APPLICATION NOTE

ABSTRACT
Microcontrollers with on–chip A/D–converters may not be the first choice for extremely cost–sensitive applications. Philips' LPC family of microcontrollers covers the range from the lowest cost end up to parts with integrated ADC.

This application note shows how the lowest cost parts can take advantage of LPC features to measure analog signals even with high–resolution.

AN10187
Low-cost A/D-Conversion with Philips LPC Microcontrollers

Torsten Eggers, Thomas Schmidt

2002 Oct 04

INTRODUCTION

To process continuously varying data with digital computers, analog values have to be converted to digital quantities. Analog-to-digital converters (ADC) work according to different principles, varying in characteristics, effort and costs.

There are microcontrollers with integrated ADC offering 10-bit and higher resolution, but the required chip-area and thorough testing to guarantee the desired accuracy add to the cost of such devices.

Philips' LPC microcontroller families P87LPC76x and P89LPC900 cover a broad range of integrated peripherals including ADCs. This application note describes two methods to implement ADC functions even with the lowest-cost parts not having an integrated ADC.

SIGMA-DELTA PRINCIPLE

The Sigma-Delta (ΣΔ) principle is becoming more and more important for high-resolution ADCs and is proven in many applications.

Its major advantage is that predominantly digital signal processing is used, which also allows it to be integrated into digital ICs.

According to the Nyquist criterion, a signal to be converted must be sampled at a rate of at least twice its maximum frequency. A ΣΔ-converter samples at a much higher frequency to decrease the quantization noise. This oversampling reduces the requirements for the sample and hold circuitry and analog anti-aliasing filter, which in many cases can be just an RC-element.

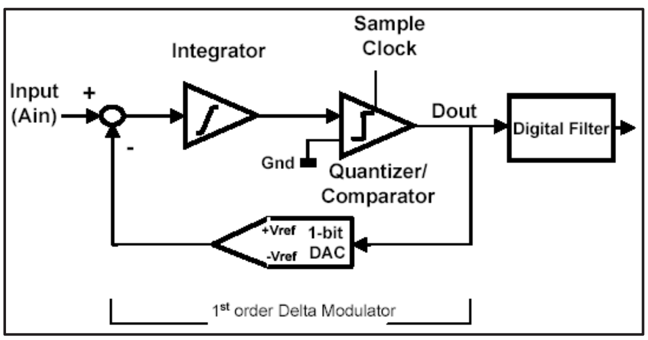

Figure 1.

Figure 1 shows a block diagram of a first-order delta modulator in its basic form. It consists of an integrator, a clocked comparator, and a single-bit digital-to-analog converter (DAC). The analog input signal Ain is summed with the output of the DAC of the feedback loop. The sum is then integrated and quantized by a comparator, which functions as a one-bit quantizer. This digital signal is converted back to analog using a one-bit DAC and fed back to the input's summing junction.

The density of digital "1s" at the modulator's output Dout is proportional to the analog input value. This bit stream is then digitally filtered and decimated to a result in a binary format by a decimation filter.

DUAL-SLOPE PRINCIPLE

Another approach for a low-cost ADC is to transform the voltage measurement into a time measurement. Microcontrollers are usually synchronized to a stable clock of an oscillator. This allows precise time measurements by software or on-chip timers/counters.

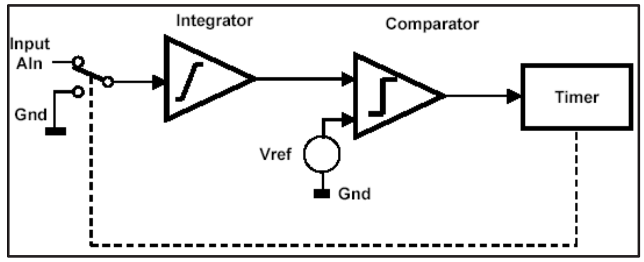

Figure 2.

The block diagram of Figure 2 shows a single-slope converter. The input of an integrator stage is switched from "0" to the analog input voltage (Ain). The output value of the integrator is compared to a known reference voltage (Vref). The time it takes for the integrator to reach the trip point of the comparator is proportional to the analog input voltage.

In real implementations of this single-slope principle the accuracy suffers from non-perfect components of the integrator (e.g. RC-tolerances, leakage currents) and comparator (offset voltage). A dual-slope converter compensates many of these effects by also counting the time it takes for the integrator to reach the "0" level again. The second slope starts when the trip point of the comparator is reached and the input switch is toggled from Ain to Gnd.

LOW-COST ADC WITH PHILIPS LPC MICROCONTROLLER FAMILY

The following features of the LPC microcontroller family particularly support the implementation of low-cost ADCs. All parts have:

- One or two on-chip analog comparators. Selectable input and output options allow the use of the comparators in different configurations.
- Two 16-bit counter / timers.
- Programmable port configuration options
  - Quasi bi-directional
  - Open drain
  - Push-pull
  - High impedance input.

Sigma-Delta ADC

Using the LPC comparators and the push-pull output capabilities, a simple Sigma Delta ADC can be built up with a minimum of external components. These blocks are used to balance switched current pulses. The pulses charge or discharge a capacitor to a voltage equal to the input voltage Vin. The LPC keeps track of the number of charge pulses during the measurement cycle. This method is relatively slow, but very accurate. Because the input voltage is averaged during the measurement cycle, the input voltage must be constant during the measurement to reach a high accuracy.

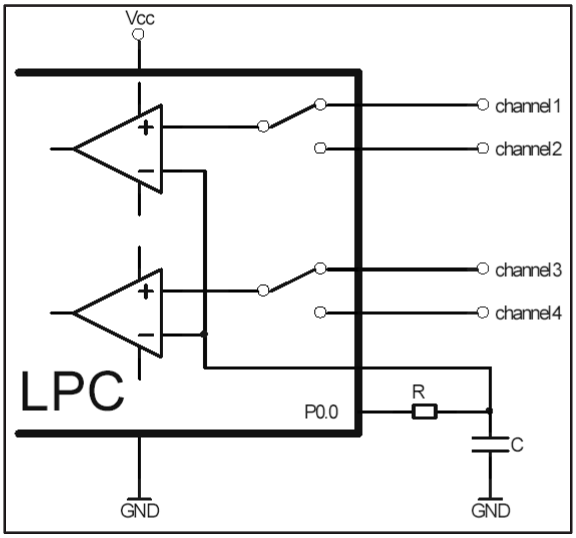

Figure 3.

The principle is suitable for many applications requiring a high-resolution measurement of slow-changing values. Some examples are battery charge control, temperature sensors, power meters and many more. The low power consumption of the LPC family makes this solution also suitable for battery-powered applications.

Figure 3 shows the schematic of a four-channel ADC with an LPC. The only external components needed are a capacitor (C) and a resistor (R), which determines the charging current. A stabilized supply voltage is mandatory to make the currents independent of power supply variations.

At conversion start C is charged via the port P0.0 to a value equal to the input voltage. C is connected to the common inverting input of the internal comparators and the voltages to be measured to their non-inverting inputs. The LPC family has up to two comparators with two selectable inputs each. Without any external analog multiplexing up to four channels can be measured sequentially.

After the precharge phase the measurement begins (see Figure 4). The software keeps the capacitor voltage (Vcap) equal to the input voltage by switching the port output to high or low depending on the comparator's condition.

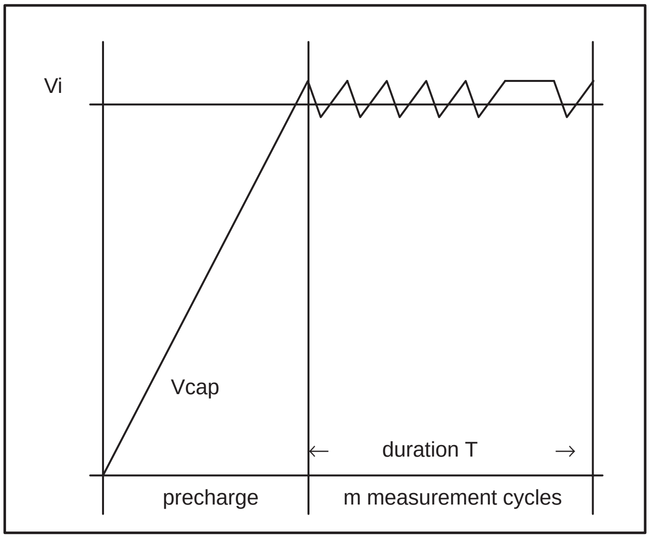

Figure 4.

Because the voltages at the beginning and the end of the measurement are equal, the charges on C at the beginning (Q_0) and at the end (Q_T) are equal, too.

When the output is high the charging current is:

$$I^+ = \frac{V_{CC} - V_i}{R}$$

When the output is low the discharging (negative) current is:

$$I^- = \frac{V_i}{R}$$

Because of the small voltage changes the exponential capacitor charging function for Q_T can be approximated:

$$Q_T = Q_0 + n \cdot I^+ \cdot T_{Cycle} – (M–n) \cdot I^- \cdot T_{Cycle}$$

where m is the total number of measurement cycles, n the number of high cycles, T_{cycle} the duration of a single cycle and T the duration of the complete measurement $(T = m \cdot T_{Cycle})$.

Because Q_0 equals Q_T:

$$(m - n) \cdot \frac{V_I}{R} = n \cdot \frac{V_{CC} - V_i}{R} \text{ and } V_i = \frac{n}{m} \cdot V_{CC}$$

The result is proportional to the number of high pulses and the supply voltage. For easy calculation it is advantageous to use as the number of measurement cycles a multiple of 10 of the supply voltage, e.g. 5000 for a 5 V supply. In this case the number of high pulses *n* is already the result in mV.

R and C are not critical and $\tau = R \cdot C$ should be chosen such that the voltage change in 1 cycle ($U_Ä$) is about 1 LSB. The charge function is again approximated:

$$U\Delta = V_{CC} \cdot \frac{T_{Cycle}}{\tau}$$

In the following example a Philips LPC76x microcontroller is used at an oscillator frequency of 18.43 MHz. One measurement cycle (m) consists of 25 machine cycles and takes T_{cycle}=8.14 µs. The target resolution is 12bit and V_{CC}=5 V.

R and C are selected as follows:

$$U\Delta = \frac{V_{CC}}{4096} = V_{CC} \cdot \frac{T_{Cycle}}{\tau}$$

For R and C follows:

$$\tau = R \cdot C \approx 4096 \cdot 8.14\ \mu s \approx 33.34\ ms$$

For a resistor of 100 kΩ the value for C is ≈ 33 nF. Values from 47 nF to 100 nF were tried in the lab with good success. The schematics of a 4-channel example are shown in Figure 5. There are only few requirements for the resistor and capacitor, e.g., temperature coefficient or tolerance can be neglected in most applications. Because any leakage current would influence the result, it is important to use a low leakage capacitor. Electrolytic capacitors are not recommended.

Special care must be taken to reduce the influence of noise and supply fluctuations on the accuracy. The integrating principle is of advantage here. For higher resolution or better noise suppression oversampling can be easily applied.

In the example the number of measurement cycles is ten times higher than calculated. For a 12-bit result 4096 cycles would be sufficient, but oversampling with 50000 cycles gives a much more reliable result.

Selecting a multiple of the mains supply period for the measurement duration suppresses mains hum. The optimal configuration depends on the application, the required resolution and accuracy and conversion time available.

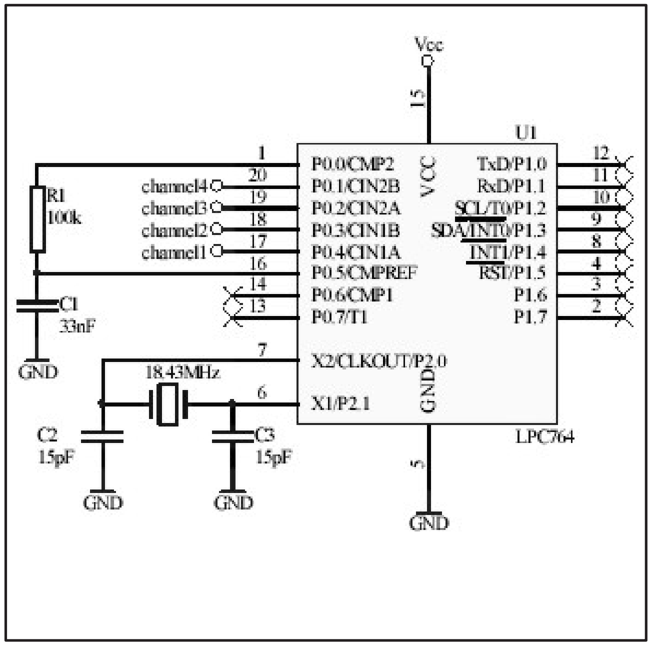

Figure 5.

The software is divided into two parts. The c-code part contains the initialization and calls the assembler conversion routine “Get_ADC”. The desired input channel (1–4 in a loop) is passed as a parameter. The result is stored in the two bytes *HighByte* and *LowByte* and subsequently sent to the serial interface via “printf”. The serial interface is configured to 19200 Baud based on the oscillator frequency of 18.43 MHz

The time-critical conversion routine is coded in assembler. The measurement cycle time, defining the charge and discharge times of the capacitor, has to be constant over the whole measurement.

Main.C

```
#include <relpc768.h> //or LPC900 header file
#include <stdio.h>
at 0xfd00 char code UCFG1=0x78; //SET UCFG1

data char LowByte,HighByte,CMPx,CMPModex;
data char CMPbufaddr,CMPbuf[2];

void        Get_ADC(char channel)
            {
            if (channel & 1)
             CMPModex = 0x030; //CinB
            Else
             CMPModex = 0x020; //CinA

            if (channel & 2)
             CMPbufaddr = &CMPbuf[1];
            Else
             CMPbufaddr = &CMPbuf[0];

            Get_SD_ADC();
            }
void main (void)
{
unsigned int Result, i;
PT0AD=0xFE;          //Disable digital inputs P0
P0= 0xFF;
P0M2= 0x001;
P0M1= 0x0FE;
P1= 0x0FF;
P1M2= 0x0DD;
P1M1= 0x022;

SCON  = 0x50; //serial port 19200 Baud
TMOD |= 0x20;
TH1   = 0xFB;
TR1   = 1;
TI    = 1;

printf (”LPC SD ADC\r\n”);

while (1)
  {
  for (i=0;i<4;i++)
            {
            Get_ADC(i);
            Result=LowByte + 256 * HighByte;
            Result=     Result/10;
            printf (”%u ;”, Result);
            }
            printf (”\r\n”);
  }
}
```

SD_ADC.A51

```
$NOMOD51
$INCLUDE(RELPC764.INC)
NAME SimpleADC
EXTRN DATA (LowByte,HighByte,CMPx)
EXTRN DATA (CMPModex,CMPbuf,CMPbufaddr)

SADC Segment Code
PUBLIC Get_SD_ADC
rseg SADC

Get_SD_ADC:
    PUSH ACC;
    PUSH PSW;
    PUSH AR4;
    PUSH AR5;

    MOV  R0,CMPbufaddr
    MOV  CMP1,CMPModex; load CMPMode
    MOV  CMP2,CMPModex; load CMPMode
    MOV  R4,#0;
    MOV  R5,#0;
Precharge:
    MOV  CMPbuf,CMP1
    MOV  CMPbuf+1,CMP2
    MOV  A,@R0 ;    get CMPx indirect
    ANL  A,#02h
    MOV  C,0E1h;    comparatorflag –> carry
    CLR  A
    MOV  080h,C ;   P0.0 = carry/comparator P0.0
;Counter
    CLR  C
    MOV  A,R4
    ADD  A,#01h
    MOV  R4,A
    MOV  A,R5
    ADDC A,#00h
    MOV  R5,A
    CJNE A,#040h,Precharge; Set Prechargecycles

    MOV  R4,#0B0h; Lowbyte count Measurecycles
    MOV  R5,#00h;
    MOV  LowByte,#00h;
    MOV  HighByte,#00h;
MAINLOOP:
;Sigma
    CLR  C
    MOV  CMPbuf,CMP1
    MOV  CMPbuf+1,CMP2
    MOV  A,@R0 ;    get CMPx indirect
    ANL  A,#02h
    MOV  C,0E1h;    comparatorflag –> carry
    CLR  A
    MOV  080h,C ;   P0.0 = carry/comparator
    ADDC A,LowByte; sum carry/comparator high
    MOV  LowByte,A
    MOV  A,HighByte
    ADDC A,#00h;    sum carry/comparator high
    MOV  HighByte,A
;Counter Measurementcycles
    CLR  C
    MOV  A,R4
    ADD  A,#01h
    MOV  R4,A
    MOV  A,R5
    ADDC A,#00h
    MOV  R5,A
    CJNE A,#0C4h,MAINLOOP; Highbyte count Mcycles
    POP AR5;
    POP AR4;
    POP PSW;
    POP ACC;

    RET
  END
```

This software application was tested with the LPC76x and the LPC900–family.

Dual-Slope ADC

AD converters based on the dual-slope principle can be implemented with the LPC families of microcontrollers with very few external components. The effort required depends on the desired resolution and accuracy. For lower resolution a simple RC-element suffices, while for higher resolution the use of additional active components like an OPAMP has to be considered.

Taking advantage of the internal comparators of the LPC76x and LPC900 microcontrollers, Figure 6 shows an example of a low-cost ADC with just an RC element. This low-cost concept is suitable to measure slowly changing analog values like temperatures etc.

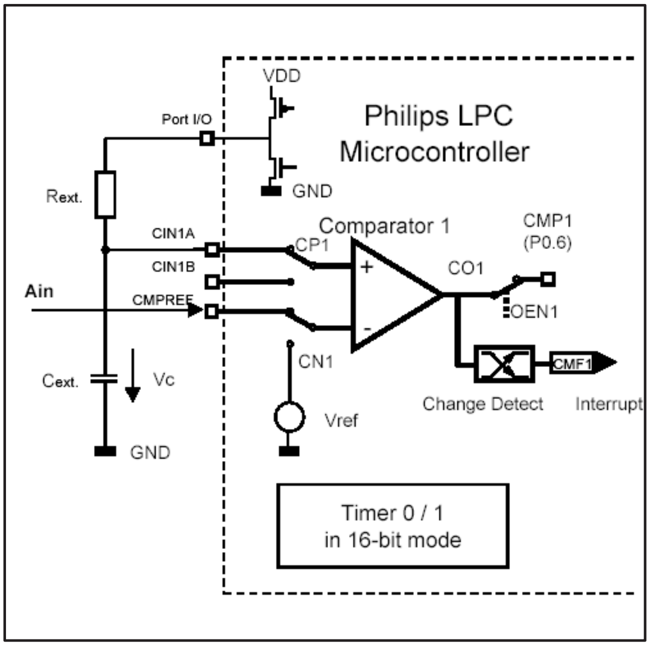

Figure 6.

The following steps are performed to determine the digital representation of the analog value of the input voltage Ain.

Calibration using the internal reference voltage

1. The inputs to the comparators are also programmable I/O-ports. To rapidly discharge the capacitor Cext the port is temporarily configured to output a logical “0”.
2. The negative input of the comparator (CN1) is then switched to the internal voltage reference of the LPC.
3. Now the I/O-port driving Rext. is switched to output a logical “1”. This starts to charge Cext via Rext. Simultaneously one of the timers T0 or T1 of the LPC is started in 16-bit mode. It counts the time it takes until the voltage on Cext reaches the level of the reference voltage. This event switches the comparator and can either be polled by software or – preferably – issue an interrupt request.
4. Cext is then discharged as described above and the timer is reset to its initial value.

Measuring of the analog input voltage

5. For the measurement of the input voltage the flow is similar to the calibration procedure. Instead of the reference voltage, the comparator input CN1 is now connected to the unknown analog input voltage Ain.
6. To start the measurement, the output-port is switched to “1” again, which starts the loading of Rext. via Rext. When the voltage on the capacitor reaches the level of the input voltage, the comparator stops the timer.

Calculation of the result

The resulting timer value corresponds to the analog input voltage.

$$V_{Ain} = V_{Cext} = V_{DD} \cdot (1 - e^{-t/RC})$$

Instead of calculating the e-function a look-up table is proposed. With this table the 16-bit timer values can be translated into voltages.

The table holds values according to the formula:

$$t = -RC \cdot \ln(1 - V_{Ain}/V_{DD})$$

Of course the table only consists of a limited number of values; an even distribution of voltage levels is recommended. For better resolution the measured timer values can be interpolated linearly between two table values.

The result of the calibration cycle is used to compensate tolerances and temperature drifts of the components (e.g. R, C, V_{dd}).

The advantage of this principle is that the CPU is not involved while the capacitor is charged or discharged. Only when the comparator’s trip voltage is reached the main task is interrupted. Some CPU time is required to calculate the result using the look-up table.

To further improve the accuracy of the result, not only the rising slope can be used. A second measurement can be done starting with a fully charged capacitor measuring the time it takes to discharge the capacitor to the reference or input voltage respectively. In this case the voltage on $C_{ext.}$ is:

$$V_{Cext} = V_{DD} \cdot e^{-t/RC}$$

A table with evenly spaced voltages and two separate timer values, one for the rising slope and another one for the falling slope, can be used to replace the calculations.

$$t_{rise} = -RC \cdot ln(1 - V_{Ain}/V_{DD})$$

$$t_{fall} = -RC \cdot ln(V_{Ain}/V_{DD})$$

The final result is the average of both slopes. This eliminates the comparator’s offset voltage. Additional measurement cycles would help to reduce noise.

Calculating or looking up the e-function can be avoided by adding an external current source (FET-circuit, OPAMP). In this case the voltage on $C_{ext.}$ would rise linearly over time.

To select the values of the external RC element the clock frequency of the microcontroller core has to be taken into account. The maximum duration of the charge or discharge cycle needs to fit into the 16-bit timer’s range. On the other hand, using as many timer values as possible increases the achievable resolution.

Assuming the timer is clocked once every 1 µs, the timer overflows after ~ 65.5 ms. Usable values for R and C are $R_{ext.}$=270 kΩ and $C_{ext.}$=47 nF. This gives ~12.7 ms for the time constant τ and a maximum charging time of about $5\tau = 64$ ms.

Special care must be taken for input voltages close to V_{SS} or V_{CC} . If the comparator's trip voltage is reached later than t=5τ, the timer overflow has to be taken into account.

REFERENCES

Please check the Philips Semiconductors web site for data sheets of the LPC Family of microcontrollers:
http://www.semiconductors.philips.com/

Low-cost A/D-Conversion with Philips LPC Microcontrollers

AN10187

Definitions

Short-form specification — The data in a short-form specification is extracted from a full data sheet with the same type number and title. For detailed information see the relevant data sheet or data handbook.

Limiting values definition — Limiting values given are in accordance with the Absolute Maximum Rating System (IEC 60134). Stress above one or more of the limiting values may cause permanent damage to the device. These are stress ratings only and operation of the device at these or at any other conditions above those given in the Characteristics sections of the specification is not implied. Exposure to limiting values for extended periods may affect device reliability.

Application information — Applications that are described herein for any of these products are for illustrative purposes only. Philips Semiconductors make no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Disclaimers

Life support — These products are not designed for use in life support appliances, devices or systems where malfunction of these products can reasonably be expected to result in personal injury. Philips Semiconductors customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips Semiconductors for any damages resulting from such application.

Right to make changes — Philips Semiconductors reserves the right to make changes, without notice, in the products, including circuits, standard cells, and/or software, described or contained herein in order to improve design and/or performance. Philips Semiconductors assumes no responsibility or liability for the use of any of these products, conveys no license or title under any patent, copyright, or mask work right to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask work right infringement, unless otherwise specified.

Contact information

For additional information please visit
http://www.semiconductors.philips.com. Fax: **+31 40 27 24825**

For sales offices addresses send e-mail to:
sales.addresses@www.semiconductors.philips.com.

Date of release: 10-02

Document order number: 9397 750 10415