EEE3096S - Tutorial 3

2022

5 ADC Metrics and PWM

5.1 Learning Objective

By the end of this practical, you will have:

- Practical understanding of ADC metrics and how this should help you quantify the performance of an ADC or DAC device; and
- Simple practical experience of using data sampled from an ADC in order to find ADC metrics.

Preparatory notes: For this tutorial, it is assumed you know the basics of sampling; if you feel a bit shaky on this theory, then read over Lecture 24 slides 4 - 23.

5.2 ADC Metrics

Consider the following scenario and then respond to the questions that follow.

Scenario:

Consider that you're in lockdown. You and your colleague are working remotely to implement sampling in the system you are tasked do build. As it happens neither of you have access to the physical platform, but being a smart engineer you hooked it up to the internet as well as a remotely controlled signal generator and remote-controlled power switch before abandoning the office. So you both have access to the platform, which happens to be running embedded Linux and is connected via ssh.

What you do know is that it's an 8-bit ADC. It's a low-cost SPI ADC that isn't of the best quality. You can assume that you've got the code to program it.

But you need to answer a few questions in order to decide on processing solutions to implement. Various data files have been obtained from tests that were run which will help you to characterize the ADC. The ADC is set up to receive a $0 - 2.55$V signal, to convert that to a 8-bit values. The ADC is sampling at 1MHz. It understandably has a DC offset of some sort and saturates at some point.

5.2.1 Question 1

First off, the most obvious thing first: we know it's an 8-bit ADC but that doesn't mean it gives that perfect resolution. Considering that a ramp waveform has been linked up to test things. The ramp is running at 250KHz, and we sample for 10K (as in 10*1024) samples (each sample being 1us, so do the math to see how long the sample block is, if you want to know that). We've set up the voltage range to go from 0 to 2.50V also. But we also already know the device saturates at some point below 2.55V, thus setting the signal generator's maximum voltage above 2.5V wouldn't make sense.

The captured data file is stored in ramp.csv and shown in figure 7.1 below. Investigate the file and determine the answer to the following sub-questions:

1. What is the resolution in bits of the ADC? Based on the samples obtained from the experiment? [5 marks]
2. What is the Q (Quantizing) Resolution in Volts for this system? (i.e. do some inspections to determine the difference between two input voltages causing the digitized output to be incremented by 1). [5 marks]

Provide clear explanations (or working) for both your above answers.

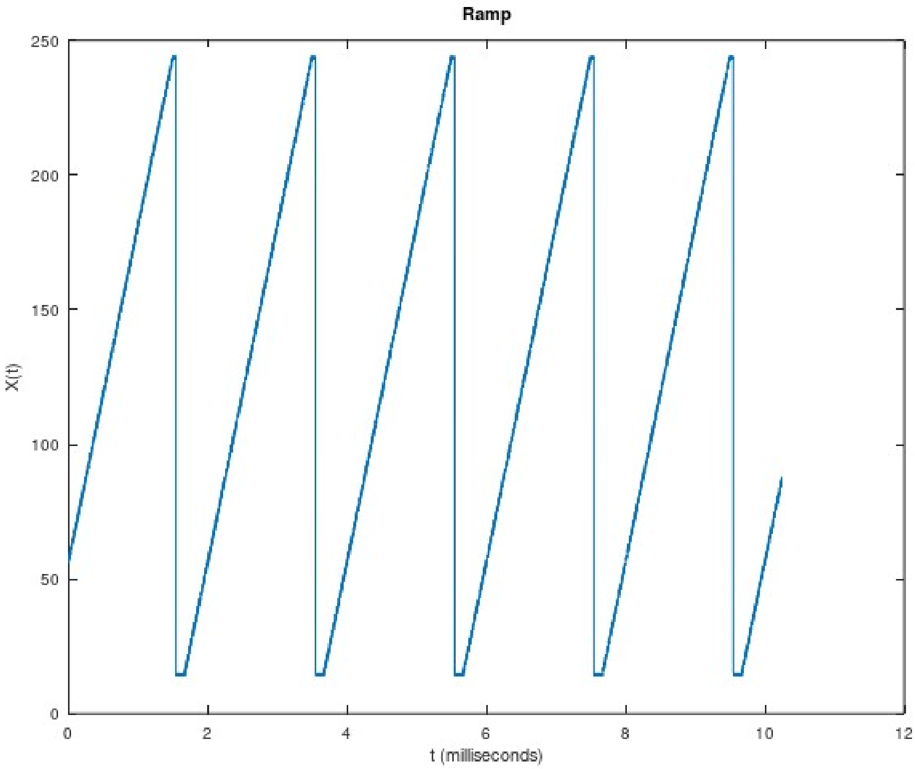

Figure 5.1: Testing of feeding a ramp into the ADC, ramp.csv sample file

5.2.2 Question 2

For this case, we have fed 0V into the ADC (or as good at the remote-controlled signal generator, that is connected to the platform, can do 0V; which is pretty good). The 10K sample recording is provided in zero.csv. You can see it's not exactly zero or perfectly behaved. Basically it is not getting down to 0V. Answer the following based on this recording:

1. What is the direct current (DC) offset error of the ADC when it is connected to OV? [5 marks]
2. A further test was done in which the signal generator was told to generate a frequency of 80KHz, for which we got another 10K samples. The recorded data is provided at freq1.csv. The frequency plot generated from an FFT is shown in figure 7.3. Use this data determine what the Spurious-free Dynamic Range (SFDR). Provide you answer as either power or S/N (or both if you want to be fancy). Note: you need to show your working, you can include a snippet of figures of your working to motivate your answer. As you have probably noticed, there is another signal picked up, assume this is an aliased 50Hz and basically something that one has to live with when using this device. [10 marks]

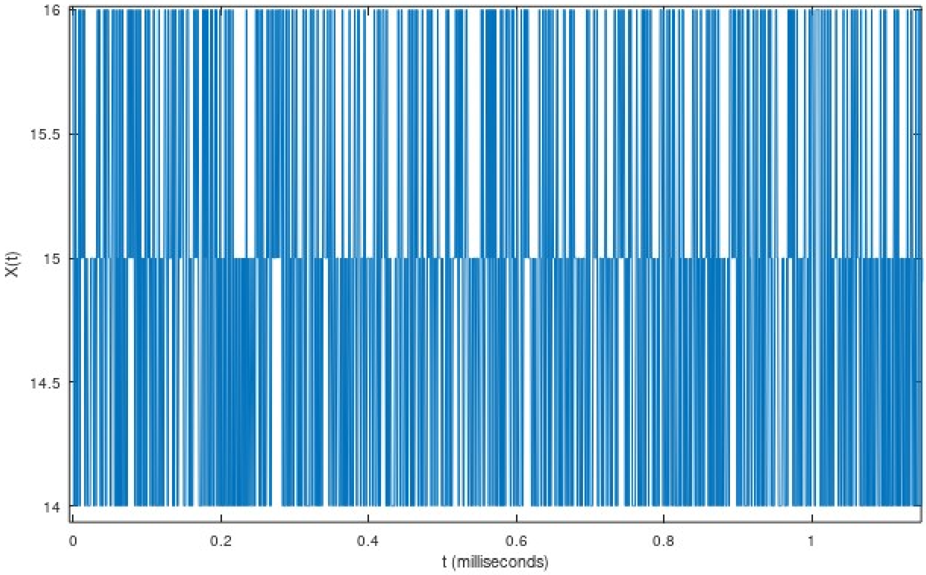

Figure 5.2: Testing at 0V input, a zoomed in view of the zero.csv captured data

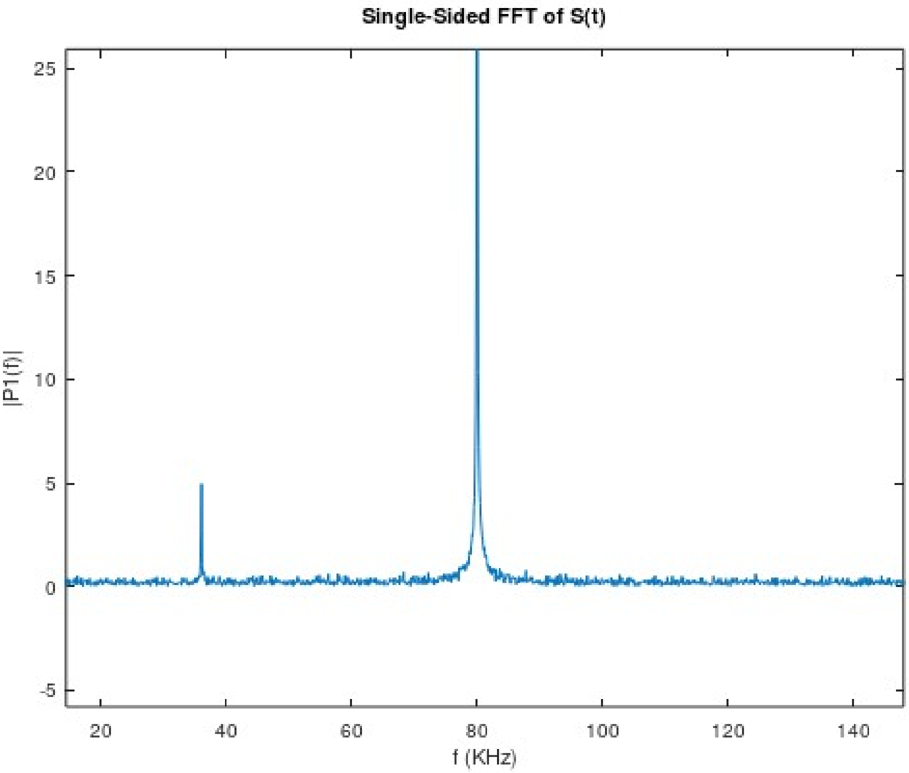

Figure 5.3: Testing with 80KHz input, a zoomed in view of frequency plot

5.3 PWM

A PWM signal is a square wave for which the frequency and duty cycle can be adjusted by the controller. By adjusting the Frequency (the number of repeating cycles per second) and the duty cycle (the percentage of time that the square wave is high versus low), the resulting average voltage of the waveform can be adjusted. By changing the average voltage the controller is able to adjust the average power delivered to a load.

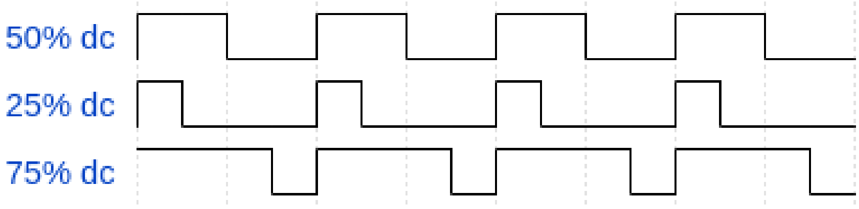

Figure 5.4: Fixed frequency, changing duty cycle PWM

5.3.1 Question 3

1. What is the difference between PWM frequency and its duty cycle? [1 mark]
2. Which parameter should you change if you are trying to increase the brightness of an LED being driven by a PWM signal? [1 mark]
3. Explain the concept of persistence of vision in the context of PWM and why this can be useful in simplifying circuit designs. [2 marks]
4. Design a 555 timer-based circuit for generating a PWM signal with variable duty cycle. Remember that Google is your friend. Make sure to include screenshots from a simulator such as LTSPice. [6 marks]

Submit a single PDF (named correctly with STUDNUM1_STUDNUM2_Tut3.pdf) with your answers to the above. If you pull from any sources, be sure to correctly cite them.