for offset or gain. Offset voltage adjustments should not be required, due to the low offset voltage specification of the LF355B op amp. If the offset voltage must be adjusted, add the offset trim circuit shown in the LF355B data sheet. The gain may be changed by changing the value of the I/V converter gain resistor (Ro).

The resistor connected to the non-inverting input of the op amp should be of the same value as the gain resistor for input bias current balancing. The 1240 Ohm resistor connected to pin 15 of the DAC and the 2500 Ohm resistor connected to pin three of the op amp may be eliminated with only a slight decrease in performance.

The MC1408-8 DAC requires power supplies of +5.0-volts and -5.0 to -15-volts; +/- 5.0-volt supplies were selected to minimize power consumption. The LF355B op amp requires bipolar supplies between +/-5.0-volts and +/-15-volts. -5.0-volts was selected for the negative rail for compatibility with the DAC, but may be replaced with -15-volts, if desired. The positive supply was chosen to be +15-volts to allow the limited output swing of the op amp to reach the five Volt upper input limit of the comparator.

The speed of the A-to-D conversion is limited by the DAC output settling time, the slew rate and settling time of the op amp, the response time and slew rate of the comparator and the time required to execute the successive approximation algorithm. The DAC output settling time and the comparator response time are negligible compared to

Figure 5. Successive Approximation Analog-to-Digital Converter



op amp delays and the time required to execute the SA algorithm, and so may be ignored. The maximum voltage step input to the op amp is five volts, which requires one microsecond to slew and four microseconds to settle (see the LF355B data sheet). This delay is accommodated in the software; consult the listing for additional information. With a 12 MHz processor clock and the resulting one microsecond instruction cycle, an eight-bit conversion can be performed in under 300 microseconds. The unknown input voltage must be held constant for the duration of the conversion.

Obvious disadvantages to the successive approximation analog-to-digital converter presented here are the need for bipolar power supplies and the large number of microcontroller I/O pins required to control the DAC. The +15-volt supply could be eliminated by replacing the LF355B op amp with a single supply, five Volt, functional equivalent with outputs that swing rail-to-rail. The number of microcontroller I/O pins required to control the DAC could be reduced somewhat by substituting a seven or six bit DAC. The parallel input DAC could be replaced with a (more expensive) serial input DAC. Alternately, logic could be added to accept serial data from the microcontroller and present parallel data to the DAC.

The software for this application may be obtained by downloading from Atmel's BBS: (408) 436-4309. Consult the comment block at the beginning of the source code file for detailed information on features and operation.