Lecture 11: Sampling, ADCs, and DACs

Thomas Schmid
thomas.schmid@utah.edu

Adapted from Prabal Dutta (prabal@umich.edu)
• Practice midterm here:

• HW2 Histogram
  - mean: 54
## Announcements

- **Graduate student presentations (selection due today!)**

<table>
<thead>
<tr>
<th>Date</th>
<th>Presenter/Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mar 19</td>
<td>Spring Break</td>
</tr>
<tr>
<td>Mar 20</td>
<td>Spring break</td>
</tr>
<tr>
<td>Mar 21</td>
<td>D. Crane: F-16 M. McCracken: XBee</td>
</tr>
<tr>
<td>Mar 22</td>
<td>B. Buckley: Teensy S. Tell: SDR on SmartFusion</td>
</tr>
<tr>
<td>Mar 23</td>
<td>A. Forys: Tegra3 J. Davies: CC430</td>
</tr>
<tr>
<td>Mar 24</td>
<td></td>
</tr>
<tr>
<td>Apr  3</td>
<td>M. Thueson: Beagle Board</td>
</tr>
<tr>
<td>Apr  5</td>
<td>J. Christianson: Kindle/eInk/LCD Tech</td>
</tr>
<tr>
<td>Apr 10</td>
<td>No Class, ECE Technical Open House</td>
</tr>
</tbody>
</table>
• USB 1.0 (1996)
  – 1.5 Mbit/s and 12 Mbit/s (Full-Bandwidth)
  – Superseded by USB 1.1 in 1998

• USB 2.0 (2000)
  – 480 Mbit/s (Hi-Speed)
  – Added many new plugs and specifications over time

• USB 3.0 (2008)
  – max of up to 5 Gbit/s
  – reasonable to achieve 3.2 Gbit/s (~400 MB/s)
USB System Design

- One host, multitude of downstream peripheral devices
- Tree topology using USB Hubs (up to 5 levels)
- Peripheral can have multiple functions (logical device)
  - (e.g. Webcam with Microphone)
  - The functions are internally connected to a hub, to which the USB cable is connected.
  - The host assigns a distinctive address to each function.
- Communication happens through logical pipes/endpoints
- Up to 32 endpoints per function (logical device/interface)
  - 16 host->logical device, 16 logical device->host
  - 1 of each reserved for configuration (endpoint 0)
- Two types of pipes: stream and messaging
  - stream: uni-directional isochronous (guaranteed data rate), interrupt (bounded latency), or bulk transfer (as fast as possible)
  - messaging: bi-directional control transfer
Discuss which pipe for which logic device

- Two types of pipes: stream and messaging
  - stream: uni-directional isochronous (guaranteed data rate), interrupt (bounded latency), or bulk transfer (as fast as possible)
  - messaging: bi-directional control transfer
USB Signaling

- 4 USB speeds
  - Low-Speed 1.5 Mbit/s USB 1.0
  - Full-Speed 12 MBit/s USB 1.0
  - High-Speed 480 Mbit/s USB 2.0
  - SuperSpeed 5.0 Gbit/s USB 3.0 (Full Duplex)

- Twisted-pair cable with D+ and D- for USB < 3.0

- Additional pair for USB 3.0 (Full Duplex)

- Differential signaling with None Return to Zero Inverted
  - 0: transition at the physical level
  - 1: no transition

- Voltage signal in the differential pair
  - \( \Delta t = \frac{1}{12\text{ MHz}} \approx 83 \text{ ns} \)
  - \( \Delta V = 3 \text{ V} \)

- Differential decoding
- NRZI decoding

- Packet format
  - Start of packet / clock sync
  - Packet ID (LSB first, 1010 = NAK)
  - End of packet

Image from Wikipedia
USB Communication

- Initially, data transmitted as packets
- After sync, 8-bit bytes, LSB first
- 4-bit PID, followed by bit-wise complement (redundancy)

<table>
<thead>
<tr>
<th>Type</th>
<th>PID value (msb-first)</th>
<th>Transmitted byte (lsb-first)</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reserved</td>
<td>0000</td>
<td>0000 1111</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Token</td>
<td>1000</td>
<td>0001 1110</td>
<td>SPLIT</td>
<td>High-bandwidth (USB 2.0) split transaction</td>
</tr>
<tr>
<td></td>
<td>0100</td>
<td>0010 1101</td>
<td>PING</td>
<td>Check if endpoint can accept data (USB 2.0)</td>
</tr>
<tr>
<td>Special</td>
<td>1100</td>
<td>0011 1100</td>
<td>PRE</td>
<td>Low-bandwidth USB preamble</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ERR</td>
<td>Split transaction error (USB 2.0)</td>
</tr>
<tr>
<td></td>
<td>0010</td>
<td>0100 1011</td>
<td>ACK</td>
<td>Data packet accepted</td>
</tr>
<tr>
<td>Handshake</td>
<td>1010</td>
<td>0101 1010</td>
<td>NAK</td>
<td>Data packet not accepted; please retransmit</td>
</tr>
<tr>
<td></td>
<td>0110</td>
<td>0110 1001</td>
<td>NYET</td>
<td>Data not ready yet (USB 2.0)</td>
</tr>
<tr>
<td></td>
<td>1110</td>
<td>0111 1000</td>
<td>STALL</td>
<td>Transfer impossible; do error recovery</td>
</tr>
<tr>
<td>Token</td>
<td>0001</td>
<td>1000 0111</td>
<td>OUT</td>
<td>Address for host-to-device transfer</td>
</tr>
<tr>
<td></td>
<td>1001</td>
<td>1001 0110</td>
<td>IN</td>
<td>Address for device-to-host transfer</td>
</tr>
<tr>
<td></td>
<td>0101</td>
<td>1010 0101</td>
<td>SOF</td>
<td>Start of frame marker (sent each ms)</td>
</tr>
<tr>
<td></td>
<td>1101</td>
<td>1011 0100</td>
<td>SETUP</td>
<td>Address for host-to-device control transfer</td>
</tr>
<tr>
<td>Data</td>
<td>0011</td>
<td>1100 0011</td>
<td>DATA0</td>
<td>Even-numbered data packet</td>
</tr>
<tr>
<td></td>
<td>1011</td>
<td>1101 0010</td>
<td>DATA1</td>
<td>Odd-numbered data packet</td>
</tr>
<tr>
<td></td>
<td>0111</td>
<td>1110 0001</td>
<td>DATA2</td>
<td>Data packet for high-bandwidth isochronous transfer (USB 2.0)</td>
</tr>
<tr>
<td></td>
<td>1111</td>
<td>1111 0000</td>
<td>MDATA</td>
<td>Data packet for high-bandwidth isochronous transfer (USB 2.0)</td>
</tr>
</tbody>
</table>

Table from Wikipedia
• Announcements

• Sampling

• DACs

• ADCs & Errors
Demo Time
We live in an analog world

- Everything in the physical world is an analog signal
  - Sound, light, temperature, pressure

- Need to convert into electrical signals
  - Transducers: converts one type of energy to another
    - Electro-mechanical, Photonic, Electrical, ...
  - Examples
    - Microphone/speaker
    - Thermocouples
    - Accelerometers
Transducers convert one form of energy into another

- Transducers
  - Allow us to convert physical phenomena to a voltage potential in a well-defined way.

A transducer is a device that converts one type of energy to another. The conversion can be to/from electrical, electro-mechanical, electromagnetic, photonic, photovoltaic, or any other form of energy. While the term transducer commonly implies use as a sensor/detector, any device which converts energy can be considered a transducer. – Wikipedia.
Convert light to voltage with a CdS photocell

\[ V_{\text{signal}} = (+5V) \frac{R}{R + R_R} \]

- Choose \( R = R_R \) at median of intended range
- Cadmium Sulfide (CdS)
- Cheap, low current
- \( t_{RC} = C_1(R+R_R) \)
  - Typically \( R \sim 50-200k\Omega \)
  - \( C \sim 20\text{pF} \)
  - So, \( t_{RC} \sim 20-80\text{us} \)
  - \( f_{RC} \sim 10-50\text{kHz} \)

Source: Forrest Brewer
Many other common sensors (some digital)

- **Force**
  - strain gauges - foil, conductive ink
  - conductive rubber
  - rheostatic fluids
    - Piezoresistive (needs bridge)
  - piezoelectric films
  - capacitive force
    - Charge source

- **Sound**
  - Microphones
    - Both current and charge versions
  - Sonar
    - Usually Piezoelectric

- **Position**
  - microswitches
  - shaft encoders
  - gyros

- **Acceleration**
  - MEMS
  - Pendulum

- **Monitoring**
  - Battery-level
    - voltage
  - Motor current
    - Stall/velocity
  - Temperature
    - Voltage/Current Source

- **Field**
  - Antenna
  - Magnetic
    - Hall effect
    - Flux Gate

- **Location**
  - Permittivity
  - Dielectric

Source: Forrest Brewer
Going from analog to digital

• What we want

Physical Phenomena → Engineering Units

• How we have to get there

Physical Phenomena → Sensor → ADC → Software → Engineering Units
Voltage or Current → ADC Counts
How do we represent an analog signal?
- As a time series of discrete values
  → On MCU: read the ADC data register periodically
• What do the sample values represent?
  - Some fraction within the range of values

→ What range to use?

Choosing the horizontal range

![Graphs showing range too small, range too big, and ideal range.](image)
Choosing the horizontal granularity

- **Resolution**
  - Number of discrete values that represent a range of analog values
  - MSP430: 12-bit ADC
    - 4096 values
    - Range / 4096 = Step
      
      \[ \text{Larger range} \Rightarrow \text{less information} \]

- **Quantization Error**
  - How far off discrete value is from actual
  - \( \frac{1}{2} \) LSB \( \Rightarrow \) Range / 8192
    
    \[ \text{Larger range} \Rightarrow \text{larger error} \]
Converting between voltages, ADC counts, and engineering units

- Converting: ADC counts $\leftrightarrow$ Voltage

\[ N_{ADC} = 4095 \times \frac{V_{in} - V_{R-}}{V_{R+} - V_{R-}} \]

\[ V_{in} = N_{ADC} \times \frac{V_{R+} - V_{R-}}{4095} \]

- Converting: Voltage $\leftrightarrow$ Engineering Units

\[ V_{TEMP} = 0.00355(TEMP_C) + 0.986 \]

\[ TEMP_C = \frac{V_{TEMP} - 0.986}{0.00355} \]
A note about sampling and arithmetic

• Converting values in 16-bit MCUs

\[ V_{\text{TEMP}} = N_{ADC} \times \frac{V_{R^+} - V_{R^-}}{4095} \]

\[ \text{TEMP}_C = \frac{V_{\text{TEMP}} - 0.986}{0.00355} \]

\[
\begin{align*}
\text{vtemp} &= \text{adccount}/4095 \times 1.5; \\
\text{tempc} &= (\text{vtemp} - 0.986)/0.00355;
\end{align*}
\]

\[ \rightarrow \text{tempc} = 0 \]

• Fixed point operations
  - Need to worry about underflow and overflow

• Floating point operations
  - They can be costly on the node
Choosing the sample rate

- What sample rate do we need?
  - Too little: we can’t reconstruct the signal we care about
  - Too much: waste computation, energy, resources
    - Example: 2-bytes per sample, 4 kHz \(\rightarrow\) 8 kB / second

\[ f(x) \]

\[ f_{\text{sampled}}(x) \]
Shannon-Nyquist sampling theorem

- If a continuous-time signal $f(x)$ contains no frequencies higher than $f_{max}$, it can be completely determined by discrete samples taken at a rate:

$$f_{samples} > 2f_{max}$$

- Example:
  - Humans can process audio signals 20 Hz – 20 KHz
  - Audio CDs: sampled at 44.1 KHz
• **Aliasing**
  - Different frequencies are indistinguishable when they are sampled.

![Diagram of aliasing](image)

• **Condition the input signal using a low-pass filter**
  - Removes high-frequency components
  - (a.k.a. anti-aliasing filter)
Designing the anti-aliasing filter

- Note
  - $\omega$ is in radians
  - $\omega = 2\pi f$

- Exercise: Find an R+C pair so that the half-power point occurs at 30 Hz
Can use dithering to deal with quantization

- **Dithering**
  - Quantization errors can result in large-scale patterns that don’t accurately describe the analog signal
  - Introduce random (white) noise to randomize the quantization error.

![Direct Samples](image1.png) ![Dithered Samples](image2.png)
Lots of other issues

• Might need anti-imaging filter

• Cost and power play a role

• Might be able to avoid analog all together
  - Think PWM when dealing with motors…
• Announcements

• Sampling

• DACs

• ADCs
A decoder-based DAC architecture in linear and folded forms
A binary-scaled DAC architecture in linear and folded forms

- Much more efficient
- Monotonicity not guaranteed
- May experience glitches
DAC output signal conditioning

- Often use a low-pass filter
- May need a unity gain op amp for drive strength
• Announcements
• Sampling
• DACs
• ADCs
ADC #1: Flash

Vref

Vin

R

R

R

R

Vcc

priority encoder

Dout

3

2

1

0
• Start: Reset counter, discharge C.
• Charge C at fixed current I until \( V_c > V_{in} \). How should C, I, n, and CLK be related?
• Final counter value is \( D_{out} \).
• Conversion may take several milliseconds.
• Good differential linearity.
• Absolute linearity depends on precision of C, I, and clock.
ADC #3: Successive Approximation (SAR)

- Requires \( \log(\text{bits}) \) cycles per sample
- Goes from MSB to LSB
- Not good for high-speed ADCs

1 Sample \( \rightarrow \) Multiple cycles
Figures and some text from:

- Understanding analog to digital converter specifications. By Len Staller
  - http://www.embedded.com/showArticle.jhtml?articleID=60403334

Key concept here is that the specification provides worst case values.
Transfer function shifted to the left 1/2 LSB to maintain <1/2LSB quantization error.

Highest code boundary at 1 1/2LSB below full-scale input voltage.

First code boundary at 1/2LSB.
Sometimes the intentional $\frac{1}{2}$ LSB shift is included here!
DNL value given in a spec is the worst-case (Same with all the others…)
Full-scale error is also sometimes called “gain error”

*full-scale error* is the difference between the ideal code transition to the highest output code and the actual transition to the output code when the offset error is zero.
The integral nonlinearity (INL) is the deviation of an ADC's transfer function from a straight line. This line is often a best-fit line among the points in the plot but can also be a line that connects the highest and lowest data points, or endpoints. INL is determined by measuring the voltage at which all code transitions occur and comparing them to the ideal. The difference between the ideal voltage levels at which code transitions occur and the actual voltage is the INL error, expressed in LSBs. INL error at any given point in an ADC's transfer function is the accumulation of all DNL errors of all previous (or lower) ADC codes, hence it's called integral nonlinearity.
Questions?

Comments?

Discussion?