CSE190 Fall 2023
Lecture 18
Analog and Wireless

Wireless Embedded Systems
Aaron Schulman
Converting between voltages, ADC counts, and engineering units

• Converting: ADC counts ↔ 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 ↔ 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 fixed-point MCUs

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

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

float \ vtemp = \text{adccount}/4095 * 1.5;
float \ tempc = \text{(vtemp-0.986)}/0.00355;

\[\Rightarrow \ vtemp = 0! \ \text{Not what you intended, even when vtemp is a float!} \]
\[\Rightarrow \ \text{tempc} = -277 \ \text{C}\]

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

• Floating point operations
  – They can be costly on the embedded system
Try it out for yourself...

$ cat arithmetic.c
#include <stdio.h>

int main() {

    int adccount = 2048;
    float vtemp;
    float tempc;

    vtemp = adccount/4095 * 1.5;
    tempc = (vtemp-0.986)/0.00355;

    printf("vtemp: %f\n", vtemp);
    printf("tempc: %f\n", tempc);
}

$ gcc arithmetic.c

$ ./a.out
vtemp: 0.000000
tempc: -277.746490
Oversampling
(sampling faster than Nyquist)

One interesting trick is that you can use oversampling to help reduce the impact of quantization error.

– Let’s look at an example of oversampling plus dithering to get a 1-bit converter to do a much better job...
Oversampling a 1-bit ADC w/ noise & dithering (cont)

- Uniformly distributed random noise ±250 mV
- V_{\text{thresh}} = 500 mV
- V_{\text{rand}} = 500 mV

Note:
- N_{1} is the # of ADC counts that = 1 over the sampling window
- N_{0} is the # of ADC counts that = 0 over the sampling window

Count

Voltagge

0 mV

375 mV

500 mV

0

1

N_{1} = 11

N_{0} = 32
Oversampling a 1-bit ADC w/ noise & dithering (cont)

• How to get more than 1-bit out of a 1-bit ADC?
• Add some noise to the input
• Do some math with the output
• Example
  – 1-bit ADC with 500 mV threshold
  – Vin = 375 mV → ADC count = 0
  – Add ±250 mV uniformly distributed random noise to Vin
  – Now, roughly
    • 25% of samples \((N_1) \geq 500\) mV → ADC count = 1
    • 75% of samples \((N_0) < 500\) mV → ADC count = 0
Can use dithering to deal with quantization

- Dithering (introducing noise)
  - Quantization errors can result in large-scale patterns that don’t accurately describe the analog signal
  - Oversample and dither
  - Introduce random (white) noise to randomize the quantization error.
Selection of a DAC (digital to analog converter)

- **Error/Accuracy/Resolution**: Quantizing error represents the difference between an actual analog value and its digital representation. Ideally, the quantizing error should not be greater than $\pm \frac{1}{2}$ LSB.

**Output Voltage Range** -> Input Voltage Range

- **Output Settling Time** -> Conversion Time

- **Output Coding** (usually binary)
Wireless communication is ubiquitous

- Bluetooth/WiFi/Cellular

- Device-to-Device (Bluetooth-Low-Energy) Device-to-infrastructure (LTE or WiFi)

- Bluetooth Low Energy
  the most popular wireless protocol.
Outline

• What are radios
  o How do they work?
• Fundamental characteristics
  o Design tradeoffs
• Common radio standards/protocols for indoor applications
  o Where characteristics fall under (above)
• Emerging radio standards/protocols for outdoor Internet-of-Things applications
  o Why the design requirement of IoT radios is different
What are Radios?

- A device that enables wireless transmission of signals
  - Electromagnetic wave
  - Transmitter encodes signal and receiver decodes it
How Radios Work - Transmitting

- **Modulation**
  - Converts digital bits to an analog signal
  - Encodes bits as changes in a **carrier frequency**:
  - Frequency, Amplitude, Phase, etc.

![Modulation Diagram](image)
Example of modulating digital data onto an analog signal

Data

Carrier

Modulated Signal
How Radios Work - Receiving

• Simplest is Envelope Detection
  o Detect changes in carrier freq.
  o Complex require synchronization
• All require filtering
• Signals must be demodulated
How Radios Work – Receiving (AM Radio Example)

• Antenna picks up modulated radio waves
• Tuner filters out specific frequency ranges
• Amplitude variations detected with demodulation
• Amplifier strengthens the clipped signal and sends it through the speaker
Radio Characteristics

• Why so many protocols for indoor and outdoor applications?

• All radios have to make tradeoffs
  o Short vs. long distance
  o High vs. low power/energy
  o High vs. low speeds
  o Large vs. small number of devices
  o Device-to-device, device-to-infrastructure
  o Indoor vs. outdoor usages