FROM LAST TIME...

System Interfacing (A/D and D/A Conversion)

- Why do A/D and D/A matter?
- Data representation
- Number systems
- Analog ↔ Digital conversion
- Sampling and Aliasing
**ANALOG-TO-DIGITAL (A/D) CONVERSION**

**Flash ADC**

- Uses comparators to determine input voltage range.
- Gate logic converts comparator outputs to digital value.
- Fast! Typical conversion time: 10 – 500 nsec.
- Typically, 4 to 8 bit precision (8 bits requires 254 comparators).
ANALOG-TO-DIGITAL (A/D) CONVERSION

**Flash ADC**

- **Example: 2 bit Flash ADC**

  - **Code:**
    - 00
    - 01
    - 10
    - 11

  - **Values:**
    - 0
    - 1.25
    - 2.5
    - 3.75
    - 5

- **Needs: 3 comparators**

- **Use truth table to get output values:**

<table>
<thead>
<tr>
<th>C3</th>
<th>C2</th>
<th>C1</th>
<th>MSB</th>
<th>LSB</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

8-line to 3-line Priority Encoder
ANALOG-TO-DIGITAL (A/D) CONVERSION

**Flash ADC**

- Need to provide sample-and-hold at the input -- if input is changing during conversion, erroneous values will be produced.
- Output valid for only a short time after input is held.
- Requires $2^N - 1$ comparators.
- Adjacent comparators must have monotonic range change.
**ANALOG-TO-DIGITAL (A/D) CONVERSION**

**Successive Approximation ADC:**
- Workhorse method.
- Used for wide variety of applications
- Typical conversion time: 1 – 100 μsec. Slower than flash ADC.
- Easily extensible to higher precision.
- Precision is limited by the quality of the components.
SUCCESSIVE APPROXIMATION ADC

Basic idea: Check bits starting from the high order bit (MSB).

- Algorithm:

```
START CONVERSION
SET Result to 0
FOR i = N-1 TO 0
    SET i-th bit of Result to 1
    IF INPUT VOLTAGE < DtoA(Result)
        SET i-th bit of Result to 0
END FOR-loop
OUTPUT Result
END CONVERSION
```

This is a form of *interval halving*.
SUCCESSIVE APPROXIMATION ADC

General structure:

- Successive-approximation converters are quite expensive.
- Usually used with a multiplexer -- many channels feed to a single converter.
- Effective conversion speed for multiplexed ADC depends on number of channels used.
- Sample-and-hold normally precedes the converter.
SUCCESSIVE APPROXIMATION ADC

Arduino ADC
6-channel 10-bit ADC
INTEGRATING CONVERTERS

- Slowest of commonly used A/D converters. Typical conversion time is many milliseconds.
- Can be made very accurate and precise – used in DVMs (several conversions per second).
- Uses timing to determine digital value of unknown (input) voltage.
INTERFACING WITH AN ADC

- Successive approximation ADC – use AD673 as example
ANALOG-TO-DIGITAL (A/D) CONVERSION

ADC with Serial Output

- Reduce pinouts and package size
- Can be easily interfaced with a microcontroller or a microprocessor with built in serial interface (SCLK, SDATA, T/R)
- Sample rate limited by the maximum SCLK rate.
- Use AD7476 as example – 1 MSPS 6 pin ADC
OVER-SAMPLING AND SUB-SAMPLING

If A/D conversion is fast enough, it is possible to sample much faster in the converter than is needed by the application:

- Digital filter can then be applied to remove unwanted spectral components.
- Anti-aliasing filter can be split between analog and digital filtering.
- DSP is often used to implement digital filter.
Over-sampling converters may be less expensive or flexible, but they do not have better performance than other converters.

- Stream of over-sampled data has more data values than can be used.
- Sub-sampling picks out every Nth data value and throws away \((N-1)/N\) of the data (N is the over-sampling ratio).
OVERSAMPLING

- One benefit of oversampling is that it spreads out broadband noise over a wide frequency range, thus lowering the “noise floor.” Other sources of noise and other errors cannot be removed by oversampling. Thus, sampling speed can be traded for amplitude resolution.

- Oversampling converters do not have any better performance than other D/A converters. They use knowledge of noise characteristics to get 24-bit performance, say, out of a 20-bit converter, by sampling much faster than the target sampling rate.
UNIT 5:
SYSTEM INTERFACING

PART B:
DIGITAL TO ANALOG CONVERSION
DIGITAL-TO-ANALOG CONVERTER (DAC)

Converts digital values to analog outputs of either voltage or current

- Current DACs require external op-amp to convert back to voltage
Digital-to-Analog (D/A) Conversion

- Digital value is stored in a register (latch), then converted. Duration of conversion is called *settling time*.
- Output of the DAC remains the same until the next value is sent to the register (latch) – a zero-order hold.

**Basic concept:**

\[ V_{OUT} = (b_{N-1} \cdot 2^{N-1} + b_{N-2} \cdot 2^{N-2} + \ldots + b_1 \cdot 2^1 + b_0 \cdot 2^0) \cdot Q \]

Integer equivalent of binary code
DIGITAL-TO-ANALOG (D/A) CONVERSION

Ideal DA Conversion:

\[
\begin{array}{cccccccc}
000 & 001 & 010 & 011 & 100 & 101 & 110 & 111 \\
\hline
0 & \leftarrow & Q \\
FS
\end{array}
\]
DIGITAL-TO-ANALOG (D/A) CONVERSION

DA Conversion Errors:

(a) Offset Error

(b) Scale Factor Error

(c) Linearity Error

(d) Non-monotonicity (due to excessive differential nonlinearity)
DIGITAL-TO-ANALOG (D/A) CONVERSION

DA Conversion Errors:

- Integral Non-linearity (INL): Max. deviation of analog output from ideal line
- Differential Nonlinearity (DNL): Max. deviation of analog output between two adjacent codes

# Digital-to-Analog (D/A) Conversion

Digital to Analog Converters - DAC

![Screenshot from Mouser.com website...](image)

### Applied Filters
- Packaging = Tube

### Available Filters
- 2,936 Matches

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Number of Converters</th>
<th>Resolution</th>
<th>Interface Type</th>
<th>Settling Time</th>
<th>Maximum Operating Temperature</th>
<th>Number of DAC Outputs</th>
<th>Mounting Style</th>
<th>Package / Case</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analog Devices Inc.</td>
<td>1</td>
<td>8 bit</td>
<td>2-Wire, I2C</td>
<td>0.011 us</td>
<td>+70 C</td>
<td>1</td>
<td>SMD/SMT Through Hole</td>
<td>CDIP</td>
</tr>
<tr>
<td>Cirrus Logic</td>
<td>2</td>
<td>10 bit</td>
<td>3-Wire</td>
<td>11 ns</td>
<td>+75 C</td>
<td>2</td>
<td>1</td>
<td>CDIP N</td>
</tr>
<tr>
<td>Exar</td>
<td>3</td>
<td>12 bit</td>
<td>3-Wire, Microwire, QSPI, SPI</td>
<td>0.012 us</td>
<td>+65 C</td>
<td>3</td>
<td>Through Hole</td>
<td>CDIP GB-28</td>
</tr>
<tr>
<td>Intersil</td>
<td>4</td>
<td>13 bit</td>
<td>3-Wire, Serial</td>
<td>12 ns</td>
<td>+105 C</td>
<td>4</td>
<td>Through Hole</td>
<td>CDIP-16</td>
</tr>
<tr>
<td>Maxim Integrated</td>
<td>8</td>
<td>14 bit</td>
<td>3-Wire, SPI</td>
<td>13 ns</td>
<td>+125 C</td>
<td>8</td>
<td>Through Hole</td>
<td>CDIP-18</td>
</tr>
<tr>
<td>Microchip</td>
<td>12</td>
<td>16 bit</td>
<td>DSP, Microwire, QSPI, SPI, Serial</td>
<td>14 ns</td>
<td>+150 C</td>
<td>12</td>
<td>Through Hole</td>
<td>CDIP-20</td>
</tr>
<tr>
<td>NJR</td>
<td>32</td>
<td>17 bit</td>
<td>I2C</td>
<td>0.015 us</td>
<td></td>
<td>32</td>
<td>Through Hole</td>
<td>CDIP-24</td>
</tr>
</tbody>
</table>

**Reset**

[Apply Filters]

[Jeff Shelton – 18 February 2015]
DIGITAL-TO-ANALOG (D/A) CONVERSION

Screenshot from Mouser.com website...

<table>
<thead>
<tr>
<th>Select</th>
<th>Image</th>
<th>Mouser Part #</th>
<th>Mfr. Part #</th>
<th>Mfr.</th>
<th>Description</th>
<th>Availability</th>
<th>Pricing (USD)</th>
<th>Quantity</th>
<th>RoHS</th>
<th>Number of Converters</th>
</tr>
</thead>
<tbody>
<tr>
<td>☐</td>
<td><img src="image1.png" alt="Image" /></td>
<td>595-DAC8812ICPW</td>
<td>DAC8812ICPW</td>
<td>Texas Instruments</td>
<td>Digital to Analog Converters - DAC 16-Bit Dual Serial Input Multiplying</td>
<td>Data Sheet</td>
<td>1,391 In Stock Alternative Packaging</td>
<td><img src="image2.png" alt="" /> $22.01</td>
<td>20.36 25.71 18.71 18.27 18.27 100. View</td>
<td><img src="image3.png" alt="Buy" /></td>
</tr>
<tr>
<td>☐</td>
<td><img src="image5.png" alt="Image" /></td>
<td>584-DAC8420ESZ</td>
<td>DAC8420ESZ</td>
<td>Analog Devices</td>
<td>New At Mouser Digital to Analog Converters - DAC Quad 12B Serial Vcout</td>
<td>Page 3,093 Data Sheet</td>
<td>508 In Stock Alternative Packaging</td>
<td><img src="image6.png" alt="" /> $47.08</td>
<td>45.69 44.31 42.93 42.93 100. View</td>
<td><img src="image7.png" alt="Buy" /></td>
</tr>
</tbody>
</table>
DIGITAL-TO-ANALOG (D/A) CONVERSION

Basic types of D/A Converters:

- String
- Thermometer
- Weighted Resistor
- R/2R Ladder
- Multiplying
- Hybrid DACs (most IC devices)
DIGITAL-TO-ANALOG (D/A) CONVERSION

String DAC

- Fast, but requires large number of resistors and switches (one for each quantization level)
- Widely used in simple DACs

![Diagram of String DAC]

Only 1 switch closed at a time!
DIGITAL-TO-ANALOG (D/A) CONVERSION

Weighted (Scaled) Resistor DAC

- Fast, but not practical for high bit count due to expense of high precision resistors across a wide magnitude range
- Virtual ground at inverting op-amp input
- Requires:
  - Accurate reference voltage
  - Higher precision resistors

\[ V_{OUT} = - \Sigma I_i \frac{R}{2} = - V_{REF} \left( \frac{b_{N-1}}{2} + \frac{b_{N-2}}{4} + \ldots + \frac{b_0}{2^N} \right) \]

Can \( |V_{OUT}| = |V_{REF}| \)?
DIGITAL-TO-ANALOG (D/A) CONVERSION

R/2R Ladder DAC

- Uses just two resistance values (2R and R, closely matched)
- Input switches define a specific resistor divider network

If only MSB (100) is asserted:

\[ V_{OUT} = \frac{4}{8} \cdot V_{REF} = \frac{1}{2} \cdot V_{REF} \]

If all bits are asserted (111):

\[ V_{OUT} = \frac{7}{8} \cdot V_{REF} \]

\[ V_{OUT} = (b_{N-1} \cdot 2^{N-1} + b_{N-2} \cdot 2^{N-2} + \cdots + b_1 \cdot 2^1 + b_0 \cdot 2^0) \cdot \frac{V_{REF}}{2^N} \]
DIGITAL-TO-ANALOG (D/A) CONVERSION

Multiplying DAC

- Conventional DAC has internal reference voltage $V_{REF}$ that is derived from the fixed power supply.
- Multiplying DAC has an externally supplied reference voltage.

$$V_{OUT} = (b_{N-1} \cdot 2^{N-1} + b_{N-2} \cdot 2^{N-2} + \ldots + b_1 \cdot 2^1 + b_0 \cdot 2^0) \cdot \frac{V_{REF}}{2^N}$$

- Advantages:
  - Use a constant frequency sinusoidal reference signal to achieve amplitude modulation, i.e. let $V_{REF} = V_R \sin(\omega t)$.
  - External $V_{REF}$ can be precisely controlled to compensate for drift.
INTERFACING WITH A DAC

- Non-multiplying DAC – use AD558 (8 bit DAC) as example

<table>
<thead>
<tr>
<th>Input Data</th>
<th>CE</th>
<th>CS</th>
<th>DAC Data</th>
<th>Latch Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>“Transparent”</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>“Transparent”</td>
</tr>
<tr>
<td>0</td>
<td>g</td>
<td>0</td>
<td>0</td>
<td>Latching</td>
</tr>
<tr>
<td>1</td>
<td>g</td>
<td>0</td>
<td>1</td>
<td>Latching</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>g</td>
<td>0</td>
<td>Latching</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>g</td>
<td>1</td>
<td>Latching</td>
</tr>
<tr>
<td>X</td>
<td>1</td>
<td>X</td>
<td>Previous Data</td>
<td>Latched</td>
</tr>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>Previous Data</td>
<td>Latched</td>
</tr>
</tbody>
</table>

NOTES
X = Does not matter.
g = Logic Threshold at Positive-Going Transition.

*V_{OUT} = 0V \text{ TO } +2.56V*

<table>
<thead>
<tr>
<th>INPUT CODE</th>
<th>V_{OUT}</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000000</td>
<td>+128V</td>
</tr>
<tr>
<td>10000000</td>
<td>0V</td>
</tr>
<tr>
<td>11111111</td>
<td>-1.27V</td>
</tr>
</tbody>
</table>

Data Inputs:

- \( t_W = \text{STORAGE PULSE WIDTH} = 200\text{ns MIN} \)
- \( t_{DH} = \text{DATA HOLD TIME} = 10\text{ns MIN} \)
- \( t_{DS} = \text{DATA SETUP TIME} = 200\text{ns MIN} \)
- \( t_{SETTLING} = \text{DAC OUTPUT SETTLING TIME TO \pm 1/2 LSB} \)
DIGITAL-TO-ANALOG (D/A) CONVERSION

Resistor methods rely on voltage dividers

- Many precision resistors necessary
- Wasted energy dissipated as heat

Pulse Width Modulation (PWM)

- Rectangular pulse wave
- Duty cycle controls average voltage
- Very high frequency content
- Need a low-pass filter to remove the sharp transitions at edges of the pulses!
- About 90% efficiency
PULSE WIDTH MODULATION (PWM)

Duty Cycle = \( \frac{t_{ON}}{T} \Rightarrow 0 - 100\% \)

Frequency (rad/sec) = \( \frac{1}{T} \)

Diagram:
- 5V
- 0V
- \( t_{ON} \) "ON Time"
- \( t_{OFF} \) "OFF Time"
- Period, \( T \)
PULSE WIDTH MODULATION (PWM)

- Poor man’s DAC
- Low pass filtering the PWM signal can produce an analog signal whose magnitude is proportional to the pulse width of the PWM signal
- For motor/motion control, the motor/motion system will act as the low pass filter
- Unipolar output
- Best suited when an analog output is needed but does not require a high resolution DAC
PULSE WIDTH MODULATION (PWM)

8-bit resolution:
- 100%/255 ⇒ 0.39% per step
- 5V/255 ⇒ 19.6 mV per step

Arduino default PWM frequency:
- Pins 5/6: ~976 Hz
- Pins 3/9/10/11: ~488 Hz
- Frequency can be increased to as much as 62.5 kHz by altering timer control registers
  - Pins 5/6: TCCR0B
  - Pins 9/10: TCCR1B
  - Pins 3/11: TCCR2B
PULSE WIDTH MODULATION (PWM)

PWM1

R
C

+10V
-10V

OPAMP

Analog out

PWM1

5V

25% Duty Cycle

Vc (dB)

0 dB

– 20 dB/decade

\( \omega \)

\( \omega_{co} \)
PULSE WIDTH MODULATION (PWM)

At low PWM frequency ($\omega_{\text{pwm}} \ll \omega_{\text{co}}$), capacitor can fully charge and discharge.
As PWM frequency increases, capacitor can barely charge and discharge fully.
As PWM frequency increases further, capacitor voltage slowly increases, since capacitor cannot fully discharge before next cycle begins.
PULSE WIDTH MODULATION (PWM)

As PWM frequency increases even further, capacitor voltage takes on appearance of a triangular wave, integrating the PWM signal.
PULSE WIDTH MODULATION (PWM)

Image: http://www.youtube.com/watch?v=YaRDbw38x7Q
PULSE WIDTH MODULATION (PWM)

With PWM frequency well above cutoff frequency, capacitor voltage loses saw-tooth appearance

\[ V_{C, \text{steady state}} = V_{\text{pwm}} \times \text{Duty Cycle} \% \]
PULSE WIDTH MODULATION (PWM)

Want signal frequency $\omega_s$ less than $\omega_{co}$. Can't always adjust $\omega_{pwm}$, so often have to play with $\omega_s$ and $\omega_{co}$

$$V_{C, \text{steady state}} = V_{pwm} \times \text{Duty Cycle \%}$$
PULSE WIDTH MODULATION (PWM)

Digital Input

PWM Waveform
COMING UP...

Sensors and Actuators

- Types of sensors and actuators
- Interfacing with sensors and actuators
- DC Motors
- Stepper Motors