Digital Acquisition of Analog Signals – A Practical Guide

Nathan M. Neihart

Senior Design Presentation
A common task for many senior design projects is to interface an analog signal with a digital system.

The digital system will typically be a microcontroller:
- MSP430
- Arduino Mega

There are several important considerations that need to be made when performing this type of interface:
- Voltage levels
- Interface circuits
- Sampling rates
Outline

- Resolution and Dynamic Range
- Sampling Rate
- Some Common Applications
Interfacing with the ADC

- The analog-to-digital converter takes an analog signal as an input and generates a digital number as the output.

- The ADC has 3 important characteristics:
  - **Reference Voltage**: The maximum voltage that can be converted.
  - **Resolution**: The minimum voltage change that can be detected.
  - **Sampling Rate**: The time interval between consecutive samples.

- We will discuss each of these in detail.

\[\text{ADC} \rightarrow 000, 001, 101, ...\]
Interfacing with the ADC

- All voltages are measured w.r.t. some reference point
  - Usually the reference point is “ground” or “common”
  - Sometimes the reference point can be another voltage

- “Earth ground” is a direct connection to the physical earth

- “Common” is simply a reference point that is common to the entire circuit

- Some power supplies use earth ground as the reference and others use a common terminal as the reference
Resolution and Dynamic Range
Resolution and Voltage Levels

- The reference voltage is the maximum voltage that the ADC can convert
  - The minimum voltage is typically 0 V

- The resolution of the ADC is the smallest voltage change that can be measured

\[
V_{LSB} = \frac{V_{ref}}{2^N - 1}
\]

- \(V_{ref}\) – Reference voltage in V
- \(N\) – Word width of ADC output
Resolution and Voltage Levels

\[
\frac{\text{Resolution of the ADC}}{\text{System Voltage}} = \frac{\text{ADC Reading}}{\text{Analog Voltage Measured}}
\]

◆ Given an analog voltage, we can compute the expected digital output

\[
D_N = V_{\text{in}} \frac{2^N - 1}{V_{\text{ref}}}
\]

\(D_N\) is an N-bit digital word
\(V_{\text{in}}\) is the analog input voltage

◆ Given the ADC digital output, we can compute the expected analog input

\[
V_{\text{in}} = \frac{V_{\text{ref}}}{2^N - 1} D_N
\]

◆ If \(V_{\text{in}} \geq V_{\text{ref}}\) then \(D_N\) is all ONEs
◆ If \(V_{\text{in}} \leq 0\) then \(D_N\) is all ZEROS
In order for your ADC to function properly, your analog input signal must be within range of the ADC reference voltage. Pay attention to circuits that require dual-polarity power supplies.
Resolution and Voltage Levels

- **DC Offset too Low**: The waveform shows a significant DC offset, which is too low. This can affect the accuracy of the signal representation.

- **Amplitude too Large**: The waveform demonstrates an amplitude that is too large, potentially causing distortion or clipping issues in the system's output.
Resolution and Voltage Levels

- You may not always have the ability to control the DC offset of your analog source

- You can shift the average value (or DC offset) of your input signal externally using an op-amp

$$V_{OUT} = -V_{IN} \frac{R_2}{R_1} + V_{DC} \left(\frac{R_2}{R_1} + 1\right)$$

To avoid saturating the output:

$$\frac{V_{SS} + |V_{in,max}| \frac{R_2}{R_1}}{1 + R_2/R_1} \leq V_{DC} \leq \frac{V_{DD} - |V_{in,max}| \frac{R_2}{R_1}}{1 + R_2/R_1}$$
# Resolution and Voltage Levels

- **Maximum resolution** will only be achieved with full-scale input.

- **For inputs with fixed amplitude**:
  - Increase gain before ADC
  - Set the correct \( V_{REF} \)

<table>
<thead>
<tr>
<th>MSP432</th>
<th>ATmega328</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.2 V</td>
<td>( Av_{cc} ) (5 V)</td>
</tr>
<tr>
<td>1.45 V</td>
<td>1.1 V</td>
</tr>
<tr>
<td>2.5 V</td>
<td>Externally Set</td>
</tr>
<tr>
<td>Externally Set</td>
<td>Externally Set</td>
</tr>
</tbody>
</table>
Resolution and Voltage Levels

- For inputs with large dynamic range
  - Variable gain block will provide best resolution

- Gain is controlled by ADC output

- ADC output is now dependent on the gain
  - The MCU must know the current value of gain

\[ G \cdot V_{in} = \frac{V_{ref}}{2^N - 1} D_N \]

\[ D_8 \geq 0b11111xxx \text{ then decrease gain} \]
\[ D_8 \leq 0b00000xxx \text{ then increase gain} \]

\[ 0.03V_{ref} \leq V_{in}G \leq 0.97V_{ref} \]
Resolution and Voltage Levels

- The analog supply voltage may be larger than the ADC supply voltage
  - This is okay if the analog signal is still within safe operating limits of the ADC

- During power up or for invalid inputs the analog signal may exceed safe operating voltages for the ADC
  - You should limit (or clamp) the analog input signal

- The output is limited to: \( V_L \leq V_{OUT} \leq V_H \)

- \( R \) is required to limit the current flowing into the output of the op-amps
Resolution and Voltage Levels

$V_{IN} > V_H$

$V_{IN} < V_L$
Sampling
One of the most important characteristics of the ADC is the sampling rate.

The sampling rate must be at least twice the highest frequency of interest.

If the sampling rate is too low you will get aliasing.

Aliasing is an effect that causes signals with different frequencies to become indistinguishable.
ADC Sampling Rate

Video taken from: https://www.youtube.com/watch?v=vLL-T4Z_TNo
ADC Sampling Rate

◆ Sweep the frequency of a sinusoidal signal from 20 Hz to 15 kHz

◆ Sine wave sampled at $f_s = 44.1$ kHz
  ■ No aliasing

◆ Sine wave sampled at $f_s = 10$ kHz
  ■ Aliasing at high frequencies

◆ Saw tooth wave sampled with $f_s = 30$ kHz
  ■ Why is there still aliasing at high frequencies?
The Fourier series expansion for the Saw tooth wave:

\[ f(t) = -\frac{2}{\pi} \sum_{m=1}^{\infty} \frac{\sin(m\omega t)}{m} \]
ADC Sampling Rate

- When choosing a sampling rate you must know something about the frequency content of your signal.

- An anti-aliasing filter should be used to limit the bandwidth.

\[ f_c = \frac{f_s}{2} \]

\[ \text{ADC} \rightarrow \text{LPF} \rightarrow V_{\text{IN}} \]
often you will need to sample signals very close to the
Nyquist rate

care must be taken when using low sampling rates

\[ V_{in}(t) = \sin(8\pi \times 10^3 t) \]
So how do we control the sampling rate?

Default Arduino ADC clock rate is $f_{ADC} = 125 \, kHz$

Normal conversion requires 13 clock cycles or 104 $\mu$s
  - Maximum sampling rate is 9,615 Hz

The actual sampling rate depends on your implementation
The serial monitor is a very useful debugging tool for Arduino.

It is common to embed the ADC conversion function into the main loop.

Takes approximately 284 μs to execute this loop.

Actual sampling rate is 3,521 Hz!

Sample Arduino Code

```c
int analogPin = 3;
int val = 0;

void setup() {
    Serial.begin(9600);
}

void loop() {
    val = analogRead(analogPin);
    Serial.print("Value is: ");
    Serial.println(val);
}
```
A much better way is to use timers and interrupts

Sampling rate is now tightly controlled by the timer

Sampling rate is still limited by other code and resolution of timer

Sample Arduino Code

#include "TimerOne.h"

int analogPin = 3;
int val = 0;

void setup() {
    Timer1.initialize(1000);
    Timer1.attachInterrupt(callback);
}

void callback() {
    val = analogRead(analogPin);
}

void loop() {
    // Do something with val
}
Common Applications
Measuring Amplitude

◆ Measuring the amplitude of an unknown AC signal with respect to ground

\[ V(t) = V_{DC} + A \sin(\omega t + \theta) \]

\[ v[n] = v_{DC} + A \sin \left( 2\pi \frac{f}{f_s} n + \theta \right) \]

◆ A common method for finding amplitude is to find the maximum value of \( v[n] \)
  ■ Do not forget the DC offset

◆ Does sampling rate matter?
Measuring Amplitude

- Accuracy can be improved by taking at least 5 samples per period
- Measuring RMS voltage is a bit more accurate

\[ RMS\{v[n]\} = \sqrt{\frac{1}{N} \sum_{n} (v[n] - V_{DC})^2} \]
Digitizing Multiple Signal Sources

Four Modes of Operation

1. Single-Channel Single Conversion
2. Sequence-of-Channels
3. Repeat Single-Channel
4. Repeat Sequence-of-Channels
Digitizing Multiple Signal Sources

**IDEAL**

V₁ (Control) → MUX → ADC

**REALISTIC**

V₁ (Control) → MUX → ADC
Summary

◆ Make sure that your analog input signal is within the valid range of the ADC

◆ For maximum resolution, you want the input signal to be close to $V_{ref}$
  ■ Variable amplification may be necessary

◆ Make sure that you are sampling above the Nyquist rate
  ■ Watch out where you place your conversion enable code

◆ You should always low-pass filter your input signal before digitization
  ■ Avoid unwanted aliasing
  ■ Minimize noise contribution from any gain blocks
Thank You