

How do Analog to Digital Converters (ADCs) work?



0

Published May 17, 2022



0

The temperature of your room, the sound of sea waves, the intensity of sunlight throughout the day – every environmental observation in the world we live in is in the form of a continuous (in both time and amplitude) analog signal.

Even when we speak over the phone, our speech is analog in nature. Yet we also live in a world of 0s and 1s. This speech (analog signal) is converted into digital bits consisting of 0s and 1s making them suitable to be transferred over a telephone or other digital communication network and the same bits are reconstructed back at the receiver's end and played on a microphone whose output is perceivable audio – an analog signal.

The conversion of an analog signal to a digital one takes place with the help of ADCs or analog-to-digital converters. These are basically sampling and mapping circuits which sample the analog signal at uniform time intervals, and map the obtained sample to digital words made of binary numbers 0 and 1. This gives a digital representation of an analog occurrence.

discuss each of the blocks in detail.



Image 1: Simplified block diagram of ADC

Sample and Hold Block

Analog signals are continuous in both time and amplitude. The first step in converting an analog signal into a digital signal involves sampling of the analog signal amplitude at regular time intervals.

Consider the analog to digital (A to D) conversion of a sine wave. The image given below is of a sine wave, continuous in time and amplitude, taken as our signal.



Image 2: A sine wave – Analog signal

This analog signal is input to the sample block which consists of a sampler switch. The sampler switch is usually a simple switch which closes and opens at the sampling frequency as shown in the image below. When the switch closes, the amplitude of the signal at that point in time is available at the output of the switch. This instantaneous amplitude is fed to the hold block as long as the switch remains closed.

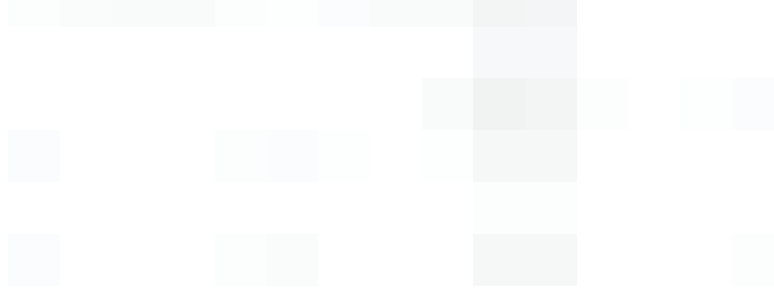


Image 3: Sample and Hold block circuitry

The holding of the sampled value is achieved with the help of a capacitor connected at the output of the sampler switch. As shown in the image above, when the switch closes the capacitor charges to the voltage value of the analog signal at that point in time. The switch can now be opened as the sampling process is complete, the capacitor will stay at the charged value of v_c until the next sampling instant – that is, when the switch closes again. The capacitor voltage is now the sampled version of the input signal. The voltage of the capacitor v_c i.e., the sampled signal vs time graph is shown below:



Image 4: Sampling the sine wave

The input sine wave was both continuous in time and amplitude. As the switch closes, the capacitor charges to the voltage at that point, and when the switch opens, it holds on to that voltage until the switch closes again. So, the capacitor voltage is in the form of rectangular strips with

output voltage (capacitor voltage) is continuous in time and *discrete* in amplitude. This voltage across the capacitor will be used for further processing.

An observation to be made from the animation here is that the samples don't exactly coincide with the signal amplitude. As shown in the image below, the sampled strips of voltage (black) and the sine wave signal (red) show an evident error (yellow). This error is unavoidable during sampling and can be minimised by increasing the sampling frequency. But increasing the sampling frequency too much will just lead to increased processing and more storage.



Image 5: Information loss due to sampling

Quantize Block

The input to the quantize block is the output voltage of the capacitor. The output voltage of the capacitor is in the form of rectangular strips across time whose amplitude remains constant throughout each sampling interval. Each strip (corresponding to each sampling interval) is now mapped to an equivalent binary number value comprising of 0s and 1s.

Let's understand the quantization process with the help of an example.

that the input voltage should range between 0V at the minimum to 1V at the maximum. The full voltage range of 0V to 1V will be mapped to a 3-bit binary number from 000 to 111 (each of the 3 places can either be a 0 or a 1: $2 \times 2 \times 2 = 8$ binary numbers).

The resolution of the ADC for these parameters is

$$\frac{1V}{2^3} = 0.125V$$

The resolution gives the minimum amount of voltage which is required to change the output of the ADC. Say the signal amplitude is about 0V, then the digital mapped number will be 000. Since our resolution is 0.125V, for all values of input signal amplitude between 0V to 0.125V, the output digital number will be 000. For a voltage above 0.125V to 0.250V, the mapped value will be 001. For signal amplitude between 0.250V to 0.375V, the output digital number will be 010. The transfer curve for the above conversion will be as shown in the image below:



Image 6: 3 bit ADC transfer curve (Quantization error 1LSB)

It is here that we talk about quantization error. For an amplitude of the analog signal of value anywhere between 0V to 0.125V, the output is the same – 000. This leads to loss of information as any value between 0V to 0.125V is treated

The quantization error can be reduced by decreasing the resolution voltage i.e., adding more bits in the digital output for the same input voltage.

The output from the quantization block is values of digital words, these are then cascaded and converted into a stream of bits for transfer through serial networks. The bits are obtained and then the signal can be reconstructed at the receiving side.

The resultant digital binary numbers can be output either in series or parallel. Parallel output requires a separate pin for each of the binary digits ultimately increasing the size of the ADC chips. Serial output requires just one pin and is more economical to manufacture.

The application of ADC is wide ranging. A basic and common example is in a microcontroller - we interface sensors which record analog data. ADCs come mounted on the microcontrollers converting the recorded analog data to digital bits for further processing. Air conditioners, cameras, oscilloscopes – almost all devices and gadgets available in the market employ some digital operation and use ADCs in them. Understanding how they work and their limitations can be incredibly useful in beneficial utilization of these modules in designs.

[ADC \(3\)](#)[Microcontroller \(83\)](#)

AUTHORED BY

Uma Bharathi D



and embedded systems.

Related EE FAQs

What's the difference between a regular relay and a power relay?



Answered by Jayesh Upadhyay

What is a PWM signal?



Answered by Jayesh Upadhyay

Why is RAM faster than other memories?



Answered by Jayesh Upadhyay

What is the Difference Between Analog and Digital Signal Processing?



Answered by Amrutha Varshini

Related Tutorials



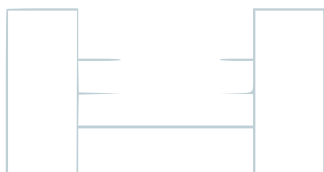
Create your own simple oscilloscope with ADC and 1602 LCD Using MCC | Embedded C Programming - Part 29

Tutorials >

In this tutorial, the reader is guided through the process of designing an oscilloscope using ADC and 1602 LCD. It explores the ...

Create your own simple oscilloscope with ADC and 1602 LCD |

This tutorial delves into Embedded C Programming, using the PIC18F14K50 MCU to create a basic oscilloscope with ADC and a 1602 ...



UART - Universal Asynchronous Receiver Transmitter

Tutorials ›

The article explores the fundamental concepts of UART, including serial communication and full duplex communication. Also, it ...

EXPLORE CIRCUITBREAD

- 216 Tutorials
- 9 Textbooks
- 12 Study Guides
- 31 Tools
- 98 EE FAQs
- 295 Equations Library
- 201 Reference Materials
- 91 Glossary of Terms

FRIENDS OF CIRCUITBREAD



BROWSE PARTS BY PRODUCT

PARTNER STORE WITH ACADEMIC DISCOUNT PRICING

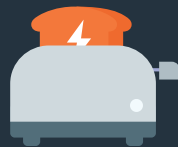


PRODUCT SAMPLES UPON REQUEST

 CircuitBread Discord*Buy me a coffee*

No Comments

[Log In](#) / [Sign Up](#) to leave a comment.



Get the latest tools and
tutorials, fresh from the
toaster.

[LEARN](#)[TOOLS](#)[EQUATIONS](#)[DISTRACTIONS](#)[Team](#) | [Policy](#) | [Terms](#) | [Announcements](#)

Copyright 2024 CircuitBread, a SwellFox project. Headquartered in Beautiful
Downtown Boise, Idaho.