The Analog to Digital Converter (ADC)

---

1 Overview

In many embedded system implementations, we need to measure analog voltages and make calculations based on their value. Control systems are a good example where sensor output voltages are measured to calculate a control signal. In this lecture we will be looking at the operation of the ADC peripheral in a somewhat superficial level, as the subject is quite deep. More information can be found in the accompanying presentation document on sensors.

2 Analog to Digital Conversion and Quantization

The ADC produces a unitless value x_q that corresponds to the equation:

$$x_q = \frac{V_{in}}{V_{REF}} \tag{1}$$

Where, x_q is the converted value, V_{IN} is the input voltage to be measured, and V_{REF} is the reference voltage.It can be seen that, the input voltage is compared to a reference voltage and the ratio becomes the ADC result $0 \leq x_q \leq 1$. However, remember that the computer cannot actually represent continuous values (that will result from this equation), but only discrete integer values of a given **precision**. Therefore, (1) is too simple and not correct. During the conversion, the continuous (or infinite precision) input voltage V_{IN} is converted to the finite precision value x_q. Some information about the input voltage is lost. This is called **"quantizaion"**. We will come back to how (1) should be changed after considering quantization in more detail.

The most prominent property of an ADC is the number of bits. An "n bit ADC" can represent the ratio of (1) in 2^n steps. Since the ADC can represent the voltage range $0 \leq V_{IN} \leq V_{REF}$ in 2^n steps, it will have a resolution $\Delta = \frac{V_{REF}}{2^n}$volts. This is the smallest change in the input voltage V_{IN} that will cause the output to change by 1.

For example, an 12 bit ADC that has $V_{REF} = 3.3V$ can represent the ratio in (1) in $2^{12} = 4096$ steps. Then the size of each step (resolution) is $\Delta = \frac{3.3V}{2^{12}} = 0.8mV$.

We can think therefore, that changes in V_{IN} that are less than Δ may get undetected; i.e., they may not cause a change in x_q. This is the **precision that we lose due to quantization**, i.e., the **quantization error**. The analysis of quantization error and its effect on the calculations (digital filter design with quantized signals and coefficients) is not difficult but somewhat involved, and therefore is out of the scope of this brief document. We can now propose a better representation of the conversion operation of the ADC in (2), considering the effect of quantization:

$$x_q = \left\lfloor 2^n \times \frac{V_{in}}{V_{REF}} + \frac{\Delta}{2} \right\rfloor \tag{2}$$

Note that (2) contains the "round down" operator, and not square brackets. It correctly represents that the ratio between V_{IN} and V_{REF} are scaled to 2^n. The addition of $\Delta/2$ causes the operation to be a "round to nearest" rather than "round down".

There are many other metrics for measuring the performance of ADCs. You can refer to:
N. Kehtarnavaz, M. Keramat, "DSP System Design Using the TMS320C6000", Prentice Hall, ISBN: 9780130910318, 2001
or
Shridhar Atmaram More, "ADC Performance Parameters", Texas Instruments Application Report, SLAA587, 2013.

The analog to digital conversion operation is a search operation. The processor needs to find a value that corresponds to (2) which is complex and time consuming.

Besides quantization as summarized above, another important part of the ADC conversion is the sampling operation. This is a subject of the digital signal processing theory, and will not be touched upon here.

3 ADC Peripheral of LPC824

There is one ADC peripheral in the LPC824, "ADC0", with a resolution of 12 bits. The microcontroller has 12 **channels**, which means that 12 pins are connected to ADC0 through an analog multiplexer. At any given time, we can select any one channel and measure its voltage, before moving on to another channel. The sampling rate can go up to 1.2M samples/s.

Remember that various functions on the chip could be connected to any physical pin on the package. For ADC this is not possible. ADC channels $0 \sim 11$ are each connected to a specific pin and cannot be relocated to another pin. For example, ADC0 channel 1 is always connected to package pin 23, which is also `PIO0_6`. However, analog function on those pins can be enabled or disabled and connected to other peripherals by using the `PINENABLE0` register of the SWM (See Sec. 7.5.13 "PINENABLE 0").

The important V_{REF} voltage is connected by default to the power supply of the microcontroller, which is $3.3V$ in Alakart. All calculations involving ADC should take that into consideration.

The ADC conversion can be triggered in several ways:

- Software: By setting bit 26 of "A/D Conversion Sequence A Control Register `SEQA_CTRL`" (See Sec. 21.6.2). See the example project "`adc_basic`".
- External pin: The logic value of an external pin starts a conversion.

- Timer: AD conversion is automatically triggered when timer SCT output 3 generates an event. See the example project "`adc_timer_trigger`".
- etc.

Especially by arranging a timer to trigger the ADC conversion, it is possible to get accurate timing for the sampling of analog signals.

The conversion may take a long time in terms of processor speed. Therefore, it should check if the conversion has been completed. That can be done in several ways:

- Polling: Main loop repetitively checks bit 31 `DATAVALID` of "A/D Global Data Register A and B" (see Sec. 21.6.4 "A/D Global Data Register A and B"). See the project `adc_basic` for an example of this.
- Interrupt: An interrupt is configured so that end of conversion directly calls a corresponding ISR. The ADC result can be serviced there. See the project `adc_interrupt` supplied in the course materials for an example of this.
- DMA: When the conversion is complete, the ADC peripheral informs the DMA peripheral, which transfers the result to a specified location in the memory. In this method, all of the sampling is completely done in hardware and the processor is not involved. See Sec. 21.7.6 "DMA control"

In many applications, several analog voltages need to be measured at each sampling time, such as the outputs of several sensors. In the LPC824, this has been made possible using two hardware sampling sequence lists. The ADC channels that need to be sampled for the application are registered into Sequence A or Sequence B registers. When the conversion of either sequence is triggered, all the registered voltages will be measured once, before the conversion complete event is generated. See Sec. 21.6.2 "A/D Conversion Sequence A Control Register" and "21.6.3 A/D Conversion Sequence B Control Register".

4 Caution Overvoltage!

The reader may think that any voltage can be applied to the pins of the processor. However **there are severe limitations** on what voltages can be connected:

- Max voltage that can be applied is V_{REF}! For Alakart $V_{REF} = 3.3V$. Any voltage exceeding that **will damage the chip permanently and may also cause damage to your PC.**
- Low power only! It is not possible to connect voltages directly from high power devices such as motor drivers or power supplies to the ADC inputs, even if their rated voltage is below $3.3V$. In such systems, voltage spikes may occur, which **may damage the chip permanently and may also cause damage to your PC.**
- Observe ground potential! Always connect the ground pin of the microcontroller to a corresponding ground pin. Observe that the grounds do not float with respect to each

other. Failure to do so - you guessed it - **will damage the chip permanently and may also cause damage to your PC.**

- No negative voltages! All voltages that the ADC peripheral can measure are positive with respect to ground potential. The ADC cannot measure negative voltages. And of course, failure to observe this **will damage the chip permanently and may also cause damage to your PC...**

Ahmet Onat 2023